



m-SportGIS

***Development of a Mobile Solution for
Spatial Data Collection using
Open Source Technologies***

Carlos Eduardo de Abreu Freire e Lopes

Dissertation submitted in partial fulfilment of the requirements for
the Degree of *Mestre em Ciência e Sistemas de Informação
Geográfica*
(Master in Geographical Information Systems and Science)

Instituto Superior de Estatística e Gestão de Informação
da Universidade Nova de Lisboa

m-SportGIS

Development of a Mobile Solution for Spatial Data Collection using Open Source Technologies

Dissertation Supervised by
Professor Marco Painho, Full Professor

September 2014

“It’s more fun to be a pirate than to join the Navy.”

- Steve Jobs, 1982 -

Acknowledgments

First and foremost, I offer my sincerest gratitude to my supervisor, Professor Marco Painho, who has supported me throughout my Master's thesis with his patience and knowledge, whilst allowing me convenient conditions to work on my own way. I attribute the level of my Master's degree to his encouragement and friendlier supervision.

Besides my advisor, I would like to express my true appreciation to my coworkers at New Technologies Laboratory (NT Lab) of Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa (ISEGI-NOVA) for their encouragement and friendship, namely Tiago Oliveira, Alexandre Baptista, Luís Almeida and Júlio Caineta, and in particular to Georg Tschorn, Luís Calisto and Otávio Sian for their insightful contributions.

Finally, I thank my family and close friends for the continuous belief and support during my life that also contributed to finish my present work.

m-SportGIS
Development of a Mobile Solution for Spatial Data Collection using
Open Source Technologies

Abstract

Information technologies (ITs), and sports resources and services aid the potential to transform governmental organizations, and play an important role in contributing to sustainable communities development, respectively. Spatial data is a crucial source to support sports planning and management. Low-cost mobile geospatial tools bring productive and accurate data collection, and their use combining a handy and customized graphical user interface (GUI) (forms, mapping, media support) is still in an early stage. Recognizing the benefits — efficiency, effectiveness, proximity to citizens — that Mozambican Minister of Youth and Sports (MJD) can achieve with information resulted from the employment of a low-cost data collection platform, this project presents the development of a mobile mapping application (app) — m-SportGIS — under Open Source (OS) technologies and a customized evolutionary software methodology. The app development embraced the combination of mobile web technologies and Application Programming Interfaces (APIs) (*e.g.* Sencha Touch (ST), Apache Cordova, OpenLayers) to deploy a native-to-the-device (Android operating system) product, taking advantage of device's capabilities (*e.g.* File system, Geolocation, Camera). In addition to an integrated Web Map Service (WMS), was created a local and customized Tile Map Service (TMS) to serve up cached data, regarding the IT infrastructures limitations in several Mozambican regions. m-SportGIS is currently being exploited by Mozambican Government staff to inventory all kind of sports facilities, which resulted and stored data feeds a WebGIS platform to manage Mozambican sports resources.

Resumo

As tecnologias de informação e os recursos desportivos detêm, respetivamente, o potencial para transformar os serviços das organizações governamentais, e exercem um papel importante no desenvolvimento das comunidades. Dados georreferenciados constituem uma fonte crucial no suporte ao planeamento e gestão desportiva. O uso de ferramentas geográficas de baixo custo favorecem sem dúvida a aquisição de dados, e o seu uso conciliando interfaces personalizadas (formulários, mapas, multimédia) com uma utilização intuitiva encontra-se ainda numa fase inicial. Reconhecendo os benefícios — eficiência, eficácia, proximidade aos cidadãos — que o Ministério da Juventude e Desportos (MJD) pode alcançar com a informação obtida a partir de uma solução tecnológica de baixo-custo para a aquisição de dados, este projeto descreve o desenvolvimento de uma aplicação móvel para mapeamento de entidades em tempo real — m-SportGIS — implementada em código aberto, de acordo com uma metodologia de desenvolvimento evolutivo. A construção da aplicação abrangeu a combinação de tecnologias móvel e web, e bibliotecas de código aberto (ST, Apache Cordova, OpenLayers) para exibir uma solução nativa-para-o-dispositivo (sistema operativo Android), tirando partido das suas capacidades nativas (*e.g.* File system, Geolocation, Camera). Adicionalmente ao serviço de mapas online, foi criado um serviço local e customizado de mapas offline (mosaicos), para permitir o mapeamento de entidades independentemente do serviço de cobertura móvel, tendo em conta as limitações dessa infraestrutura em diversas regiões de Moçambique. Neste momento, a m-SportGIS está a ser explorada pelas equipas do Governo Moçambicano para inventariar todo o tipo de instalações desportivas, em que a base de dados serve de fonte de informação a uma plataforma WebGIS para disseminar, processar, e gerir instalações desportivas.

Keywords

Apache Cordova

Hybrid Software Development

Minister of Youth and Sports

Mobile Mapping

MVC

OpenLayers

Sencha Touch

Spatial Data Collection

Tile Map Service

Palavras-Chave

Apache Cordova

Desenvolvimento Híbrido de Software

Ministério da Juventude e Desportos

Mapeamento Móvel

MVC

OpenLayers

Sencha Touch

Aquisição de Dados Espaciais

Serviço de Mapas em Cache

Acronyms

ADT	Android Development Tools
Ajax	Asynchronous JavaScript and XML
API	Application Programming Interface
App	Application
ASCII	American Standard Code for Information Interchange
BBox	Bounding Box
BSD	Berkeley Software Distribution
CENACARTA	Centro Nacional de Cartografia e Teledetecção
CGI	Common Gateway Interface
CPU	Central Processing Unit
DOM	Document Object Model
EPSG	European Petroleum Survey Group
FOSS	Free and Open Source Software
GI	Geospatial Information
GIS	Geographic Information System
GPS	Global Positioning System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IIS	Internet Information Server
INE	Instituto Nacional Estatística
INGC	National Disasters Management Institute
ISD	Information System Development
ISDM	Information Systems Development Methodology

ISEGI-NOVA Instituto Superior de Estatística e Gestão de Informação, Universidade
Nova de Lisboa

ISs Information Systems

IT Information Technology

Lat Latitude

LGPL Lesser General Public License

Lon Longitude

MJD Ministério da Juventude e Desportos

MVC Model-View-Controller

NT Lab New Technologies Laboratory

OS Open Source

OSGeo Open Source Geospatial Foundation

OSM OpenStreetMap

PNG Portable Network Graphics

QGIS Quantum GIS

RAD Rapid Application Development

RAM Random-Access Memory

SCSS Sassy CSS

SDI Spatial Data Infrastructure

SDK Software Development Kit

SDLC System Development Life-Cycle

SRS Spatial Reference System

SSADM Structured System Analysis and Design Method

ST Sencha Touch

SWOT Strengths Weaknesses Opportunities Threats

TMS Tile Map Service

UI User Interface

URL Uniform Resource Locator

UX User Experience

WFS Web Feature Service

WMS Web Map Service

WMS-C Web Mapping Service-Cached

WSGI Web Server Gateway Interface

Table of Contents

Acknowledgments	iii
Abstract	iv
Resumo	v
Keywords	vi
Palavras-Chave	vii
Acronyms	viii
1 Introduction	1
1.1 Background	1
1.2 Problem Statement and Motivation	4
1.3 Objectives	5
1.4 Assumptions	5
1.5 General Project Methodology	6
1.6 Structure	6
2 Mobile GIS Data Collection in Sports Inventory Systems	9
2.1 Introduction	9
2.2 GIS Data Collection	10
2.3 Mobile GIS Technology	11
2.3.1 Tablet PC and Smartphone Technology	12
2.4 Spatial Planning for Sports Venues Management	15
2.4.1 Projects in Sports Data Collection	17
2.5 Mobile GIS Mapping: State of the Art (Related Work)	18
2.5.1 Employment of Handheld Devices in Mapping	19
3 Hybrid Mobile Development, Mapping Mechanisms, and Software Development Methodologies	21
3.1 Hybrid Mobile Development	21
3.1.1 Thinking Mobile	21
3.1.2 Mobile Applications: Native <i>vs.</i> Web <i>vs.</i> Hybrid	22
3.1.3 Designing JavaScript-Driven Mobile Applications	27

3.1.4	Mobile Web-based Standards	30
3.1.5	JavaScript User Interface Frameworks for Mobile	31
3.2	Mapping Mechanisms	39
3.2.1	Geolocation Services	39
3.2.2	Web Map Client (WMC)	40
3.3	Software Development Methodologies in Organizations	49
3.3.1	Methodology for m-SportGIS Development	53
4	Development and Implementation of m-SportGIS	55
4.1	Introduction	55
4.2	System Requirements Overview	55
4.3	System Architecture	56
4.4	Development and Implementation	57
4.4.1	Data Definition and Preparation	57
4.4.2	Creation of a Tiled Map Service	58
4.4.3	System Features and User Interface	60
4.5	WebGIS Platform	79
4.6	Considerations	80
4.7	Usage of m-SportGIS and Statistical Results	81
5	Conclusions	83
5.1	Synthesis	83
5.2	Contributions	84
5.3	Further Developments	84
5.4	Dissemination of Project's Research and Results	84
	References	85
	Appendix A	95
	Appendix B	101
	Appendix C	129
	Appendix D	132

List of Tables

2.1	SWOT Analysis of Mobile Computing Methods for Field Mapping (adapted from Wagtendonk and Jeu (2007)).	11
2.2	Sport framework: spatial planning activity and outcomes of innovation (adapted from England (2013); Sport England (2005)).	16
3.1	Native mobile app: pros and cons (source: Boston Technology Corporation (2013)).	24
3.2	Mobile Web App: Pros and Cons (source: Boston Technology Corporation (2013)).	25
3.3	ST API Groups (adapted from Garcia, Moss, and Simoens (2013)). . . .	35
3.4	Groups of UI Widgets in ST Framework.	38
3.5	Problem Situations and Methodologies (adapted from Zaied, Aal, and Hassan (2013)).	51
A.1	HTML5 offline storage capacity (adapted from Hochdörfer (2013)). . .	95
A.2	Classes of Shapefiles Selected for QGIS Project.	95
A.4	m-SportGIS Tree Structure (Fields Form Navigation Panel).	95
A.3	Tile content (layers) according to zoom level.	100

List of Figures

1.1	Project Structure.	7
2.1	Principles of sports spatial planning (Sport England (2005)).	16
2.2	Guidelines for sports grounds planning (adapted from Sportbuchhandlung (2000)).	17
3.1	HTML5 vs. Native features scope (icapps.com/html5-vs-native, accessed on 4 October 2013).	25
3.2	Hybrid Capability = Native + HTML5 Aptitudes (wiki.developerforce.com/page/Wiki, accessed on 27 September 2013). 26	
3.3	PhoneGap Architecture (source: slideshare.net/loianeg/corso-phonegap-cordova-aula-01-introduo-ao-phonegap)	27
3.4	Mobile Web-based App Framework Structure (adapted from Williams (2012)).	27
3.5	MVC Components Communication (Williams (2012)).	33
3.6	ST MVC Structure (adapted from Kosmaczewski (2012)).	34
3.7	ST Data Infrastructure Classes (source: Kumar (2011)).	34
3.8	ST App Architecture Components (Kumar (2012)).	36
3.9	ST App Folder Structure.	36
3.10	ST Components (Example of Container and Toolbar).	38
3.11	Web Map Client - Map Server Model (adapted from Hazzard (2011)). .	41
3.12	MapServer applications architecture (source: University of Minnesota (2013)).	44
3.13	Tile Server Infrastructure.	45
3.14	Tile cache hierarchy (adapted from (ESRI, 2006; Potmesil, 1997)). . . .	47
3.15	Tile map origin and coordinates scheme (source: OSGeo (2012)). . . .	47
3.16	Tiles hierarchical grid arrangement (adapted from Xu et al. (2012)). . .	47
3.17	GeoWebCache Architecture (source: OpenGeo (2013b)).	49
3.18	Waterfall <i>vs</i> Agile Methodologies (adapted from Prakash, Kumar, and Mishra (2013)).	52
4.1	National Atlas of Sports System Architecture (source: de Abreu Freire and Painho (2014)).	55
4.2	m-SportGIS Architecture.	56

4.3	QGIS Project.	58
4.4	Tiles Production Schema.	59
4.5	Mock-up: Initial View.	61
4.6	Mock-up: Collecting View with Map Panel (BaseLayers).	61
4.7	Mock-up: Displaying Picture Panel over Form View.	61
4.8	Mock-up: Displaying Saved Records List and Map Panel.	61
4.9	ST Resources Content.	62
4.10	m-SportGIS: Start View.	64
4.11	m-SportGIS: Technician ID (“ID Técnico”) View.	64
4.12	m-SportGIS: Facility ID (“ID Instalação”) View and Select Field.	72
4.13	m-SportGIS: MultiListOverlay and MultiListOverlayItem Classes.	73
4.14	m-SportGIS: MultiListOverlay Class in Sports Activities (“Modalidades”) Field Set.	74
4.15	m-SportGIS: Technical Features (“Características”) View.	75
4.16	m-SportGIS: Informations (“Informações”) View.	75
4.17	m-SportGIS: Example of Record Validation Alert.	76
4.18	m-SportGIS: List of Saved Records and Spatial Location View.	76
4.19	m-SportGIS: Data Edition (Saved Record) View.	77
4.20	m-SportGIS: Saved Photo Indexed to a Record View.	77
4.21	m-SportGIS Package Folder Structure.	78
4.22	m-SportGIS assets/www Content.	78
4.23	WebGIS: Displaying Sports Facilities Spatial Distribution.	79
4.24	Displaying Statistical/Demographical Analysis.	79
4.25	Displaying Sports Facilities Spatial Analysis.	80
4.26	Displaying Outputted Report.	80
4.27	Number of Collected Facilities by Month.	81
4.28	Number of Collected Facilities per Type of (Main) Sports Activity by Province.	82

List of Program Listings

3.1	ST 2 Component (Fieldset) Code Fragment.	37
3.2	Script Tags to Load OpenLayers API and Google Maps API (version 3.X).	41
3.3	Instances and Methods to Create and Render a Simple OSM.	41
3.4	Instantiation of the Hybrid Layer of Google Maps.	41
3.5	Instantiation of Offline Layer	42
3.6	Instantiations of OSM and Google Map and Layer	42
4.1	m-SportGIS MainView.	62
4.2	Instantiation of Toolbar and Save Button.	64
4.3	Instantiations of Location Objects	66
4.4	m-SportGIS Map Instantiation.	67
4.5	m-SportGIS Show Picture Function.	68
4.6	m-SportGIS Show Camera Function.	69
4.7	Submit all data records function	74
B.1	m-SportGIS Loader Function (Ext.Application).	101
B.2	m-SportGIS StartView	102
B.3	Tree List of Mozambique Administrative Divisions.	103
B.4	m-SportGIS MultiList class.	104
B.5	m-SportGIS MultiListOverlay Class.	104
B.6	m-SportGIS MultiListOverlayItem Class.	105
B.7	Validations (rules) in Model.	108
B.8	Facility Controller.	111
B.9	Map Controller.	120
B.10	PictureView Buttons Functions.	126
B.11	m-SportGIS index.html.	127
B.12	Java Statement to Launch Cordova API.	127
B.13	Android Manifest.	128
C.1	Example of TileCache Configuration File.	129
C.2	Header Settings for Map Files.	130
C.3	Map Object Code Fragment in moz_3857.map	131
D.1	Districts ("Distritos") Store.	132

1 Introduction

1.1 Background

Governmental organizations produce critical data about the nation's population, economy, services, and resources. These institutions are coming under increasing pressure, both societal and financial in nature, to develop and implement Information and Communication Technologies (ICTs), particularly Internet-based solutions, inside and surrounding their administrations. These emergent technologies (*e.g.* e-Government, e-Infrastructures or electronic public administration services) are being appraised as priorities in several activities by a majority of governments around the world, supporting a new paradigm of society and modernization, committed to public service, making it more efficient, effective, transparent, more citizen friendly, and more capable of delivering cross-agency public services (*e.g.* storage system and knowledge exchange) in an integrated way (Pollitt, Thiel, & Homburg, 2007; Soares & Amaral, 2011).

In these last years, a remarkable success in the use of handheld computers — smartphones, tablet PCs, Personal Digital Assistants, and notebooks — is being appreciated as appropriate, affordable, and convenient technology for data collection in diverse fields (Casademont et al., 2004). They hold the potential to reduce the logistic burden, cost, and error rate of paper-based methods in data collection works (Seebregts et al., 2009). Nevertheless, there is a lack of appropriate and technical low-cost solutions implemented in the market. The notable continuing growth of the Information Systems (ISs) industry creates opportunities and challenges for interesting software app developments and implementations. This blending encourages new branches of data collection tools. The ability to handy interact with digital data on the field offers important enhancements for data quality and operational efficiencies (S. Nusser and L. Miller and K. Clarke and M. Goodchild, 2003). Improving quality of data by reducing errors in data collection and storage, taking advantage of electronic tools, is essential for the success of any large public project (Yu, de Courten, Pan, Galea, & Pryor, 2009). The continuous development of the Network and distributed computing technologies are evolving Geographic Information Systems (GISs) and the electronic public administration services into a common sphere. GIS attempts to the data capture, display, analysis — data relationships, patterns, and trends — and produce outputs (reports) in the spatial domain. GIS and mapping systems easily store and manipulate spatial objects such as areas/polygons, boundaries/lines, and points, representing real-world features, such as facilities locations, roads, spatial demographic information, and po-

litical boundaries. All of this data can be combined, interrelated, and analyzed using GIS tools (Gaffney, 2002). Geospatial data articulated with other data sources provides efficient means for planning, decision-making, and management of resources and activities in organizations, bringing forward cost savings, process quality, and efficiency increase. It is estimated that more than 80% of the data that population have collected so far constitutes geospatial data; this type of data is largely used in many aspects of socio-economic activities (J. Zhang, Liu, & Wang, 2010). ISs are instruments which assemble, store, process and deliver proper information to an organization or society, in such a way that the information is accessible and useful to those who wish to use it, including managers, staff, clients and citizens (Buckingham, Hirschheim, Land, & Tully, 1986).

Based on the interpretation given by Drucker (1966), where information is data endowed with relevance and purpose, organizations and managers seek to turn data into information to beef up their knowledge and understanding of nation development and sustainability, as well as allowing them to communicate their findings in a way that is appropriate to the nature of the data. Managing public and private resources requires structured information and communication. At the same time, the collaboration and data exchange capacity of different stakeholders is result of the interoperability capacities, strategies, frameworks, and platforms among their ISs. In this sense, Spatial Data Infrastructures (SDIs) play an important role in delivery access to information services in the spatial domain. A crescent number of national mapping agencies are committing an important role in SDI developments. A SDI provides a basis for spatial data discovery, evaluation, and tool for users and providers within all government levels — information is essential from administrative activities, such as facilities and utilities, to education and public health. SDIs are also important as platforms for economic and social development — commercial sector, non-profit sector, academia, and citizens in general (GSDI Association, 2012). In Mozambique, local stakeholders are looking to support their activities using SDIs aspiring to guide good governance, public resources optimization, and public service (*e.g.* health) improvements. A key aspect of electronic governmental services is to bring public administration closer to citizens and enterprises, providing access to governmental information throughout organized access and structured content (Bertot, Jaeger, Simmons, Grimes, & Shuler, 2009). Geospatial Information (GI) integrates properties (attributes, characteristics, variables) with locations and times, within the geographic domain; this data can come out from sources inside and outside an organization and be linked to the GIS. GI is attained from data that is created through some process of production, whether by observation, accurate surveying, cartography, aerial photography, printing, digital copying, or any combination of these (Goodchild, Fu, & Rich, 2007). Humans need GI in order to conduct many forms of human activity as planning, administering, exploiting and managing resources. GIS reinforces the strategy of administrative modernization such as new public management and good governance. They represent a propitious generation of

powerful tools in Public Administration, developed with support of both traditional and new ICTs, contemplating a more extensive and intelligent mode of policy-making, holding a transformation effect on the structures, processes and functions of the public services (Donk & Taylor, 2000). Geospatial technologies, mobile computing, data storage and transfer, and the Internet are continuing drawing effective assets for a wide range of urban planning projects, assuming a relevant role in research and generate resources proceeding from sources of field information. The mobility of the mapping systems has always been a concern. Mapping in a mobile distributed environment has been investigated and developed by researchers and engineers since the late 1980s, encouraged by the availability of location-aware technology for civilian uses (*e.g.* facilities mapping) and concerns about the mobility of mapping systems. Mobile information technology has emerged from a combination of the Internet and wireless communication (J. Drummond, R. Billen, E. João, 2007). Mobile computing systems and hardware are experiencing significant progresses, and they are changing the way GIS is being used by moving GIS, in particular data collection tasks, from the desktop into the user's hands, making the mobility an enticing aspect of GIS (Döner & Yomral, 2008). Mobile mapping technology has evolved to a stage which allows geospatial industries to apply it in order to obtain high flexibility in data acquisition, more and richer information with less time and effort, and high productivity (R. Li, 1997). Handheld computers (*e.g.* tablet PCs) integrating location-based services, digital orientation, wireless communications, and sensing technologies, are useful equipments to gather geospatial data directly into a digital spatial environment, boosting field data collection in a more expeditious and operational way. Tablet PCs also bring the power to process data and disseminate it, including its storage, transfer, and synchronization in databases, and its combination with other data sets (Döner & Yomral, 2008). Others advantages of Mobile GIS are reflected in data integrity by validating it at the time of its registration. Mobile GIS can offer a user-friendly interface to its users (without GIS background), and can reduce transaction costs due to reduced errors at the source, faster communication protocols, and direct position recording. Mobile computing and mapping technologies are being exposed as an revolutionary and low-cost solution for data collection in a range of scientific backgrounds (Aanensen, Huntley, Feil, Al-Own, & Spratt, 2009; Chen & Xiao, 2011; De Donatis & Bruciatelli, 2006; Green & King, 2004; Lane, Miluzzo, Lu, Peebles, & Choudhury, 2010).

Recognizing the potential benefits that the MJD can achieve with improvements in the strategic planning of sports resources and public services, a mobile low-cost mapping platform was develop to handle sports data collection. This resource constitutes the information source of a low-cost governmental SDI (WebGIS). The WebGIS offers to governmental managers and planners tools to visualize, analyze, create, edit, and process sports and statistical information; its construction is not within the scope of the current research work) (R. Li, 1997; J. Zhang et al., 2010).

1.2 Problem Statement and Motivation

Throughout this century there has been a trend towards a decrease in the amount of research into field data collection. In developed countries is comparatively easier to gather data than in developing countries. The lack of rich data (including spatial data) available is more notorious in these last ones due to both, geospatial data infrastructure policies, and high cost of conventional data collection and data processing methods (Montoya, 2003). The existent need of reliable infrastructure, competent tools, ubiquitous connectivity, and proper expertise makes data collection operations very arduous. The ability to handily interact and process raw data enables individuals and organizations to effectively create information crucial to plan and manage services and resources, and make better decisions. Most of the current organizations collect data using traditional methods — paper forms — facing their inefficiency issues, such as data inputs errors, time consuming and inconsistency in data transcription, and long delays until information about that data is available (Anokwa, Hartung, & Brunette, 2009). Consequently, particularly in developing countries, and in the spatial information sphere, when a project specifications require advanced information, organizations actually realize the real importance of carrying on modern, cost effective, and appropriate data collection solutions in field operations (mobile mapping) (Döner & Yomral, 2008; Montoya, 2003). In developing regions, the actual expansion of mobile technology usage, and improvement of technological infrastructure is inciting high motivation to use that technology, expecting to overcome current deficiencies in data availability.

In Mozambique, the current governmental technical situation lacks of a platform to plan and manage sports activities and venues (facilities, infrastructures), and to share relevant information with communities and stakeholders (République de Moçambique, 2002). In this sense, Mozambican sports bodies demand a cost effective solution to gather detailed data including spatial attributes, customized with sports specifications. This mapping solution should meet the needs of MJD in terms of usability, data integrity, maintenance costs, and information details.

This project was focused upon the development and implementation of a standalone location-based software app for real-time field data collection, targeting tablet PCs (or smartphone) running Android operating system. Technicians can work offline inputting data directly on the mobile platform, which is validated by format rules, and stored (primarily) locally. The stored data can be sent to a server-side spatial database using communication techniques. The data in the server geodatabase feeds the sports WebGIS platform. m-SportGIS constitutes an attractive and low-cost solution to the Mozambican Government, in particular to sports stakeholders.

1.3 Objectives

The main goal of this project is to develop a location-aware mobile software app — m-SportGIS — as a vehicle to create an inventory of sports facilities. This data is a crucial source to allow sports data analysis, planning, and management in the MJD. The project research involves analysis and implementation of low-cost resources to build the mobile software solution, targeting tablet PCs (and smartphones) running Android operation system. That solution will be a pioneer tool to the Mozambican Government, capable of collecting, validating, storing, editing, visualizing, and transferring data (alphanumeric and media) processed *in-situ* (field). Afterwards, stored data (in a server database), together with statistical data from National Statistics Office (“Instituto Nacional Estatística” (INE)) of Mozambique, are the data sources of a SDI with spatial functionalities (analysis and edition tools). To face m-SportGIS development life-cycle, and regarding the achievement of a successful product, the research of adequate methodologies and approaches to software development in organizations can be very valuable, since they propose adequate software development standards. To build a handy system and functionalities is necessary:

- research, evaluate and identify the feasible and suitable OS technologies and tools to develop a mobile app (with mobile web technology);
- research on feasible hardware resources requirements;
- research and develop a spatial framework interface and the process of location-based service to work on both conditions, online and offline map service;
- design the app architecture and components;
- create an aesthetic GUI;
- implement a data insertion and validation process, and a data storage method;
- implement a data transfer service technique;
- build a prototyping and testing procedure for the app;
- create an app manual.

1.4 Assumptions

The conception, development and implementation of this project is conducted based on the following assumptions:

- The institution MJD is capable to effectively utilize the acquired knowledge, which is a condition of access to information, and it drives to attain competitive advantage;
- The creation of a sport facilities inventory database afford a bright input for the decision-support system of governmental organizations, specially in the management, strategic planning and dissemination of sport resources and public service;

- Free and Open Source Software (FOSS) technologies provide a low-cost software development solution, namely in what concerns to software licensing costs, and it constitutes the software option to deploy this app;
- The app takes an aesthetic GUI being really intuitive to involved operational-level staff, who probably lack of experience and knowledge in new hardware/mobile equipments and GIS technologies;
- The institution stakeholders own suitable resources/hardware for the surveying works, and system stakeholders have training to make good use of the resources;
- The Internet and mobile wireless broadband technology is available for communications and data exchanges, at least in the government installations.

1.5 General Project Methodology

To conduct the development of the software app, a research on GIS data collection methods and tools was undertaken. Inside this field the use of mobile digital technology and location-based services was studied. Considering these approaches, a research on mobile GIS collection in sports inventory systems was reviewed, and a research on sports spatial planning was undertaken. To implement m-SportGIS using low-cost resources, web, mobile, and spatial development frameworks were analyzed. Regarding Android devices specifications, where the app is being installed, requirements of native functionalities accessed using web based code was also analyzed. For the map interface, both online and offline spatial services and servers were studied; in this field, attending to the offline capabilities, the conception of cache mechanisms were analyzed. Concerning the development and implementation of software in the core of a governmental organization, convenient methodologies in information systems development were examined and an approach was considered to hold the life-cycle of the app. Before the development of m-SportGIS, a market research of current mobile GIS mapping and apps was undertaken. The app development contemplated system requirements analysis, data production, code production, implementation and testing stages. Project is concluded commented final considerations, and further developments.

1.6 Structure

The project comprises five chapters and reflects the investigations and works taken to accomplish the intended goals of the current dissertation work (Figure 1.1).

The first chapter introduces contextualizes the project in holistic terms, namely, the modernization, development, and impact of information and communication technology in organizations, and contextualizes the importance of the implementation of a SDI in the Government of Mozambique, to manage sports resources. In this context, the existent technological problem is stated. Additionally, the motivation to develop

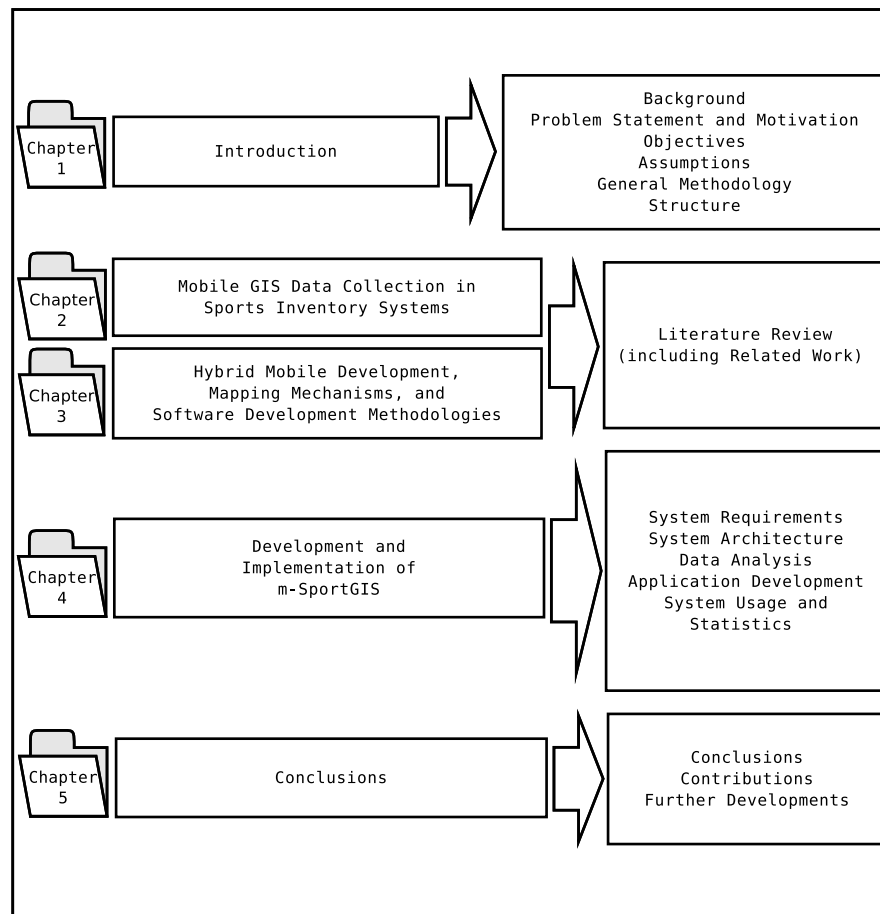


Figure 1.1: Project Structure.

the m-SportGIS solution and its main objectives are enumerated. Furthermore, assumptions for the deployment of the software app are bringing forward. This chapter ends with a presentation of the general work methodology, followed by a delineation of the project structure. In the second chapter, GIS data collection and mobile GIS technology concepts are reviewed, together with its advantages in sports facilities data collection and information management. The chapter ends with a description of the mobile GIS mapping state of the art, enumerating data collection apps in use available in the market. The third chapter explores technologies, frameworks, and mechanisms to build mobile apps and mapping services. Particularly it is presented different mobile software development approaches, and it exhibits different mapping implementations to serve spatial data to the user. This chapter ends with a review of software development methodologies in organizations, focusing on web-based systems contributions to the modernization of the information platform in governmental bodies. Then, it is presented an adopted methodology for m-SportGIS development. The fourth chapter comprises all the development stages undertaken to produce the mobile mapping solution, from system requirements analysis, data preparation and processing, coding, and production. In this chapter it is also presented an overview of the integrated GIS system — mobile component, server component, server database. It also presents sports data statistics in Mozambique, showing the effective impact of m-SportGIS usage inside the organization. The chapter ends with considerations on the app implementation. The last chapter includes project conclusions and contributions regarding the success of the work, presenting advantages and limitations of m-SportGIS, stating also suggestions for further developments.

All citations to references in the thesis text with no source citation were produced by the project's author.

2 Mobile GIS Data Collection in Sports Inventory Systems

2.1 Introduction

Over the last decade, mobile computing systems had made significant progress, and the enhancement of technologies such as the Internet, telecommunications, GPS and mobile computing devices are changing the way GIS is being used, by moving GIS from the desktop into the field user's hands, and making the mobile an enticing aspect of GIS. The potential of mobile GIS and others mobile computing methods to add superior value to support efficient scientific data acquisition and management is widely acknowledged (Döner & Yomral, 2008; Wagtendonk & Jeu, 2007). Mobile-computing technologies in GIS distributed environment are progressively becoming the means for capturing and visualizing electronic spatial data in a variety of technical and socio-economic sectors. GIS products are now used in a large number of apps that deal with spatial data to drive socio-economic planning, marketing analysis, facilities management, and environmental assessment (Longley, P., Maguire, D., Goodchild, M., Rhind, 2011).

Traditional data collection using paper forms and maps meant overlong streamlined work-flow, from data collection, to post-processing (*e.g.* evaluation) to make spatial information available, leading to a low data service life. Too many processes vulnerable to produce errors were implicit, reducing quality and accuracy of data, hindering data quality control (Chen & Xiao, 2011; Yan, Yu, Wu, & Ma, 2009). The need to explore new ways of data collection has been actually a crucial matter.

Mobile computing can combine GIS, GPS, and communication technologies, extending the capabilities of conventional mapping, improving human and capital resources required in data handling and processing (Clegg et al., 2006). Mobile GIS makes possible to bring field and office activities into a collaborative environment (Green & King, 2004; Guadagno et al., 2004). Mobile mapping technology has evolved to a stage which allows mapping and GIS industries to adopt it in order to obtain large flexibility in data acquisition, more information with less time and effort, and high productivity (R. Li, 1997). As more spatial referenced information of all kinds is put directly into GIS, as more these systems are becoming the repositories reference of spatial information, and the information source from where both, hard copy maps and other kinds of cartographic displays, are derived (Dangermond, 1997).

2.2 GIS Data Collection

Data collection is recognized as one of the most time-consuming, expensive and error-given tasks in any GIS project. It is common that technicians and researchers use a mix of survey instruments, maps, and paper-based forms to acquire geospatial and attribute data in geosciences fieldwork (Wagtendonk & Jeu, 2007). Regardless of conventional fieldworks methods have proven to be adequate in different inventory and monitoring operations, a crescent community of scientific researches suggests that the use of hand-held computers, mobile GIS, and related mobile location technology (*e.g.* GPS) has several distinct advantages in comparison to traditional paper and data entry collection techniques, improving process efficiency, integrity, and capacity (Aanensen et al., 2009; Tamelen, 2004; Wagtendonk & Jeu, 2007). This is especially true in developing countries where current collection and analysis of administrative data are rare and limited due to poor infrastructure and organization (Seebregts et al., 2009). Often, data input on paper-based forms is unstructured and stored values are corrupted. This problem induces errors, data quality issues, inconsistencies, and workload burden during data transcription to digital formats.

There are several sources of geographic data, and different methods are available to enter them into a GIS. The two main methods of data collection are data capture (direct data input) and data transfer (input of data from other systems). Mobile mapping software is usually explored for direct input data, but it is able to handle with both methods; it is important to identify the type and source of data produced from mobile devices. Scientific studies and governmental administration require large scale data and associated attributes (R. Li, 1997). This constitutes one more important reason to adopt mobile GIS tools in spatial data collection.

The largest mobile mapping benefits come in the post-processing phase, where all those traditional data methods and techniques are overcome. A Strengths-Weaknesses-Opportunities-Threats (SWOT) analysis of mobile computing methods for fieldworks has been addressed by Wagtendonk and Jeu (2007), taking into account their success and added value in the long term (Tabela 2.1).

In this process of adapting technology to assist planning and management, it becomes essential for technical staff enrolled with the project, understand the reliability of using the technology to collect data, compared with pen and paper methods, and the steps that can be taken in programming mobile computers to aid in the collection of reliable data (Guadagno et al., 2004).

GIS Data Collection Participation: Empowering People

The Internet and the World Wide Web (WWW) were recognized to have the potential to revolutionize this closed world view of GIS by dramatically increasing opportunities for GIS-based apps, through integration of mapping, GIS and non-spatial informa-

Strengths	Opportunities
Data collection efficiency (<i>e.g.</i> smart menus)	Wireless information retrieval
More objective data collection	Real-time streaming of field data
Minimized post-processing time	Monitoring and virtual supervision
Spatial and attribute accuracy	Dynamic changes to data collection strategy
Real-time error control and validation	Seamless integration of post-processing work
Searchable access to field materials	Faster field data distribution (handy GUI)
Wireless integration with other devices	
Weaknesses	Threats
Complex system development (specific skills requirement)	Non well-trained field operatives
Development, implementation, and maintenance costs	New risks (data loss) in its transfer and integration
Vulnerability of equipment	Lack of resources for regular system updates
	Growing technical dependency

Table 2.1: SWOT Analysis of Mobile Computing Methods for Field Mapping (adapted from Wagtendonk and Jeu (2007)).

tion technologies, conceiving new schemes of representation and new ways to address significant questions to society (John Wilson and Stewart Fotheringham, 2008).

Current mobile technology tools such as tablet PCs/smartphones empower potentialities for data collection, extending spatial planning and GIScience to groups of people to participate in technological projects and decisions, contributing to develop their communities and promoting improvement of public services (Devisch & Veestraeten, 2013). Participatory methods are mentioned as an attempt to utilize GIS technology in the context of the needs and capabilities of communities that will be involved with, and affected by development projects and programs (Kienberger, 2001).

2.3 Mobile GIS Technology

We live in a networked society founded upon modern ICT linking computers with computers, computers with individuals and between individuals on an unprecedented scale. Mobility is a revolutionary phenomena that has transformed the way we work, perceive, and access information. The most important current trend in the software development world is, without a hint of a doubt, the mobile frontier, and in particular, the rise of the smartphone and the touch tablet PC (Kosmaczewski, 2012). Increasingly powerful mobile devices, their ability to be location-handled, mobile computing, and the rapid growth of high speed networks, are expanding the possibility of employing information and their content based on the location, to support organizations establishing services and products for the information community.

Mobile computing technologies, like handheld computers (*e.g.* tablet PCs) integrating location-based services, digital orientation, wireless communications, and sensing technologies, are being essential tools to collect geographical data directly into a digital spatial environment, boosting field data collection in a more expeditious and operational way. Consequently, data processing and availability is improved, including its transfer and storage in databases, and its combination with other data sets (Döner &

Yomral, 2008). Others advantages of Mobile GIS are reflected in data accuracy through the validation at the time of data registration, reduction in transaction costs due to errors minimization at the source and faster communication protocols, direct position recording, and user-friendly solution for the field workers.

Goodchild in Longley, P., Maguire, D., Goodchild, M., Rhind (2011) exposes an interesting perspective on the implications of GIS becoming more distributed, based upon four different locations: the location of the GIS user and user interface (UI), denoted by U ; the location of the data being accessed by the user denoted by D ; the location of data processing, denoted by P ; the area/field site that is the focus of a GIS project, denoted by S . Traditionally, in GIS projects $U = D = P \neq S$, this is, the UI, the data, and the data processing all occur at the same location, this means they occur at a laboratory rather than at a field site. In the new era of distributed and mobile GIS, it is possible for $U \neq D \neq P = S$, that is, the UI, the data, and the data processing can be materialized at different locations, and some or all of them can take place at the field.

As the technology has become more affordable, portable, and easier to use, together with innovations in hardware (programmable mobile phones) and software, the smart-phone has lead as the universal mobile telecommunication computing technology, world-wide adopted by a wider community of professional users (*e.g.* social science researchers), as a tool for collecting data, and accessing information on the move (R. Kwok, 2009; Raento, Oulasvirta, & Eagle, 2009). As a reflection of advances in mobile technology, mapping industry has been developing mobile GIS software (*e.g.* GeoCollector for ESRI ArcPad system) to simplify capturing, editing, and displaying location-based data more conveniently. In the recent years, significant advances have taken place in GPS technology for the compact portable electronic containers such as tablet PC, specially in terms of integrated positioning, navigation, and timing. The ability of tablet PCs to run rich mapping software apps, and to position themselves, allow field technicians to capture spatial information in real-time and real-space.

2.3.1 Tablet PC and Smartphone Technology

Tablet PCs and smartphones are rapidly becoming the central computer and communication device in people's lives, and are revolutionizing many sectors of our economy, including business, environmental monitoring, transportation, health-care, and governmental activities. Tablet devices have been around for more than a decade, but it grabbed Apple's iPad to reestablish the category to the world. Apple sold out more than 17 million iPads from March through December of 2010 (David, 2012). According to Firtman (2013) and Lane et al. (2010) a well-off smartphone is a container owing a multitasking OS, a full desktop browser, Wireless LAN (WLAN/WiFi) and 3G CDMA

¹/4G LTE² connection support, a music player, and several others features such as GPS antenna, digital compass, digital camera, microphone, accelerometer, gyroscope (sensors of motion, vibration, and orientation detection), and touch support. These kind of embedded features make smartphones capable handheld devices able to process geospatial information. The same author defines a tablet PC as a touchscreen flat device (7 to 11 inches size range) running a mobile operating system (such as i-OS instead of Mac-OS), or a touch-optimized version of a desktop operating system (such as Windows for tablets instead of Windows Phone), holding WiFi support and/or 3G/4G connections. These devices (later to the iPhone 2007) are open and programmable computers affordable by third party developers that can make use of cross-compile code and existing software such as JavaScript (JS) libraries (*e.g.* Ext JS, jQuery, Dojo) to support apps development. The hardware is controlled by powerful operating systems (*e.g.* Android, Apple iOS, Mobile Windows) supporting the apps community with Software Development Kits (SDKs), APIs, and software tools.

Mobile devices are much more efficient in the form factor (size, shape, weight, *etc.*), chip type, internal storage capacity, battery lifetime and operating system compared to conventional computers. Along with hardware development, the operating systems running on these devices are becoming more compact and functional (Lane et al., 2010). Thanks to them and to the sensing, positioning, computing, and communication functionalities incorporated in tablet PCs, we are breathing an emergence boost of personal, group, and community-scale geospatial apps (Lwin, 2011).

Economic factors around location services and products and the versatility (programmability) of tablet PCs are greatly enhanced by the availability of OS code for software developers to produce and deploy their own apps directly on the devices. OS technologies concede the use of their source code for the development of proprietary and/or free software apps. OS manufacturers like Android/Open Handset Alliance³ provide open and free SDKs to user community to develop, modify and distribute custom software apps to be run on the mobile platform. In addition to the variety of data — text, photos, location, audio, video, barcode scans — that can be collected, mobile platforms have proven to be definitely faster at both, collecting the data and making it available to project planners and decision makers (Anokwa et al., 2009).

Mobile Communication

People lean to use their mobile devices just about everywhere, however mobile networks are not always there to support them, even when they are, coverage can be expensive and sporadic. Developing for mobile means designing for this reality. Despite the fact that tablet PCs are evolving in sophistication, and developments within mobile and

¹CDMA: Code Division Multiple Access standard

²LTE: Long Term Evolution standard

³openhandsalliance.com

web technologies provide the opportunity to expand approaches in new data collection, visualization and analysis, it is presumable to notice hardware resources constraints and challenges when developing embedded features in those equipments. It is important to focus on defining specific mechanisms and use whichever client and server-side technologies available to improve the efficiency of data processing, taking into account that typically mobile devices have to rely on communication technologies, namely user positioning system and wireless connection for data interchange/interoperability (Sumrada, 2002; Wroblewski, 2011). Sending and receiving data using wireless Internet connections (3G/4G/Wi-Fi) introduces a variable amount of latency due to the nature of network nodes/points (stationary or dynamic stage) that can be very unpredictable. Mobile devices typically know intermittent connectivity to the network, much more evident in developing countries, where the services offered on a mobile device may only be accessible occasionally. Therefore any client-side app (*e.g.* m-SportGIS) that interacts with communication services on mobile platforms needs to be designed to operate in an opportunistic manner, leveraging such services when they become available (Berger, Mcfaddin, Narayanaswami, & Raghunath, 2003).

Positioning: Global Positioning System (GPS)

The Global Positioning System (GPS) is a satellite-based navigation system, based on a receiver/sensor that provides accurate location, speed and time to users, anywhere in the world, and under any weather condition. This technology is commonly employed during data collecting field works to determinate, in real-time, the current geographical information in a spatial reference system (SRS). The SRS defines a specific map projection and transformations between different SRS; within the GIS sphere, this constitutes a crucial feature since all different data must be manipulated under the same SRS.

The position (object coordinates) received in the GPS during mobile mapping is one of the most important data quality measures criteria (R. Li, 1997). Improvements in the quality of recorded data using mobile technology are a consequence of new advances in the system location accuracy integration. Since reliable location information has become a cornerstone of many mobile apps, those advances mean a lot to location-based software. Popular GPS device manufacturers announce that one can expect that the data produced by general GPS units, without enhanced features assistance, has an inherent accuracy in the range of 6-12 meters (Green & King, 2004; Montoya, 2003).

Alternatively to GPS technology, tablet PCs provide seamless integration with Wi-Fi positioning and cellular network positioning. Wi-Fi positioning uses terrestrial based Wi-Fi access points to determine the device's location, and cellular positioning relies on cell service areas, each of these has a known location base station which associates its position to the mobile device's location (cell identification method). The combination of the three positioning modes is denominated by hybrid positioning (Zandbergen, 2009). According to this author, smartphones take advantage of that hybrid positioning

system, achieving through its usage position accuracy averages of 10 m (GPS), 30 m (Wi-Fi), and 500 m (cellular). Most tablet PCs and smartphones models are GPS-enabled, making them handy mapping tools in several GIS-based projects.

2.4 Spatial Planning for Sports Venues Management

MJD sets clear expectations regarding country sports planning by “providing the right facilities in the right places”. “Spatial planning goes beyond traditional and use planning to bring together and integrate policies and programmes, which influence the nature of places and how they function” (Sport England, 2005). Spatial planning states for the physical aspects of location and land use, but also factors that make places attractive to live in: health provision, public services, employment opportunities, environmental quality, crime prevention, *etc.*, and a wider, more inclusive approach to decision-making (Figure 2.1).

Citizen access to sports education and activities is a right established in the Mozambique constitution. The search of opportunities to participate in sports infrastructures and facilitates sustainable development is recognized as fundamental to the health and well-being of communities. MJD, working together with partners, seeks to engage planners and technicians at regional and local levels, to ensure that interests relying on sports and its recreations are well represented, promoted, and developed, attending to the interests of all, including the aspirations of its partners in public, private and voluntary sectors (República de Moçambique, 2002).

MJD, alike others developed sports organizations (*e.g.* English Sports Council⁴), aims to ensure a good spatial planning for sports venues (facilities, infrastructures) based on prosperous assessments of needs for all levels of sports and all sectors of the community. This analysis allows to identify opportunities to plan new resources and services that meet the current and future demands to sport participation (England, 2013). Spatial planning involves partnership through the development of shared aspirations, and a willingness to make creative use of land use planning in combination with a range of other policy instruments, such as community strategies.

The commitment view of these organizations intend, in addition to hold a sports system, provide organizational, planning, and regulation measures (legislation), promote sports evolution, ensure the existence (and good management) of sports venues of quality, improve quality and performance of top-level sports, and promote intra and inter-institutional communication and cooperation. Tabela 2.2 shows registered outcomes of implemented sports planning projects in socio-economic domain.

m-SportGIS constitutes the primary tool to support the spatial planning system. According to the German Federal Institute of Sport Science — BISp⁵ — and Cadima,

⁴sportengland.org

⁵bisp.de

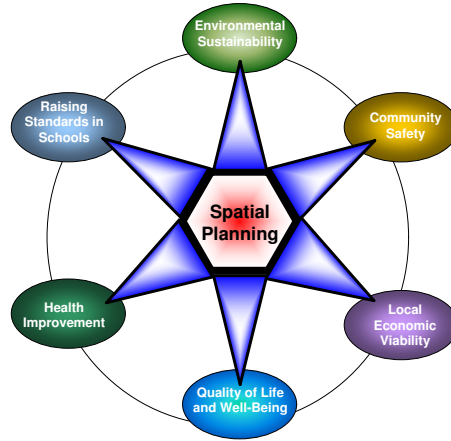


Figure 2.1: Principles of sports spatial planning (Sport England (2005)).

Themes for Spatial Planning	Outcomes of Sport and Active Recreation Innovations
Environmental sustainability	Increasing participation on sport and active recreation
Community safety	Improving levels of performance
Local economic viability	Widening access (new opportunities for people)
Improving quality of life and well-being	Improving health, and social and cultural well-being
Health improvement	Stronger and safer communities
Raising standards in schools	Improving education
	Benefiting the economy (sustainable growth)
	Efficient use of resources

Table 2.2: Sport framework: spatial planning activity and outcomes of innovation (adapted from England (2013); Sport England (2005)).

Fernandes, Viseu, Parente, and Brito (2002), an essential duty in the management of installations, sport grounds, and recreational areas leads to research and inventory useful data about the current and intended type and number of sport venues. Further, is also important to take into consideration policies on sports practice trends of population, besides the provision of services by clubs, organizations, Olympic Committee, and others public and private institutions (Répubblica de Moçambique, 2002). The BISp also notice that the sports practice is only experienced if there are available installations and appropriate sports grounds conditions.

One of the pieces of strategic planning conveys to inventory information about what offer and type of installations and sports grounds are current available to the community, how many are, its functionality and architecture, type of equipment/material, and location (Figure 2.2). Taking into account the goals of sports planning, and the range of sports activities, the adoption of a database where sport activities identifying relevant specifications and sports subjects are embedded, constitute a crucial resource to analyze sports as public services (Cadima et al., 2002).

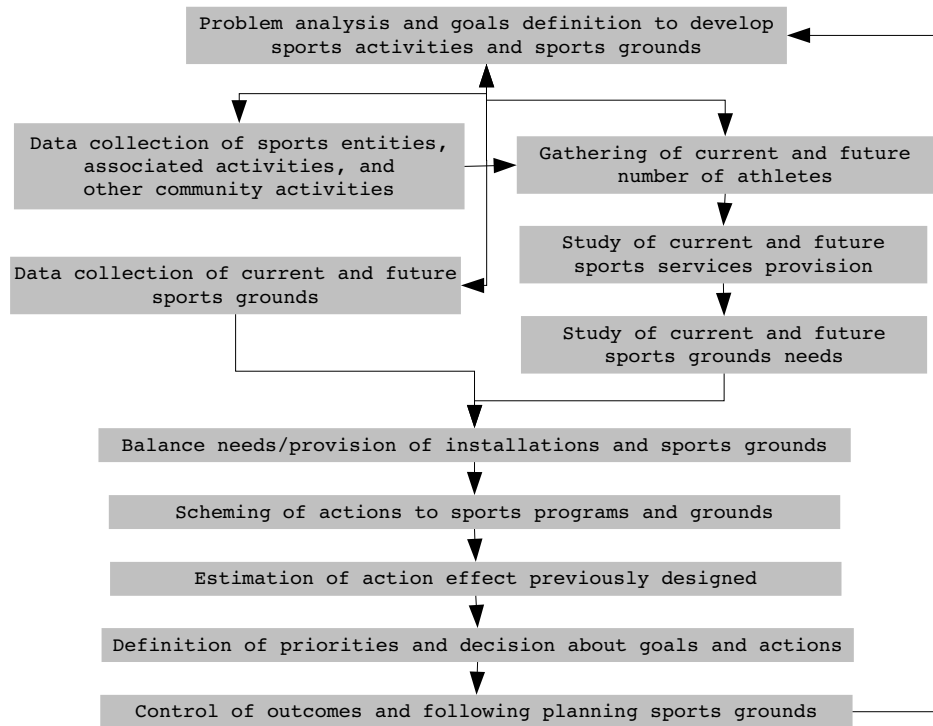


Figure 2.2: Guidelines for sports grounds planning (adapted from Sportbuchhandlung (2000)).

2.4.1 Projects in Sports Data Collection

In a research study about citizen access to sports activities in Portugal, Cunha (1997) mentioned that the sports domain is defined based on criterion, such as the identification of particular goals that leads to agents' actions and deal with the processes and activities, the existence of an organizational structure where sports actions are planned and practice, the presence of agents as human resources and athletes to manage processes and take them into action, the recognition of the target population depending on the way they live the sports sphere, and the statement of the organizational processes associated with an established board of activities.

To build a very complete sports venues attributes inventory for the MJD, it became necessary investigate what sort of specifications sports agencies consider relevant to embody in a field census report. Analyzing some sources about projects carried out by Portuguese organizations, such as INATEL⁶ and Porto de Mós County⁷ (Instituto Português do Desporto e Juventude, 2012), and information about the African Olympic Games (Maputo, 2011)⁸, made possible to create a list of sports venues characteristics. This information was the base to develop a structure of sports data in m-SportGIS

⁶www.inatel.pt/content.aspx?menuid=21 (accessed on 22 October 2012).

⁷www.municipio-portodemos.pt/page.aspx?id=24 (accessed on 30 October 2012).

⁸en.wikipedia.org/wiki/2011All-AfricaGames (accessed on 24 October 2012).

(Tabela A.4).

Recreational Areas

According to Cunha (1997), sports industry legislation classifies the facilities type based on its sports activities as recreational, educational, specialized (for competition), and specific for sports entertainment. The first type of facilities intend to shelter sports activities as leisure practices; the second consist of infrastructures conceived to organize sports education trainings (*e.g.* large football field, sports hall, swimming pool) attending to reach competition levels; the third type of facilities coordinate a selection of specialized activities (*e.g.* golf field, canoing site); and the fourth type are special facilities that associate specialized activities with entertainment components (*e.g.* stadium seating, social media sites, and technology).

2.5 Mobile GIS Mapping: State of the Art (Related Work)

GIS mapping related work states that, among the most popular advantages of using mobile GIS mapping, are benefits in data acquisition agility, consistency of stored data, reduced data error along the handling process, reduced data transfer times, and better data share among stakeholders. Fieldwork is the basis activity of collecting location-specific data, and by mobilizing the electronic acquisition, the inspection and verification of collected data can be carried out at the same time as it is being observed. Advantages of modern mobile mapping have been addressed for electronic geospatial data collection in diverse fields, yet, its implementations are still in an early stage. Low-cost solutions combining an intuitive customized and rich form with mapping are not well-explored yet (Wagtendonk & Jeu, 2007). Cases of mobile GIS are noticed in biology (*e.g.* epidemiology and ecology) and environment monitoring; researchers and public health laboratories physicians employed handsets to inventory and manage data such as soil contaminated sites, plant and animal species, using data forms, being submitted directly into databases or statistical packages through Internet service, which further could be linked to geographic information, properly performing spatial analysis and planning (Aanensen et al., 2009). Field workers in coastal zone management studies in England take spatial data acquisition including media information (digital photography to provide record of the study area) to determine shoreline movement (erosion) over time (Green & King, 2004; Kim et al., 2013). In United States, Environmental Protection Agency, and California Environmental Protection Agency worked on the construction of air quality management systems to analyze emissions of air pollutants and greenhouse gases, supported by air pollution apps, which combine GIS field tools and Internet to produce GIS-based emission inventories (Gaffney, 2002; Gkatzoflias, Mellios, & Samaras, 2013). Geological field mapping (information of stratum, rock, and structure) is crucial for geologists and geoscience users, such as petroleum and

mining industries, to provide basic geological materials (information from accurate raw data) for territorial planning, mineral exploration, engineering geology, *etc.* (Chen & Xiao, 2011; De Donatis & Bruciatelli, 2006; Jones, McCaffrey, Wilson, & Holdsworth, 2004). In forest management inventory, where conventional data surveying and processing methods need high-intensity labor force, but with low accuracy due to large amounts of data involved, forestry science workers pay close attention to mobile mapping, which shows remote sensing images and vector maps on the device, and displays the current position of the user in the vector and grid map. Mobile GIS supports a modern electronic governmental forestry service — e-forestry — integrating automatic mapping and warehousing (Dong, Liu, Wang, & Chen, 2010; C.-g. Li & Jiang, 2011). In the context of participatory planning projects (*e.g.* National Spatial Data Infrastructure), US citizen communities employ smartphones to register, mapping, and tracking almost anything that is conceivable with importance for location analysis; smartphones are perceived as “networked mobile personal measurement instruments”. Projects relying on this technology are thus referred to as second-generation “citizen science” or m-citizen science (Devisch & Veestraeten, 2013).

2.5.1 Employment of Handheld Devices in Mapping

Scientific research investigating the use of tablet PC/smartphone technology to gather field data collection is relatively novel. Digital surveying of facilities is not known yet. In this sense, the research has been taken analyzing the use of mobile mapping in different backgrounds as mentioned before. An investigation about mapping apps for smartphones disclosed examples such as EpiCollect⁹, and Magpi¹⁰, in environmental studies. The IBM Almaden Research Center has designed an iPhone app — Creek Watch¹¹ — to monitor watersheds, where users with the phone’s GPS enabled, take a photo and submit crucial pieces of data based on observations (IBM, 2010). The Center for Embedded Networked Sensing at the University of California, has also developed an Android app to locate invasive species in Santa Monica mountains, by making geo-tagged observations and taking photos, to study the spread of habitat-destroying by invasive plants and animals (What’s Invasive¹²) (The Center for Embedded Networked Sensing, 2010). Computer scientists at the University of Washington have launched an Android app — Ultra Mobile Field GIS — to acquire georeferenced data of public facilities (*e.g.* bicycle stands and their capacity, car parking lots, and illegal garbage dumps) to assist administrators in maintaining the campus landscape and managing the student facilities (Lwin, 2011). Docking special sensors to smartphones have also been conducted to run environment-centric apps (*e.g.* Haze Watch¹³) in Australian urban

⁹epicollect.net

¹⁰magpi.com/login/auth

¹¹creekwatch.researchlabs.ibm.com

¹²whatsinvasive.org

¹³pollution.ee.unsw.edu.au

spaces; these “mobile stations” afford air quality monitoring, collecting environmental parameters as geodata related to air quality or noise pollution. Mobile stations collect less accurate measurements than settled stations, yet are able to recognize mobile unpredictable events (Kanhare, 2011).

3 Hybrid Mobile Development, Mapping Mechanisms, and Software Development Methodologies

3.1 Hybrid Mobile Development

3.1.1 Thinking Mobile

Mobile web development is heating up fast. The biggest change in web development over the last recent years has been the remarkable rise of mobile computing (Zakas, 2013). “The scenario relies on a collection of technologies, already years in development, that are opening its way into the mainstream of computing. HTML5 and its introduction to mobile devices is the geeky umbrella term for this assemblage. A glaring element of the HTML5 specification is the strong focus in apps. Engineers argue the technology makes possible to write web apps, accessed with a browser, that are as visually rich and lively as the so-called native apps, designing to run on mobile devices, such as iPad and Android-based tablet PCs” (Lohr, 2011). The Web has been shifting toward focusing on mobile platforms, bringing a whole new world of possibilities, and promising mobile web-based advances and better quality of life for developers devoted to HTML5 implementations. Developers can create a much more faithful user experience (UX) by taking over the entire screen and engage the user in their mobile web apps world. Coding mobile web apps to reach all platforms and all browsers can be a heavy undertaking, since the new mobile web adds a special support dimension to the apps programmers create. These constraints create new opportunities and commitment to innovation as developers face spread approaches among actual platform’s SDKs. There are different tools, build systems, significant APIs’ fragmentation, and devices with particular capabilities for each platform. In fact, the only thing these operating systems have in common is that they all integrate a mobile browser that is accessible programmatically from the native code (Charland & Leroux, 2011). One of the advantages of this recent trend of mobile devices is the availability of feature-rich and fully standards-compliant mobile web browsers (*e.g., Chrome, Safari*), supporting most of the current standards like HTML5, CSS, JS, and others standard technologies. Browsers’ providers are experiencing the implications and opportunities provided by a standardized, distributed, and simplified app development framework, such as offline

storage, network connectivity, multimedia, sockets and threads, drawing and animation, and advanced form controls (Kosmaczewski, 2012). The growth in power of the mobile web is bringing innumerable possibilities, particularly in developed countries where smartphones/tablet PCs constitute an economical way to access and collaborate with online data and services.

3.1.1.1 Designing Mobile Content

The power of the apps comes from people's ability to not only view and consume content, but to contribute and create content as well; input on mobile is just as important as output. Managing data and displaying it on a screen employing UI framework widgets constitutes a prime way to users interact with data. A mobile user has a different context than a desktop user; field sets on mobile platform come with their own set of constraints and capabilities that delineate how they should be conceived. Working with UI objects for this platform requires special considerations regarding their layout due to the restricted screen size, and how much data inputs the app needs to handle; for instance, design how nested objects must be rendered to be accommodated in a touch-capable mobile container, where a virtual keyboard pops-up and takes part of the room available on screen. Developing forms represents a common task in every app developer's life. Create forms can be a dull subject, both for the user, who is filling them out, and for the developer, who creates them. Depending on the type of field and data required, different types of input fields for mobile browsers can be used. The mobile device serves as a blank canvas for interactive elements; optimizing mobile labels means take advantage of new input types, attributes, and masks to make mobile input easier than ever, choosing handy layouts for sequential, non-linear, and in-context forms, and tuning input field using mobile device capabilities (Wroblewski, 2011). Considering how touch gestures are used to navigate and interact with objects and screens promotes new ways to create content affording natural user interactions, and therefore appropriate mobile on-hover menus and transitions (Firtman, 2013; Pilgrim, 2010; Williams, 2012).

3.1.2 Mobile Applications: Native vs. Web vs. Hybrid

Mobile devices are today's most personal computer; they can do more than traditional PCs. These devices give us outstanding aptitudes through the sensors within them (Luke Wroblewski, 2013). Apple shifted the mobile channel with the release of the iPhone (2007), and our expectations for mobile experiences were completely burnished (Charland & Leroux, 2011). However, life would be so much simpler if IT community could build and deploy apps once and make them instantly available on all devices and operating systems. These natural constraints of mobile devices, networks, and usage patterns help mobile web developers focus and simplify mobile experiences. The mobile channel leads us questions such as which code approach — native *vs.* web *vs.* hybrid — can be appropriated, which testing platforms have developers available,

which libraries can they use to speed up the development, and what level of standard support holds the major mobile browsers (Kosmaczewski, 2012). Building mobile web apps offers us browser-based apps and advanced features consigned to full-screen or native web apps (*e.g.* Wikipedia app), using native web APIs such as Cordova¹ (*aka.* PhoneGap²) (Firtman, 2013). An API works as an interface for developer's code, including commands (working as methods) rendering the more complex stuff that is done in the background or by third-party software. The deciding factors regarding one or another development approach are, ultimately, the user's requirements and expectations from the mobile solution, developer's capabilities, project specifications, available tools, administrative and security concerns, budget, time-lines, and long-term goals.

3.1.2.1 Native Applications

Native code-based apps run “natively granting the best access to the capabilities of the device (*e.g.*, camera, file system, accelerometer) — handling the actual UI behavior — assuring the best UX (*e.g.*, motion gestures recognition), and mobile web apps (browsers) do not. Native mobile apps respond 10-15% times faster than web apps, offer higher security, and are more customizable. Table Tabela 3.1 shows pros and cons of native app implementation. Nevertheless, based on mobile software development experiences, programmers cannot reuse the UX — analysis, design, implementation, deployment — from a specific device when they are creating an app for another. Operating systems — iOS (App Store), BlackBerry (App World), and Android (Google Play Store) — require skills of expert programmers in a specific mobile vendor SDKs and IDEs to deal with its own conventions which its users expect. When the IT team is deploying an enterprise native mobile app for one platform — distribution/installation through mobile app stores (*e.g.*, Google Play, App Store, App World) — chances are it will not be able to create one for every platform, due to the cost of maintaining them — Apple's iOS demands Objective C; Google's Android relies on Java; Microsoft's Windows Phone 7 requires Silverlight; Samsung's Bada needs C++; Blackberry uses Java, WebWorks, and Adobe Air solutions (Fowler, 2012; Williams, 2012). The performance judgment that native apps are faster can take relevance when they require large amounts of graphics processing, Central Processing Unit (CPU), and Random-Access Memory (RAM) (*e.g.*, 3D games or image-processing), but there is a negligible or unnoticeable performance penalty in a well-built business app using web technology (Charland & Leroux, 2011).

3.1.2.2 Web Applications

The mobile web, from a user's perspective, falls essentially in web content accessed from a mobile device (mobile-optimized web browsers); from a developer's perspective,

¹cordova.apache.org

²phonegap.com

Pros	Cons
Full access to mobile device features	Longer development cycle
Deep integration capabilities	Non-portable code
Deeper security implementations	Expensive development and support
Highly responsive and smooth performance	Accessibility via app stores (control, restriction, submission and update approval cycle)
Rich UX / Performance	Needs skilled developers of native code for every mobile OS
Easy installation and payments	
Superior customization	

Table 3.1: Native mobile app: pros and cons (source: Boston Technology Corporation (2013)).

the mobile web it is a group of best practices, design patterns, and new code (Firtman, 2013). iPhone’s launch (2007) brought forward the best way to experience the mobile web due to the Mobile Safari built on top of a WebKit operating system platform (David, 2012). Web technology communities such as Mozilla Foundation³ have been changing the restrictive native apps abilities in terms of hardware features access, and addressing the balance in favor of web standards. Tabela 3.2 depicts the main pros and cons of web app development. Contrasting with the web apps of yesterday which were not tied to the operating system and capabilities of hardware, developers are now required to create apps to be compatible across browsers, platforms, and devices (*e.g.*, Android’s *WebKit*-based browser). WebKit is a lightweight, yet powerful, rendering engine browser, emerged out of the KDE project’s HTML layout engine KHTML⁴ and JS engine KJS⁵, and further implemented by Apple in Mac OS X with the help of an adapter library and renamed WebCore and JavaScriptCore (*Ubuntu*, 2008). Its explosion comes from the fact that it powers many of the leading mobile platform browsers (*e.g.*, Chrome, Dolphin, Safari, Opera). The core feature of WebKit is its support to HTML5, adapting the latest W3C specifications as they are published (Hales, 2013). As of Mid-2013 its market share^{6,7} is evidently in the edge of both engines used by Internet Explorer (IE) and Firefox (Trident and Gecko, respectively). Respecting to current most popular and flexible mobile operating system — Android — and bringing forward its native APIs available to web developers, through JS APIs, the Web community can potentially reach into the same APIs available to native app developers (Williams, 2012).

HTML5, CSS3, and related cutting-edge technologies are very well supported in many

³mozilla.org

⁴en.wikipedia.org/wiki/KHTML

⁵[en.wikipedia.org/wiki/KJS\(KDE\)](http://en.wikipedia.org/wiki/KJS(KDE))

⁶gs.statcounter.com/#browser-ww-monthly-201210-201310

⁷netmarketshare.com/browser-market-share.aspx?qprid=0&qpcustomid=1

mobile web browsers⁸. Technical capabilities like location detection, device orientation, offline storage, forms, and touch events are available on many mobile web browsers today. Figure 3.1 presents a capability scope comparison between web and native apps. These features enforce programmers to definitely act about how users are handling apps throughout their activities. An advanced mobile web app can play a really interesting role as a portable platform; device manufacturers do the work of carrying on the web platform to their device, ensuring a good UX that is consistent with other web apps on that device.

Pros	Cons
Lightweight solution / Powerful technology	Slower performance than native solutions
Reusable and scalable code	Limited offline functioning
Web developer expertise	Limited multimedia support
Inexpensive coding and deployment	Cannot leverage all native OS features
Can be freely accessed / Vendor independence	Cannot replicate native touch-and-feel
Shorter time-to-market (develop, distribute and update)	Mobile browser incompatibility
	Inconsistent rendering across browsers

Table 3.2: Mobile Web App: Pros and Cons (source: Boston Technology Corporation (2013)).

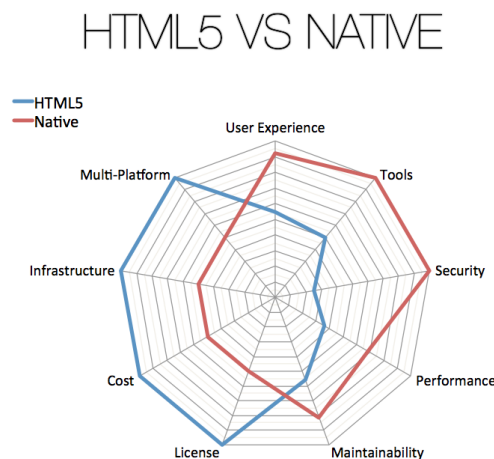


Figure 3.1: HTML5 vs. Native features scope (icapps.com/html5-vs-native, accessed on 4 October 2013).

⁸HTML5 compatibility on mobile browsers: mobilehtml5.org

3.1.2.3 Hybrid Applications

A Hybrid app is part native app, part web app (*aka.* native web app). It is effectively a web app, but hosted inside a native app container (full-screen WebView). The app is developed with web technologies — HTML5, CSS and JS — and packaged and compiled as a native app. This blending makes it mobile platform independent (Figure 3.2). It is installed, launched and used like any native app. This approach relies on native-to-the-device capabilities offer by web-to-native wrappers such as Cordova/PhoneGap which communicate between JS and native code, either one way or bidirectionally (Boston Technology Corporation, 2014; Firtman, 2013).

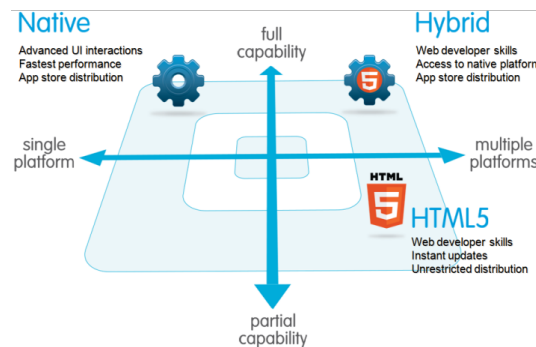


Figure 3.2: Hybrid Capability = Native + HTML5 Aptitudes
(wiki.developerforce.com/page/Wiki, accessed on 27 September 2013).

Going Native

Native mobile apps can be built taking advantage of native APIs, such as Cordova / PhoneGap, capable of delivery native apps for multiple platforms. Cordova has been developed by Adobe and Nitobi, under Apache License Version 2.0, and allows developers to create native-like apps using web technologies. It provides a JS-based API with device-specific native backing code (cordova.js and cordova.jar) that works in parallel with the Software Development Kit (SDK) of the target platform (*e.g.*, Android SDK Tools⁹, iOS SDK¹⁰), granting to mobile web developers access to device built-in functionalities (*e.g.*, media/camera, local storage, file system) from the client JS code Apache Cordova (2013). Figure 3.3 depicts a native web app built upon its structure. Often, organizations build hybrid apps looking to get a presence in the app market without spending significant resources and costs. “About 70-80% of our code can be reused across apps. That saves us a lot of time and a lot of skill. It is very hard to find different native skills and combine them in one team, but it is comparatively easy to educate people in JS (Appcelerator, 2013; Williams, 2012). Setting up a software development environment, incorporating a UI framework such as ST¹¹ or jQuery Mobile

⁹developer.android.com/sdk/index.html

¹⁰developer.apple.com/library/ios

¹¹sencha.com/products/touch

(jQuery Mobile)¹², and a native API, a native-like app can be developed with web technologies (HTML, CSS, and JS) (Figure 3.4). Taking these tools, the same code components can be reused on different mobile operating systems (iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada, and Symbian) with few minor changes, reducing the development costs (Budiu, 2013; Cowart, 2013). Others popular mobile cross platform tools are Titanium¹³ and Xamarin¹⁴ (Wayner, 2013). Attending to the hardware held by the project client, the Integrated Development Environment (IDE) was set up focused on the Android Cordova branch.

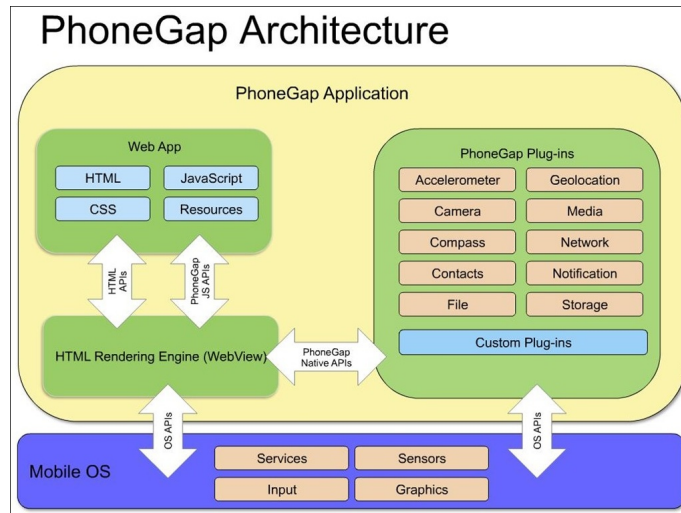


Figure 3.3: PhoneGap Architecture (source: slideshare.net/loianeg/corso-phonedgap-cordova-aula-01-introduco-ao-phonedgap)

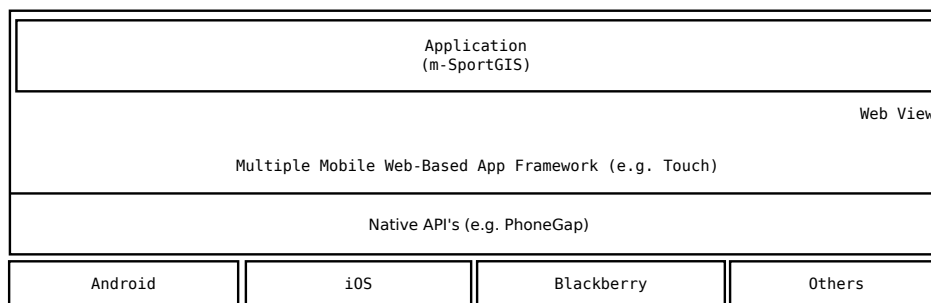


Figure 3.4: Mobile Web-based App Framework Structure (adapted from Williams (2012)).

3.1.3 Designing JavaScript-Driven Mobile Applications

Mobile JS-driven apps refer to browser-based software created for handheld devices which can be connected wirelessly. Mobile devices hold relevant differences between

¹² jquerymobile.com

¹³ appcelerator.com/titanium

¹⁴ xamarin.com

each other in terms of performance characteristics. The success of any mobile web app depends on two factors: design and performance. Web-based standards assist well mobile apps relying on dynamic data, as it can use Asynchronous JavaScript and XML (Ajax) to quickly deliver content to the app over the network. It can also cache and store data to be used offline when there is no network connectivity (Williams, 2012). Improvements in the app’s performance rely on offline capabilities, animations on the UI, and backend services that retrieve and send data via RESTful or WebSocket endpoints (Hales, 2013). Minimizing network connections and the need to transmit data — efficient media handling, effective caching, and employing longer data-oriented operations with fewer connections — is undoubtedly one of the best ways to improve mobile performance (Kosmaczewski, 2012). Leverage Web Storage¹⁵ (*aka.* Local Storage or Document Object Model (DOM) Storage) to store objects (*i.e.* HTML, CSS, and images) in the background can yield an alternative and better UX — twofold conception: the more that can be done on the client-side, the fewer trips across the network and the better UX provided (Google Developers, 2013c; Matsudaira, 2013; Sencha Inc., 2011; Zakas, 2013).

JS engines on mobile browsers (*e.g.*, WebKit) focus on the idea of sending the “right” data across the network, keeping code small both in size and execution time on mobile devices regarding performance cost (Google Developers, 2012; Matsudaira, 2013). As concerns the UI and hardware-acceleration, CSS code should be preferred for graphics presentation and effects, since it can be handled by graphics processing unit (GPU) of the device (Kosmaczewski, 2012). ST libraries offer the best UX on iOS and high-powered Android devices (Garcia et al., 2013). Hales (2013) stats that projects like jQuery are enhancing JS-driven apps, since developers are improving web standards and specifications, and device providers are coming up on implementing them, inciting the OS community to participate in those implementations.

3.1.3.1 Client-Side Architecture

Nowadays, web technology client-side development requires more attention and investment in the architecture of HTML-driven apps. Developers have never been more concerned with performance in the client-side platform. Current web browsers work as client-side platforms, and apps run inside of them. Front-end code is now packaged in native apps, extensions, and operating systems that are all driven by HTML5. Before HTML5, the architecture of advanced web apps was mostly managed on the server. Current browsers are much more powerful, including support to all kind of interesting data. The technology that drives the UI is evolving as browsers become more mobile and allow more hardware access through JS APIs (*e.g.* Cordova). Both UI client and server domains hold data that requires to be presented and data that needs to be collected. Today, the dependence on the server can be negligible, since intermittent

¹⁵w3.org/TR/webstorage

RESTful or WebSocket connections are associated to the UI through a client-side/JS framework. The architecture of client-side apps is still in an elementary stage, but tools and processes to organize code and app structure are getting better over time. JS is definitely considered the assembly language of the current Web-based projects (Hales, 2013; Zakas, 2011). HTML5, the Open Web, and mobile devices have helped push the browser-as-a-platform forward, giving browsers the capabilities of storing data and running applications in an offline state.

Storage on the Client Device

Client-side storage relies on data passed to the browser's storage API, which stores it on the local device in the same area as it stores other user-specific information. Each storage object provides access to a list of key-value pairs, also called items. Keys are strings. Any string (including the empty string) is a valid key. Values are similarly strings. Beyond saving data, APIs let us retrieve data, perform searches, and batch manipulations. The storage ability relies on a simple structure of key-value pairs like any JS object, and it has become one of the most popular HTML5-related additions to the web developer toolkit (Hales, 2013; Mahemoff, 2010). A few popular methods related with client-side operations are listed below (Mahemoff, 2010; W3C, 2013b).

- getItem(key) method returns the current value associated with the given key;
- setItem(key, value) method checks if a key-value pair with the given key already exists associated with the object; if it does not, a new key-value pair is added; if the given key exists, its value is updated;
- removeItem(key) method causes the key-value pair with the given key to be removed from associated with the object; if no item with that key exists, the method does nothing.

From the JS code we can reach HTML5 Storage through the localStorage object on the global window object (Pilgrim, 2010). Google and Bing store JS and CSS in persistent storage (localStorage) to improve their mobile site performance (Zakas, 2011). "I found that the mobile version of Bing used local storage to reduce the size of their HTML document from ~200 kB to ~30 kB. This is a good best practice in general and makes even more sense on mobile devices where latencies are higher, caches are smaller, and local storage is widely supported" (Souders, 2011). Implementing the local storage key-value storage enables developers to store much more data; nevertheless, browsers impose limits on storage capacity (Quota API). Offline limitation storage of most popular client-side platforms can be observed in Tabela A.1 (in Appendix A). New apps depend upon the data storage in some sort of persistent action in order to be useful, handling it with either a server-side database or HTML5 storage capacity: Web Storage, Web SQL Database, Indexed Database, and File Access (Stark, 2012).

Security

The `localStorage` object security relies on the same-origin principle — the store is tied to the origin of the site that creates it (typically this amounts to the site domain/-subdomain), so it cannot be accessed by any other origin. Origins are the fundamental currency of the Web’s security model. Stores and origins are therefore associated 1:N, where N is the number of distinct storage mechanisms represented by the current browser/origin (*e.g.*, “file:///index.html”) (Mahemoff, 2010).

3.1.4 Mobile Web-based Standards

3.1.4.1 HTML5

The HTML5 is the latest iteration of the HTML standard (markup language), and includes a collection of new features, improvements to existing features, and scripting-based APIs. Its specification¹⁶ is managed by the Web Hypertext Application Technology Working Group (WHATWG)¹⁷ and the World Wide Web Consortium (W3C)¹⁸, and was designed with the core principle of simplification of the whole HTML specification, make web content easier to code, use, and access, and ensuring that it works on just about every platform (Goldstein, A. and Lazaris, L. and Weyl, 2011). At its core are CSS and JS, which provide limitless capabilities for providing media rich content on multiple web and mobile platforms (AppFoundation, 2013). HTML5 allows developers to produce more meaningful and machine-readable content. Creating offline and location-based apps bring us the great importance of HTML5 built-in API’s such as Web Storage (*e.g.*, local storage) and Geolocation¹⁹ (Hales, 2013). Mobile web apps providing location services over a map interface — most popular map services are offered by Google Maps API²⁰ and OpenLayers API²¹ — handle the location dimension using scripts to determinate its data associated with the hosting device. This data is acquired by applying a client specific algorithm, creating a position object, and populating it with appropriate data (W3C, 2013a). HTML5 has never been so important to the mobile industry, particularly to produce cross-platform mobile apps.

3.1.4.2 CSS3

Designing mobile web apps predicates the control over appearance features such as animations and transitions. CSS3 introduces the appearance for DOM elements. While HTML5 is concerned with the structure of a web app, CSS3 is concerned with its presentation. CSS is a simple mechanism for adding styles to HTML markup. Previous to the

¹⁶whatwg.org/html

¹⁷whatwg.org

¹⁸w3.org

¹⁹dev.w3.org/geo/api/spec-source.html

²⁰developers.google.com/maps/documentation/javascript/tutorial

²¹docs.openlayers.org

current CSS version — CSS3 — developers relied upon using JS to provide presentation features applying styles to DOM elements. CSS should be used for styling, and JS for interaction. By adopting CSS3 for stylishness, it offloads a lot of the heavy lifting, often passed onto the device’s CPU using JS, to the device’s GPU (Williams, 2012), leveraging UI to deal with more robust and complex features (*e.g.*, border-radius, box-shadow). UI styling can be empowered through the use of CSS pre-compilers/pre-processors in the form of Syntactically Awesome StyleSheets (Sass)²² or dynamic stylesheet language (LESS)²³ (Goldstein, A. and Lazaris, L. and Weyl, 2011). Sass syntax is more logical and robust. Sass works as a meta-language on top of CSS, and as an extension of CSS3, providing a syntax — Sassy CSS (SCSS) — to write super functional (*i.e.* using variables, mixins nested rules, *etc.*) CSS, with the help of a command-line program or web-framework plugin (as compiler), such as Compass style library²⁴ (Keith, Cederholm, Kissane, Design, & Marcotte, 2013).

3.1.4.3 JavaScript

JS is a very flexible and powerful development language providing to developers the capacity to build full-blown solutions. It is a scripting object oriented dynamic language including types and operators, core objects, and methods. Its syntax relies on Java and C language, however does not hold classes as them; instead, the class functionality is taken by object prototypes (JS inheritance principle). JS’ functions are objects capable of hold executable code and be passed around like any other object (Willison, 2013). JS code need to be interpreted within a virtual machine translator — compilation of JS into native machine code — that runs inside the web browser, in order to be run in high performance. The technology behind this mechanism (*e.g.*, V8²⁵ from Google) implements ECMAScript²⁶ standards/specifications incorporated in recent web and mobile browsers (*e.g.* Chrome since Android 2.2) (David, 2012). Due to these client-side platforms, deploying JS code is experiencing high expansion, and client-side apps are much more complex than before (Google Developers, 2013b).

3.1.5 JavaScript User Interface Frameworks for Mobile

JS frameworks constitute a great support to the complex boosting of any type of front-end development, supporting anything from DOM manipulation to all the way through to the kitchen sink (repository), upgrading the app UI from small screens to tablets and desktops (Williams, 2012). At this time there are more than 20 mobile frameworks, and support is being rapidly added to existing DOM libraries (Charland & Leroux, 2011). The most prominent frameworks/tools for building rich apps include ST, jQM,

²²sass-lang.com, thesassway.com

²³lesscss.org

²⁴compass-style.org

²⁵developers.google.com/v8

²⁶ecma-international.org/publications/standards/Ecma-262.htm

Lungo²⁷, Telerik's Kendo UI, Intel App Framework²⁸, Wink Toolkit²⁹, and The-M-Project³⁰. These frameworks use web standards HTML5, CSS3, and JS to create an optimized experience for mobile WebKit platform, bringing a native look (full-screen view) and feel to mobile platforms (Hales, 2013; Wayner, 2013). Selecting a mobile library or framework to pursue a mobile-based project should take in account the file size, number of files, activity, and CSS3 support. The library should have a small footprint (file size); the number of assets required should be optimized (number of files); the library support and updates release cycle should be kept up with ultimate trends (activity); and its performance enhancements should be supported in GPU (CSS3 support). Facing those frameworks, Firtman (2013) remarks that jQM, ST, and Lungo are optimal targeting multi-platform compatibility. David (2012), Williams (2012), and Tolkach (2012) highlight ST and jQM, both with free solutions, as the most serious client-side framework options, and note that it is important to point out one particular difference between them — jQM seeks to work on a broad range of mobile devices, and ST targets the most operative ones: iOS, Android 2.3+, and BlackBerry 6. ST is considered by those authors as a sensational framework widely used by mobile web developers, and it embraces a robust Model-View-Controller (MVC) system architecture, enhanced support for touch events, and great API documentation.

3.1.5.1 Model-View-Controller Architecture

The MVC is an architecture pattern and describes a way to organize an app code through a separation of concerns. Using the MVC approach an app is divided into three functional parts: Model, View, and Controller. The Model handles the business data (*e.g.* access to relational database), and data validations (*e.g.* presence and/or format-matcher of the object as depicted in Listing B.7); the View holds the UI; and the Controller manages the functionalities (business logic) and actions provided to users (Kumar, 2012; Osmani, 2013). The Controller is the app decision maker and the glue between the Model and the View. Controllers are responsible for responding to events that occur within an app. The controller updates the view when the model changes. It also adds event listeners to the view and updates the model when the user manipulates the view (Google Developers, 2013b). Views use stores directly; views are 100% aware and dependent on the state of a store — data binding (Figure 3.5).

Every object-oriented framework or toolkit (*e.g.* ST) has conveyed its own sort of MVC, showing the large flexibility of the software, and the adaptability of design patterns to different situations and environments (Kosmaczewski, 2013). In a typical web-based app, the MVC architecture is implemented on the server-side to keep the view logic away from the business logic. This makes the app more maintainable and scalable,

²⁷lungo.tapquo.com

²⁸app-framework-software.intel.com

²⁹inktoolkit.org

³⁰the-m-project.org

allowing parallel development, and pursuing progressive enhancement goals. Client-side development requirements do not differ so much from the ones of the server side. HTML and CSS shape the view, JS becomes the controller, and the objects, containing the state of the UI, make the model (Kumar, 2012). In an elementary app, where most of the main logic is done by front-end MVC architecture, all necessary code (HTML, JS, and CSS) is retrieved with a single content load, or the appropriate resources are dynamically loaded and added to the system, in response to user actions, reducing round tripping, and enhancing the UX (Takada, 2012).

A typical structure of an app relying on MVC pattern is shown in Figure 3.6. Proxies are used by stores to handle loads and saves of data in the Model, client or server type; client proxies save their data locally (LocalStorageProxy or MemoryProxy) and server proxies save their data by sending requests to a remote server (Ajax or JavaScript Object Notation with Padding_(JSON-P)) (Sencha Inc., 2013b). Readers are employed to load the data model/store. Some different classes of the ST data infrastructure can be seen in Figure 3.7.

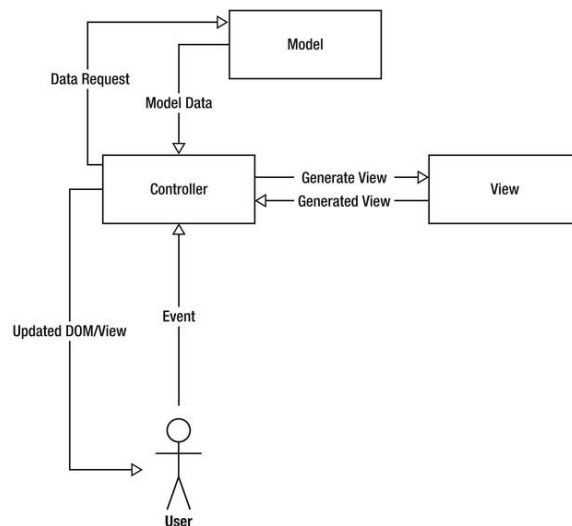


Figure 3.5: MVC Components Communication (Williams (2012)).

In these last years, different JS MVC frameworks are being developed, such as ST, backbone.js³¹, ember.js³², AngularJS³³, and Kendo UI³⁴ (Google Developers, 2013b). Among these, ST architecture constitute a very solid MVC pattern reference; it includes dozens of components and a built-in MVC written completely in JS, convenient to apps running on mobile browsers (Firtman, 2013). Below are listed ST framework classes that support programmers to build true MVC-based apps:

- Ext.app.Application acts as the entry point (throughout a `launch()` method) in

³¹backbonejs.org

³²emberjs.com

³³angularjs.org

³⁴telerik.com/kendo-ui

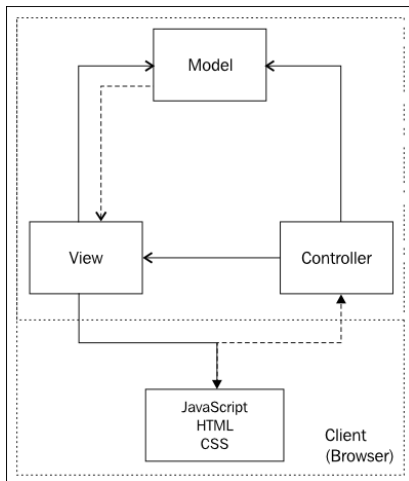


Figure 3.6: ST MVC Structure (adapted from Kosmaczewski (2012)).

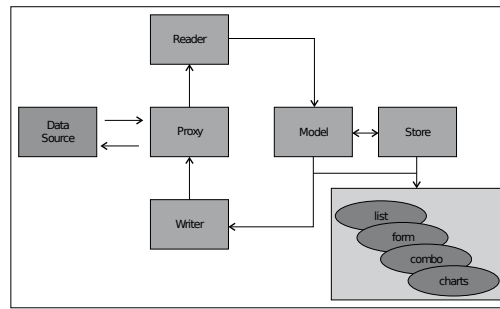


Figure 3.7: ST Data Infrastructure Classes (source: Kumar (2011)).

the MVC-based app (loading dependencies in the form of controllers);

- Ext.app.Controller lists out the views and models that it will interact with (loading the views and models); controllers hold the code that makes our app work;
- Ext.Component/Ext.container.Container are the base classes any View component extends to; Views are the framework-defined visual components (*e.g.*, form panel, buttons);
- Ext.data.Model is the base model class to define a model using the framework; model is a collection of fields and their data;
- Ext.data.Store acts as a cache containing a collection of models; Stores are abstractions over arrays of model instances;
- Ext.app.Profile gives the app profile functionality, defining different profiles for different target devices (*e.g.*, iPad, Android tablet).

An illustration of these ST components is presented in Figure 3.8.

3.1.5.2 Sencha Touch

ST is a modern JS framework specially designed to create mobile apps (*e.g.*, utility and productivity type of apps) for touchscreen devices. It is a product of Sencha³⁵ and its system derives from evolutions observed in popular JS library projects such as Ext JS³⁶, jQTouch³⁷ and Raphaël³⁸. ST offers to developers a pure JS API for building complex apps, and is designed to be extensible and modifiable out of the (*e.g.* extend

³⁵sencha.com

³⁶docs.sencha.com/extjs/4.0.7

³⁷jqtouch.com

³⁸[raphaeljs.com](http://dmitrybaranovskiy.github.io/raphael/)

new instances), simplifying arduous parts of JS by providing its own class system³⁹ (Kosmaczewski, 2013). It answers to multi-platform mobile app development questions by offering to programmers powerful tools and a fully featured development methodology, including a range of UI enhancements and widgets, app data management, and binding, to build multi-platform apps that mimic natively compiled apps, while making full use of HTML5 and CSS3 (Firtman, 2013; Garcia et al., 2013). Its JS structure is created out of JavaScript Object Notation (JSON⁴⁰), filled with pieces of HTML stored as strings, which are handled by the framework to deal with components (*e.g.* Button) in a decent manner (Wayner, 2013). ST also includes the ability to store data offline and online employing proxies (Williams, 2012).

At the time of this project development, the ST version available is 2.0.2 (ST 2). The base library of ST is known as Sencha Platform, relying on Ext JS 4.0 (version available at the time of this writing). However *ST* \neq *Ext JS*; ST is intended to the development and improvement of mobile apps only; it is not designed for desktop Rich Internet Apps (Garcia et al., 2013). ST is based on WebKit platforms (supporting only this kind of browsers); its developers can use many of the WebKit capabilities, including the debugging tools (Kosmaczewski, 2013). ST can be seen as an “all-in-one” framework. Tabela 3.3 depicts a break down of the major functional groups that make up the ST framework.

Group	Purpose
Platform	Base of ST & Ext JS 4
Layout	Set of managers for visually organizing widgets on screen
Utilities	Group of useful odds and ends for the framework
Data	Information backbone for ST; includes means for retrieving, reading, and storing data
Style	ST’s theme generated via Sass
MVC	ST’s MVC framework
UI Widgets	Visual components for user interaction

Table 3.3: ST API Groups (adapted from Garcia et al. (2013)).

ST MVC apps have a well-defined folder structure, as shown in Figure 3.9. The code of the app is contained within the app folder, as is explained in chapter 4. The app folder tree includes:

- model (holds the data model definitions; represents the object which the app uses and works with)
- store (contains the store definitions; holds a collection of models)
- view (holds the visual components)
- controller (contains classes defined in controllers)

³⁹ST App Gallery (Maps): sencha.com/apps/category/maps

⁴⁰JavaScript Object Notation: json.org

- profile (stores the different app profiles)

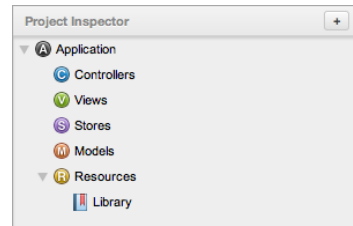
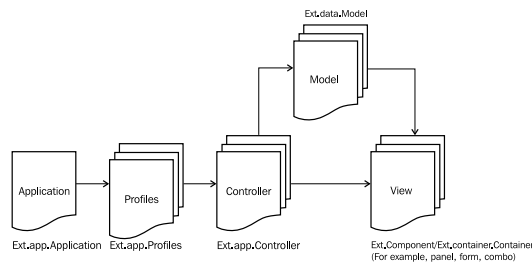


Figure 3.9: ST App Folder Structure.

Figure 3.8: ST App Architecture Components (Kumar (2012)).

Ext.app.Application defines the set of Models, Controllers, Profiles, Stores and Views that an app consists of. It automatically loads all of those dependencies and can optionally specify a launch function that will be called when everything is ready, as shown in Listing B.1.

ST class system is based upon two main functions:

- Ext.define() (*aka.* Ext.ClassManager.define()): allows developers to define a new class, including all its configuration options, static members, instance methods, event handlers, and other pieces; for instance, the presence of a config parameter provides customization options included to customize each new instance of a defined class during the lifetime of the app;
- Ext.create() (*aka.* Ext.ClassManager.instantiate()): allows developers to create new instances of a class, overriding the default configuration options and providing useful run-time information (*e.g.*, Listing 3.1).

ST includes several key features, such as a set of GUI-based components highly optimized for touch input (through a virtual keyboard) within mobile web apps, as listed below Kosmaczewski (2012).

- Buttons with device specific themes and effects;
- Form elements such as text fields (*e.g.*, email, date picker), sliders, selectors, and comboboxes;
- List component (holds momentum scrolling along with index);
- Minimalistic icon;
- Toolbars/Menus;
- Movable tabs;
- Bottom toolbars;
- Map component (supports multi-touch input: pinch and zoom);
- Components can be themed according to the target device (using Sass).

Most of the visual classes users interact within ST are components; they can be navigation components (*e.g.* [Toolbar](#), [Button](#)), store-bound components (*e.g.* [DataView](#), [List](#)), form components (*e.g.* [form](#), [Fieldset](#)), or general components (*e.g.* [Viewport](#), [Map](#)) (Sencha Inc., 2013a). One of the most employed components in the m-SportGIS development was the [form](#). A declaration of a ST form component is shown in Listing 3.1.

Listing 3.1: ST 2 Component (Fieldset) Code Fragment.

```
Ext.create('Ext.form.Panel', {
    fullscreen: true,
    items: [
        {
            xtype: 'fieldset',
            title: 'ID Tecnico',
            items: [
                {
                    xtype: 'textfield',
                    name: 'technician',
                    label: 'Tecnico'
                },
                {
                    xtype: 'textfield',
                    name: 'mobile',
                    label: 'Telemovel do tecnico'
                }
            ]
        }
    ]
});
```

ST provides several wrapped native HTML5 input fields, and custom widgets. The fields text field, check box, URL field, email field, text area, number field, password field, and radio all implement native HTML5 input elements with enhanced styling (CSS3). Widgets elements can be segmented based on its functionality as shown in Tabela 3.4. A basic building widget is a [Container](#); this component can contain other components, and handles the basic behavior of contained items. These items are rendered by the [layout](#) config property (*e.g.*, [auto](#), [fit](#), [hbox](#)). A [form](#), a [Toolbar](#), a [Fieldset](#), or a [TabPanel](#) is a container (Figure 3.10). All the container classes with certain specific behavior extend the [Ext.Container](#) class. In the cases we need a customized view design ST offers the [Template](#) and [XTemplate](#) class types to create templates using HTML fragments (Kumar, 2011).

A particular element of the ST class system is the [xtype](#) property, mapping an alias to a real name of a class in the ST class system. Through this syntax, objects can be instantiated with the [Ext.create\(\)](#) function. As an alternative, the [Ext.widget\(\)](#) function can be used to create objects just by using the [xtype](#) instead of the whole class name (Kosmaczewski, 2013).

ST is powered by a custom core which is optimized for mobile, taking the core level in ST lighter and better optimized than that in jQueryTouch (jQuery) — a jQuery plugin for

Group	Purpose
Containers	Widgets for managing child items (<i>e.g.</i> , Tab panel). Containers implement layouts (Figure 3.10)
Sheets	Pop-ups/side-anchored containers requiring users to interact with them before moving forward (<i>e.g.</i> , Date Picker)
Views	Widgets that implement data stores to display data (<i>e.g.</i> , List, Nested List)
Misc	Collection of widgets (<i>e.g.</i> , Buttons, Maps, Media)

Table 3.4: Groups of UI Widgets in ST Framework.

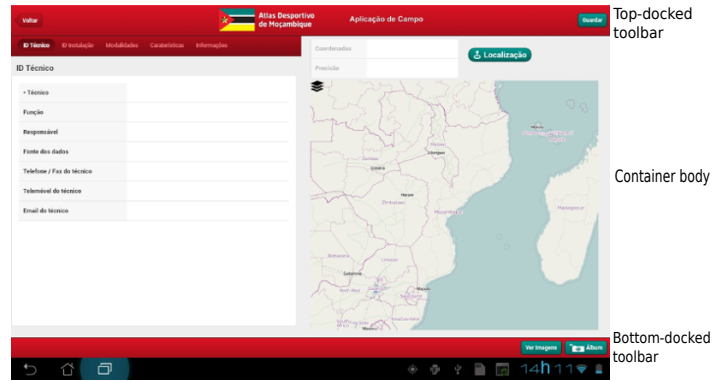


Figure 3.10: ST Components (Example of Container and Toolbar).

mobile WebKit browsers (Kosmaczewski, 2012). Its core increases scrolling frame rates, layout speed, load time, and how quickly buttons responses to touch; it gives to the native operating system — Android — its own optimized mechanism to achieve both smooth scrolling and fluid animations (Sencha Inc., 2011).

ST holds a gesture library (*e.g.*, tap/double tap, pinch, swipe, and scroll) that allows users interact with the touch-enabled device. The `scrollable` class — which uses hardware-accelerated CSS3 transitions — is a good feature example that comes close to the feel of native apps (Garcia et al., 2013).

ST is available under OS and commercial licensing. There is a ST version through the GNU GPLv3 license, which can be employed by the community, and a Free Libre and Open Source (FLOSS) license available as well.

3.1.5.3 jQuery Mobile

jQM is an OS UI framework based on semantic HTML5 markup. It is a touch-optimized web framework for smartphones and tablet PCs, however it does not hold MVC code structure; although it enjoys of a great AJAX-powered navigation system (Tolkach, 2012). jQM takes valid HTML tags as input and gives them enhanced styles according to the browser capabilities. jQM developers suggest the content should be marked up in HTML with occasional calls to JS — DIVs are the base of the app development; CSS

and JS then boost the HTML — CSS is a substantial part of the equation, rendering styles to DIVs by giving them CSS class names (Wayner, 2013). jQM works over jQuery and it manages only the UI of the app — controls, and navigation and transitions (Firtman, 2013). According to Kosmaczewski (2012), jQuery mobile can help to reach rapidly prototype projects, modeling interactions, and navigations just by using HTML standards, and offers flexibility in a very large range of mobile platforms.

3.2 Mapping Mechanisms

3.2.1 Geolocation Services

Location-based services (LBS) and mapping make up very useful features of modern mobile web apps. Today, they are very useful, then users can get web services and APIs from different providers to integrate them into mobile web development. The core of a mapping app is the map where users add and visualize data. The handset's geographical location is gathered using the W3C Geolocation API — implemented in devices such as Android — or using third-party APIs from carriers (*e.g.*, OneAPI⁴¹, BlueVia⁴²) (Firtman, 2013). The Geolocation API represents an object able to programmatically obtain position information of a device. This mechanism is supported through one or the combination of the following positioning sources: external location data (*i.e.* GPS), cell tower triangulation, and Wi-Fi data (Zandbergen, 2009). By default, device's position information is granted on the mobile devices, yet that capability is rising beyond native mobile apps, thanks to developments of the mobile web as a viable platform for mobile app development (Satrom, 2011). The Geolocation API implementation in a WebView (mobile browser) takes the navigator object in JS and passes us the geolocation property (window.navigator.geolocation), allowing us to interact with that interface (Firtman, 2013). This property provides three methods to deal with location services (Orchard, 2013):

- getCurrentPosition() for one-shot position requests;
- watchPosition() for tracking the position, repetitive function invoked whenever the device location changes;
- clearWatch() that removes the action of the watchPosition() (*e.g.*, clearWatch(watchId)).

There are different techniques to handle a geographical position request, and all of them has some accuracy error (average error distance) associated with, usually specified in a distance metric (meters or kilometers). A successful callback (navigator.geolocation.getCurrentPosition(onSuccess, onError)) receives a position (*i.e.* coordinates object) which can hold Latitude (Lat), Longitude (Lon), altitude, accuracy, altitudeAccuracy,

⁴¹oneapi.aepona.com

⁴²bluevia.com

heading (direction), and speed properties (Hales, 2013). Both `getCurrentPosition()` and `watchPosition()` (examples depict in Listing B.9) accept a success callback, an optional error callback, and an optional `PositionOptions` object. This last parameter specifies the options to fine tune the response when calling the geolocation backend. It holds three properties:

- `enableHighAccuracy` (boolean) to require the device the best possible results (supported by the device GPS chip) when is set to true (default is false);
- `timeout` is a positive long value representing the maximum length of time (in milliseconds) the device is allowed to take in order to return a position (default is Infinity);
- `maximumAge` is a positive long value indicating the maximum age (in milliseconds) of a possible cached position that is acceptable to return. If set to 0 the device cannot use a cached position and must attempt to retrieve the real current position (default is 0).

On the other hand, the `PositionError` object (code and a message/DOMString) represents the reason of an error occurring when requesting location's device. This class handles constant values: `PERMISSION_DENIED`, `POSITION_UNAVAILABLE`, and `TIMEOUT` (Firtman, 2013).

3.2.2 Web Map Client (WMC)

The native presentation of maps via a map server is an image (map component) on a web view. Imagery is one of the most important kinds of data to work within a GIS system. In order to efficiently manage map information on the client, a WMC (*e.g.*, Google Maps, OpenStreetMap (OSM), CloudMade, MapQuest, Bing Maps, ESRI ArcGIS Online) communicates with integrated WMSs. These are a standard protocol implemented by many spatial servers (*e.g.*, GeoServer, MapServer) to serve georeferenced map images to the client, through Hypertext Transfer Protocol (HTTP) requests to server spatial databases (Figure 3.11).

WMC is manipulated by spatial APIs, such as OpenLayers and Google Maps, which when included/called in an app data loading, (as can be depicted in Listing 3.2) provide special functions and methods to create and render spatial objects (Google Developers, 2013a; Perez, 2012). Listing 3.3 shows an example of OpenLayers instances and methods to create and render a map.

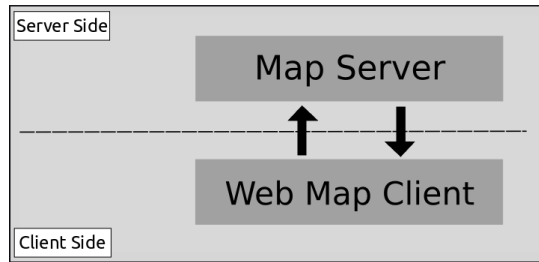


Figure 3.11: Web Map Client - Map Server Model (adapted from Hazzard (2011)).

Listing 3.2: Script Tags to Load OpenLayers API and Google Maps API (version 3.X).

```

<script type="text/javascript"
src='http://openlayers.org/api/2.11/OpenLayers.js'>
</script>

<script
src=
'http://maps.googleapis.com/maps/api/js?v=3.X&sensor=false'>
</script>

```

Listing 3.3: Instances and Methods to Create and Render a Simple OSM.

```

<div id="map" style="width: 100%; height: 95%;"></div>
<script type="text/javascript">
    var map = new OpenLayers.Map("map", options);
    var osm = new OpenLayers.Layer.OSM();
    map.addLayer(osm);
    map.zoomToMaxExtent();
</script>

```

Google Maps

Google Maps offers web services as an interface to request maps data from external services, designed to be used within a map instance, under provider service license. Unlike OS API (*i.e.* OpenLayers), users employing this API are not in control of the map client and map server back-end. Google layers (imagery) types (*e.g.*, ROADMAP, HYBRID) are defined in the API's class `google.maps.MapTypeId` as shown in Listing 3.4; when no map type is specified, Streets/ROADMAP is rendered by default.

Listing 3.4: Instantiation of the Hybrid Layer of Google Maps.

```

var hybrid = new OpenLayers.Layer.Google("Google Hybrid",
{
    type: google.maps.MapTypeId.HYBRID,
    numZoomLevels: 20
});
map.addLayers([hybrid]);

```

OpenLayers

OpenLayers is an OS JS map viewing framework, providing tools to build and customize rich web-based geographical objects. OpenLayers permits to users playing with multiple different back-end map servers, using different layer objects (*e.g.*, Google Maps) and adding them to the OpenLayers map. Some important instruments (*i.e.* classes / subclasses) that OpenLayers API provides to handle objects on the client-side include OpenLayers.Map, OpenLayers.Layer.TileCache, OpenLayers.Layer.Google, OpenLayers.Layer.Vector, Geometry (*e.g.*, OpenLayers.Geometry.Point), and Feature (*e.g.*, OpenLayers.Feature.Vector). TileCache constructor is employed to request (read) pre-populate cache tiles from the browser's Local Storage or disk-based caches (Listing 3.5). OpenLayers includes classes to integrate different imagery providers (*e.g.* Google Maps, Bing Maps, OSM, WMS service provider); Google constructor wraps code around Google Maps API, allowing users to benefit from Google Maps tiles within layer class (Listing 3.6). Vector class renders vector data from a variety of sources with a geometry and visual style associated with it. This class contains a set of features (instances of OpenLayers.Feature.Vector), where each feature has attributes and an OpenLayers.Geometry class instance associated with it, which will determine the visual representation. A feature symbolizes any real phenomenon or aspect we can visually represent with points, lines, polygons, and so on. For example, sports facilities can be represented by a point geometry, creating an OpenLayers.Geometry.Point instance with the coordinates (same projection as the one used by the map) of the point (Listing B.9) (OpenLayers, 2013; Perez, 2012).

Listing 3.5: Instantiation of Offline Layer

```
// tiles layer :
var offlineLayer = new OpenLayers.Layer.TileCache(
    "Camada base", [ "/mnt/extSdCard/tilecache" ],
    "Export_Output", {
        maxResolution : mercatorOfflineExtent.getWidth() / 256,
        maxExtent : mercatorOfflineExtent,
        numZoomLevels : 9
    });
```

Listing 3.6: Instantiations of OSM and Google Map and Layer

```
var map;
function init() {
    map = new OpenLayers.Map({
        div: "map",
        projection: new OpenLayers.Projection("EPSG:3857")
    });
    var osm = new OpenLayers.Layer.OSM();
    var gmap = new OpenLayers.Layer.Google("Google Streets");
    map.addLayers([osm, gmap]);
    map.addControl(new OpenLayers.Control.LayerSwitcher());
```

```

// set view to zoom maximum map extent:
// center(position, zoom)
map.setCenter(
    new OpenLayers.LonLat(lon, lat)
    .transform(
        new OpenLayers.Projection("EPSG:4326"),
        map.getProjectionObject()
    ),
    5);
}

```

OpenLayers can work with base and non-base layers (overlays). A base layer is a type of layer which is always visible and determines the available projection (coordinate system) and zoom levels available on the map. A map can have multiple base layers, but only one of them can be active at a time. Non-base layers can be enabled at a time, and always are rendered above base layers. OpenLayers API and Geolocation API (Cordova) methods were together included in the app Map Controller (Listing B.9) functions to handle WMS data.

3.2.2.1 Web Map Server

The most popular web servers out there are OS Apache and IIS. Map images accommodate spatial relationships, hence, to manipulate them it is convenient use specialized software — a map server — appended onto the web server. Map servers carry built-in tools to create services such as Web mapping and tile mapping (Battersby, 2012). The most popular OS map servers are listed bellow.

MapServer

MapServer is a reliable, robust OS web mapping infrastructure for publishing and analyzing any kind of dynamic GIS data (raster, vector, and database formats), and mapping apps, through a connected web server (*e.g.*, Apache). It supports scripting languages and development environments, on-the-fly projections, and is very customizable (outputs/images formats). A map file (American Standard Code for Information Interchange (ASCII) text file) addresses the MapServer configuration. It consists of a Map object including layers which in turn contains data (attributes and geometry) and styling (defined under the CLASS and STYLE directives) (University of Minnesota, 2013).

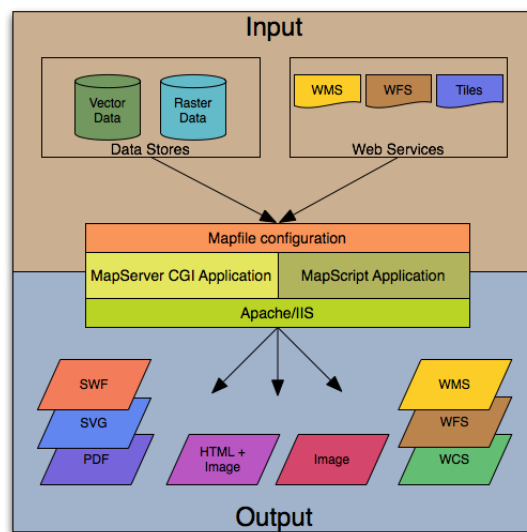


Figure 3.12: MapServer applications architecture (source: University of Minnesota (2013)).

GeoServer

GeoServer is an effective and user-friendly (graphic interface) OS map server for rendering and editing geospatial data (*e.g.*, WMS, Web Feature Service (WFS)) as an open standards web service. It is an easy method of connecting existing information to web-based maps such as OpenLayers and Google Maps, providing REST services and supporting any common servlet container (OpenGeo, 2013a).

Quantum GIS (QGIS) Server

QGIS is a popular OS GIS desktop app and server providing data viewing, editing, and analysis capabilities. QGIS Server offers WMS and WFS based on libraries from QGIS desktop app. QGIS provides extensive functionality with integration of GIS packages, such as PostGIS spatial database and MapServer. Publish data with QGIS Server is usually done running the app in Common Gateway Interface (CGI)/FastCGI module within the Apache Webserver (Quantum GIS, 2013).

3.2.2.2 Tile Map Service

Web map content that users find navigating on in-browser maps (*e.g.* OSM, Google Maps) depends on a WMS which renders it on the client-side, using HTTP requests throughout a web server (*e.g.*, Apache, Microsoft's Internet Information Server (IIS)). A tile server mechanism (or tile cache server) works as a proxy between client-side and WMS-Server, storing and managing the content of a WMS as tiles (map images) in cache (Figure 3.13). Users send map requests to the tile caching server, which manage

those calls and dynamically retrieves the content requested (tiles) to the browser/client side. Examples of this tile server are TileCache, GeoWebCache, MapProxy, MapCache, MapTiler, and Mapnik. Tiles can be pre-rendered — (seeding, *i.e.* manual requesting and caching all tiles) attending to situations where client’s requests are done offline — or generated dynamically on demand.

TMS (or Web Mapping Service-Cached (WMS-C)) provides a service to serve up cached maps of georeferenced data (*i.e.* tiles) from a tile server, established by Open Source Geospatial Foundation (OSGeo). It is employed in mapping apps and widely supported by web mapping clients and servers; OpenLayers API supports TMS natively. TileCache is one of the most accepted supporting mechanisms of TMS (Battersby, 2012). Online WMS provides us dynamic (and updated) services. In-browser maps (*e.g.*, OSM, Google Maps) take advantage of an effective tile service to generate map tiles (*e.g.*, Portable Network Graphics (PNG) format) (*aka.* slippy maps), which are stored on their servers, combining a caching system to render raster tiles to a JS-based browser display, as is being requested by user OpenLayers (2013). Map caches (2-dimentional space) represent pictures of data (files) stored in a directory exhibiting a map extent at specific scale levels (tiling the map up into a pyramid of images at multiple zoom levels (Figure 3.14)), which are accessed via a REST interface (Xu et al., 2012). Nevertheless, network communication (online services) can occasionally not be available, depending on the IT infrastructural environment: several Mozambique areas have important network limitations. In order to guarantee that users can always have access to a mapping service (locally/statically at least), creating network-communication-independent cached maps as a base map solution (offline layer) was fundamental.

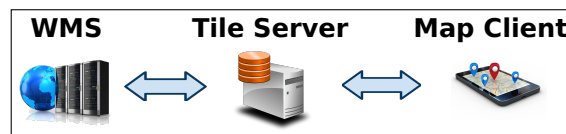


Figure 3.13: Tile Server Infrastructure.

Map Caching

Map caching is a very effective way to enhance usability through faster map data delivery and/or serve offline maps (H. Li, Fei, Wang, & Li, 2009). The mechanism is built by setting up map image tiles in a hierarchical grid arrangement according to a tile special position. Map caching is an industry standard and is becoming a popular mechanism over Internet mapping sites specially to display complex base-maps (ESRI, 2008). Caching also means quality since mapping performance is not affected by how much detail is in the map. Tile layers can be very useful to add extensive imagery, typically covering large geographical areas, ensuring scalability and improving the UX. Constructing a tiled map service presupposes planning a tiling scheme according to

available resources, together with coordinates, tile bounds and projection diligences, to design usable map content at each scale level — the largest and smallest scales at which users make effective use of the map view. OpenLayers works by requesting map tiles from a backend map server. The caching mechanism is based on file systems — each tile is saved in the cache as a separate operating system file (Hazzard, 2011). Image parameters taken into account to build the grid include the image extent, image zoom levels, scale/resolution, size of tile, tile format, and SRS. The image zoom levels are adopted looking at the original scale of images. The tile has fixed size and file format, which commonly and recommended is 256x256 pixels and PNG, respectively. The SRS usually used is the Spherical Mercator projection. Projections in GIS are generally referred to by their European Petroleum Survey Group (EPSG) codes. A current most common geographic coordinate system is the EPSG:4326 (WGS84 Datum). Most of web map representations use the official Mercator projection (Pseudo Mercator) EPSG:3857, which coordinates in projected units are meters format X/Y. Google provides services adopting the web Mercator projection, which is the most popular map projection among existing web mapping systems. This projection places the parallels and meridians in regular grids where the North is always facing up, which allows web mapping systems to divide the entire map into regular map pieces (i.e., map tiles). The mechanism of image data must be organized and indexed in order to allow effective access and map navigation. There are three main systems of tile addressing: Google XYZ, Microsoft (Bing Maps) quadkeys and TMS. TMS, specified by OSGeo, is an alternative of Google tile addressing, which is used in OS projects such as OpenLayers or TileCache (Klokan Technologies, 2008). Tiles are generated and stored/organized using a file hierarchy/pyramid (Figure 3.14) folder structure to comply an unique URL tile request format: “/tiles/{z}/{x}/{y}.png”, where {z} is the directory of the zoom level, {x} represents the folder names, and {y} takes the filenames; these denominations are determined by calculating the projected coordinates for the upper-left corner of each tile and indexing them according to the total number of map tiles (Battersby, 2012). Map caching starts from the original image data, generating a group of discrete scale maps according to a geometric rate proportional to the enhancement of detail and zoom levels. Each zoom level is a directory, each column of tiles a sub-directory, and each tile within that column a separate file. Hence, tiles are named with different rows and columns in the same zoom level according to their spatial location. The image cutting occurs inside the boundary/extent. In the case of the TMS, the cutting origin starts in the bottom-left corner of images, breaking down from the left to right, bottom to top continuously (Figure 3.15); Google’s and OSM image cutting origin is the top-left corner of images. 0 level constitutes the minimum scale in the grid where one or two 256×256 tiles with full-screen are displayed, and when the map is amplified one level the scale increases one time, and the original 256×256 image is turned into four 256×256 tiles (Figure 3.16) (Klokan Technologies, 2008; Xu et al., 2012). Depending on layers extensions and number of zooms levels produced, the information needed for

a file system to locate and open these files (and the number of correspondent tiles) can be enormous, and impossible to exist in the device main memory (Y. Zhang, Li, & Zhu, 2008).

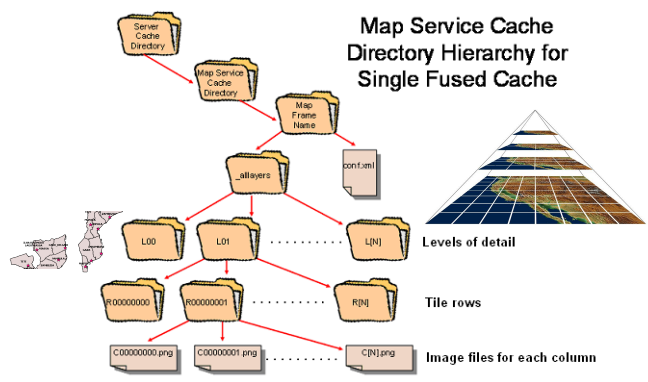


Figure 3.14: Tile cache hierarchy (adapted from (ESRI, 2006; Potmesil, 1997)).

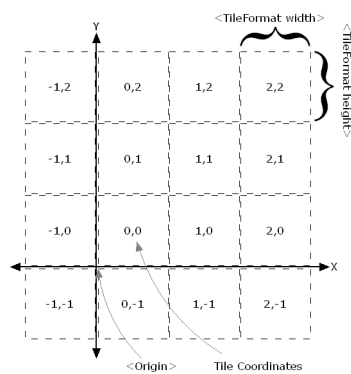


Figure 3.15: Tile map origin and coordinates scheme (source: OSGeo (2012)).

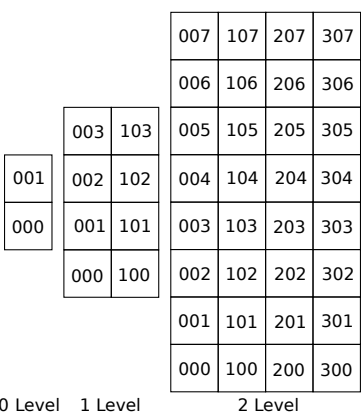


Figure 3.16: Tiles hierarchical grid arrangement (adapted from Xu et al. (2012)).

The number of scale levels and which scales to use constitute a very important decision since its number implies an exponential cache storage space requirement, and cache creation time. Tuning the map for accurate generalization, symbology, and labeling can be a challenge since the cached map viewed on a computer/tablet PC screen must be readable and useful at all cached scales. A naive way to create a tiling scheme is decide the closest and appropriate scale (maximum resolution) at which users will need to view the map (conveying information with more detail), and then raise the scale until reach a scale factor that would contain all of the area of interest in just one or two tiles (maximum extent: Mozambique). A well-design map considers scale-dependencies to labels and features — reveal elements (*e.g.*, administrative divisions, roads, city points, points of interest) at certain levels included in the tiling scheme — to help map holding information in an organized manner (ESRI, 2008).

3.2.2.3 Tile Caching Mechanisms

TileCache, GeoWebCache, and MapProxy are likely the most employed tile caching servers (University of Minnesota, 2013). These OS servers are based on upon the popular OS map server MapServer and GeoServer. Despite these solutions can run as a standalone product, usually the adoption of a tile server depends on which map server API developers are using or familiar with (Battersby, 2012).

TileCache

TileCache is a Python-based protocol/implementation of a TMS (or WMS-C) for serving maps as tiles, distributed under Berkeley Software Distribution (BSD) license. It was designed associated with OpenLayers, creating a disk cache (*e.g.* Disk, Memcached) for maps and layers. Its service can automatically determinate a set of scales according to a maximum extent, maximum resolution, and number of zoom levels given. Considering OpenLayers API, `OpenLayers.Layer.TileCache` constructor is used to request tiles cached in a web accessible location. TileCache acts as an enhanced map service, sitting between the map client tier and the map server tier, to render maps (WMS layers) efficiently and to increase usability using a cache of static images (pre-rendered tiles) (ESRI, 2006; OpenLayers, 2013). TileCache holds different implementation modes (scripts) such as CGI, Web Server Gateway Interface (WSGI), `mod_python`, and IIS (MetaCarta Labs, 2010), capable to draw “meta-tiles” where a large tile is rendered, and then broken down (tiled) into smaller tiles using the Python Imaging Library (MetaCarta Labs, 2010). Its installation include a configuration file to control the map caching process, where options and parameters are defined (Listing C.1 in Appendix C). The seeding scripts (*e.g.*, CGI) provide an interface between the web server and apps that create web content (WMS layers). To seed tiles executed running `tilecache_seed.py` tool defining options, layer, zoom start, and zoom stop parameters (<options> are extra seeding configurations, <layer> is the target layer stated in the configuration file, <zoom start> is the level to start the process, and <zoom stop> the level to end the process) (MetaCarta Labs, 2010). Most used request mechanisms used with TileCache are WMS and TMS requests (MetaCarta Labs, 2013).

GeoWebCache

GeoWebCache is an OS tiling server integrated with GeoServer (map server), implementing service interfaces (*e.g.*, TMS, Google Maps) through a web-based administration interface over a configuration file (`gwcConfiguration`). It has a philosophy similar to TileCache, and is licensed under the Lesser General Public License (LGPL) (OpenGeo, 2013b). Diagram in Figure 3.17 shows the components GeoWebCache system deals with.

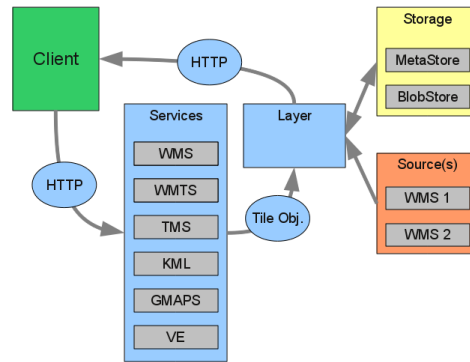


Figure 3.17: GeoWebCache Architecture (source: OpenGeo (2013b)).

MapProxy

MapProxy is a flexible and OS proxy, focused on the tile caching mechanism to speed up mapping apps by pre-rendering and integrating maps from multiple sources (*e.g.* MapServer, GeoServer) and storing locally. It also offers support to innovated features to WMS clients. MapProxy has been developed by Omniscale⁴³ under Apache Software License 2.0, and can be used for simple or complex implementations (tiles for OpenLayers or central SDI nodes/distributed WMS services).

3.3 Software Development Methodologies in Organizations

Information management is critical to innovation in organizations, and it represents the process of creating, storing, maintain, and sharing information and knowledge to improve organizational performance, expertise and competitive intelligence (Alavi & Leidner, 2001; Davenport & Prusak, 2000). That information takes relevance as it boosts transparency and accountability of the decision-making process, and it can therefore contribute to good governance. Consequently, now more than ever, the seek and demand to barely transform the way public services operate received the interest and attention of political bodies, who have started considering the formulation and implementation of strategies, that could lead to the achievement of that transformation.

ITs aid the potential to transform public administration sector operating paradigm, making it more efficient, effective, transparent, more citizen friendly, and more capable of delivering cross-agency public services, like storage system and knowledge exchange, in an integrated way (Soares & Amaral, 2011).

Methodologies are step-by-step procedures to carry out the development activities in different phases of a system development life-cycle (SDLC). They organize sets of behavioral and technical rules into a coherent approach, which prescribes how to address major development problems with the required functionality and quality. Those proce-

⁴³ omniscale.com

dures incorporate assumptions, and working principles and techniques about the reality to enforce discipline, tools to generate the deliverables for activities, and define roles (who) for the assignments in the development (solution architects, business consultants, and developers) (Dahiya & Dahiya, 2010). These personnel are central players in the way that in-house development of IS methodologies and apps are conceived, analyzed, designed and implemented.

An information system development (ISD) project should involve a solution for determining which system development approaches best satisfy in efficiency and effectiveness the project assignment requirements, regarding its successful implementation. Enterprise software development and its quality control assessment are potential expensive, depending on the scope of project's tasks; therefore, setting up a proper software development, and testing and maintenance environment is very important to success. Software development can be very arduous and expensive without a proper framework, architecture and model. Adopting an appropriate methodology profits software developers to communicate with users/stakeholders, enabling a better product requirements analysis, improving the product design before its conception, speeding up its development and enhancing its functionalities along the that phase, optimizing development and production costs, and making the product scalable (flexible/reusable code) (J. Zhang & Chung, 2003).

ISDs and implementations has been a part of the IT community since the inception of the modern digital computer (Avison & Fitzgerald, 2006b). The current evolution of ICTs takes effect on software development and usage, in a human and organizational dimension. In this "digital economy era", outcome from the convergence of computer and telecommunications technologies, organizations pursue to be responsive, adaptive, and flexible in their operations and strategy. Organizations are understanding that their operations, products, services, information, markets, competition, and economic environment are all potentially being affected by this ongoing dynamic business environment. Those agents are seeking opportunities to create more digital content in their products and services. In this sense, Information Systems Development Methodologies (ISDMs) play an important role to achieve successful software project implementations (Papatsoutsos, 2001). According to Avison and Fitzgerald (2006a) an ISDM is a collection of philosophical views, phases, procedures, rules, techniques, tools, documentation aids, management and training which help systems developers implement a new IS. Over the recent decades, ISDMs are being adopted by organizations (*e.g.* governments) to structure, plan, and control the process of developing new ISs, taking specific methodologies as prerequisites in order to embrace important software projects. There is a variety of ISDMs variants taken up to develop and improve ISs, and software engineers struggle to select a right methodology that works for all development situations. In this line, no methodology can claim that can be applied to any organization, despite those models keep a basic structure relying on a feasibility study, system inves-

tigation, system analysis, design, implementation, testing, and maintenance. Tabela 3.5 shows typical IS problem situations and corresponding methodologies based on IS development practices (Zaied et al., 2013). The same authors, based on business survey, identify that most of companies implement hybrid or customized approaches as the most adopted methodology in web system design, followed by traditional SDLC, and Rapid App Development (RAD) or agile development methods.

Problem Situation	Methodologies
Technical problem situation	Traditional methodologies, Agile methodologies, Rapid Application Development methodology, and Prototyping
Social problem situation	Holistic methodologies
Socio-technical problem situation	Socio-technical methodologies
Complex problem situations	Frameworks (techniques from multiple methodologies)

Table 3.5: Problem Situations and Methodologies (adapted from Zaied et al. (2013)).

In terms of techniques, prototyping and entity-relationship diagrams are the most used by web developers. RAD methodology is based on designing product functionalities prototypes in parallel processes ,and then re-engineering them into production quality code; this approach should be taken by expertise developers. Agile modeling is a practice-based methodology grasping a collection of values, principles, and practices for modeling software, employed on a software development project in a more flexible way than traditional modeling methods. The ability to respond to the changing requirements of the project result of a continuous communication (inputs) between the development team and the costumer represents the base principle of the agile methodology; the costumer satisfaction is its main aim (Balaji & Murugaiyan, 2012; Prakash et al., 2013). A comparison between typical conventional (Waterfall) and incremental (agile) development approaches is shown in Figure 3.18. The decision of a suitable approach should be a reflection of requirements stability, nature of end-users, size of the project, system developers experience, and the involvement scope (enthusiasm) of users/stakeholders (Avison & Fitzgerald, 2006b).

SDLC has had important influence as a general approach to develop IS, and has been improved and modified over the years, in response to ever-changing scenarios and paradigm shifts related to the building or acquiring of software. The most obvious is the Structured System Analysis and Design Method (SSADM) which attempts to improve the way that processes in businesses are carried out. For the development of large software and systems in business, industry, and government, the SDLC has been treated as one of the two leading information systems development methodologies along with prototyping (Piccoli, 2012). SSADM is a structured methodology, created to develop data systems by the UK government in the 1980's. It takes a rational and engi-

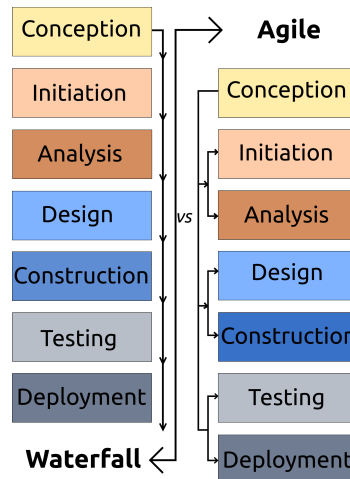


Figure 3.18: Waterfall *vs* Agile Methodologies (adapted from Prakash et al. (2013)).

neering view of the world, data structure orientation, adopting the “Waterfall” model. It is based on the clear requirement of orders and some detailed rules, assuming that all requirements are identified at the start, all the system is analyzed, then designed, hence written, and finally tested. Therefore, all the system is handed over to the client, so there is only one run through the life-cycle (Middleton, 1999; Munassar & Govardhan, 2010). The main advantages of this methodology rely on its simple approach over the issues through the user’s vision, large flexibility in analysis and projection, simple implementation of the system, and the documentation is very suggestive and complete (Pascu, 2010). However, this top-down process is somewhat naive, and it holds a lack of attention given to the human context of ISD (Hirschheim, Klein, & Lyytinen, 1996). Prototyping and agile methodologies are based on the idea that planning all requirements and contingencies in all but trivial system developments is impossible; software requirements analysis in a pre-implementation system phase may not capture all relevant requirements, which eventually will emerge when the system is being designed in detail and well implemented, involving small or large project changes, which imply different project costs (Vidgen, 2002). Based on current real world problems inside organizations, ISDs are iterative processes, carrying out a series of iterations when new requirements are identified, modified, or sub-phases are taken as unnecessary or inappropriate. Regarding this changing and evolving reality, and focusing on a costumer value by encouraging a productive climate of participation and collaboration — agile principle — evolutionary and incremental development models such as spiral model, prototyping, or RAD, constitute a more logical solution to mobile and web-based systems (Dahiya & Dahiya, 2010).

Prototyping involves constructing a small-scale version of a complicated system in order to acquire the critical knowledge required to build a full system, in a fast, cheap, and reliable way; this facilitates design checking, testing, discussion, and revision by developers and stakeholders (Bochicchio & Fiore, 2004). Nevertheless, a good emphasis

on the analysis and design stages — concepts of SSADM — over the mobile web development can reduce the coding and production costs.

Software apps often evolve, both in size and functionality. A methodology that considers this evolving approach can speed up and cost-optimize the development process, focus on quality, accordingly to user concerns, and improve features and solve issues as technologies are being reviewed and updated. Inside organizations, it is assumed that a certain amount of method tailoring is required in order to make use of any methodology, or indeed several methodologies may need to be combined (Howcroft & Carroll, 2000). For this reason and based on prototype methodologies, the framework selected to implement the app integrates a combination of dynamic systems development methods, being flexible enough to take into account the mobile web technology singularities.

3.3.1 Methodology for m-SportGIS Development

Regarding the governmental organization structure and its players, and the IT project characteristics — evolving both in size and functionality — and considering Howcroft and Carroll (2000) advices, an hybrid approach was taken as the best solution to the m-SportGIS development.

The following section presents the considered development procedures. The first large project phase considered principles of Structured System Analysis and Design Method (SSADM) and RAD philosophies. This contemplated a software analysis to set up the requirements and the conception environment. This analysis includes the hard and soft skills required to implement a software solution, the type of end-users, costs estimation, and the risk involved within the app development life-cycle. This phase is probably the most crucial phase of the methodology. To discuss and plan main project guidelines, meetings between Mozambican Government staff, and software developers and managers of NT Lab of ISEGI-NOVA took place. Developers must be aware of the current ICT and low-cost solutions in order to fulfill project objectives effectively, particularly, mobile and web development under OS software. A technological analysis on low-cost mobile and web technologies was conducted based on outcomes from stakeholders in order to fulfill the objectives effectively.

A second important phase involved the design of an app model attending to the specifications defined. Build m-SportGIS in a controlled way needs a good design architecture, iteratively built, adding sections and processes during its lifetime due to new functionalities. The response of IS developers to this scalable strategy is encourage users feedback based. It was launched an iterative operation consisting in the conception of a mock-up (model-driven) of the full system (using OS Pencil Project), delivering a suitable outlook of the final product with its general functionalities, minimizing risks of incomplete or inadequate issues.

The next significant phase was focused on the development and production of the full app, driven by the accepted model-driven product. Using OS resources such as an IDE,

and web and mobile APIs, a beta version was produced and tested at the laboratory using mobile devices Samsung and Asus (Android system v 4.0.3 and 4.1). After, the app was tested in-situ taking into account the technological environment that users can deal with during their field works.

Later, technical documentation (manual) was written, and users in Mozambique were trained. Next, a refined official app version and its documentation were made available in the Mozambican governmental platform (“Atlas Desportivo de Moçambique”).

Currently, a maintenance period is being taken, with users support, technical corrections, updates (*e.g.* new functionalities), and improvements, to embed incremental value to the product.

4 Development and Implementation of m-SportGIS

4.1 Introduction

m-SportGIS is a field mapping app, developed under OS technologies, that has been created to be launched on Android devices held by technical staff of Mozambican Government. This mobile component was conducted as a NT Lab adjudication project to implement the National Atlas of Sports of Mozambique. m-SportGIS constitutes a low-cost governmental solution to collect sports facilities in that country. The full system has been designed with two main components: a desktop SDI (WebGIS platform) and a mobile mapping platform. The SDI allows to visualize information and process spatial data, namely, search data, make spatial and statistical analysis, create new data and edit existing one, and produce different type of customized reports, and thematic maps. Data is permanently stored and managed in a spatial server database (PostgreSQL - PostGIS). Full system architecture is presented in the diagram of Figure 4.1.

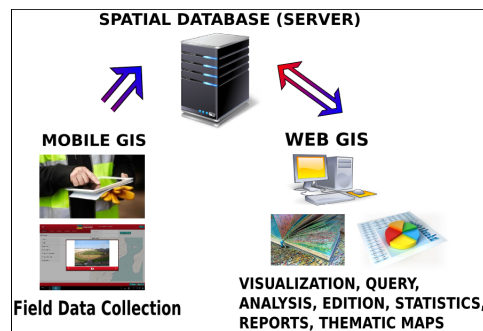


Figure 4.1: National Atlas of Sports System Architecture (source: de Abreu Freire and Painho (2014)).

4.2 System Requirements Overview

Attending to the mobile system specifications, and taking into account the project assumptions (outlined in Section 1.4), the outset point of activities comprised the definition of MJD requirements for the solution that would be implemented; thereunto, meetings with governmental stakeholders took place, including a demonstration of a

possible platform environmental structure (prototype). The collected requests conducted a research on app development methodologies and procedures (considerations detailed in Section 3.3). A special attention in the app design was given to usability issues. The platform is constituted by alphanumeric and spatial data inputs panels, and data is validated before being stored. Alphanumeric saved data is able to be edited. The app needs to be communication-independent (offline capabilities), but ensuring that data acquisition is conducted *in-situ*. Data stored locally is able to be sent to a server when network communication is available.

4.3 System Architecture

The establishment of the mobile system architecture contemplated different low-cost software tools and environments to accomplish the required purposes. Beyond their functionality, the selection also took into account the availability of resources (*e.g.* documentation, forums, new trends), licenses, standards, specifications, operation system, and programming language. Client-side software apps conceived with a multi-tier methodology offers an optimal amount of scalability to the system, allowing any tier to be upgraded, replaced, or interchanged independently (Ravula, 2009). A typical multi-tiered app consists of a presentation tier, a business logic tier, and a infrastructure (data access) tier. m-SportGIS was constructed based on this methodology (Figure 4.2).

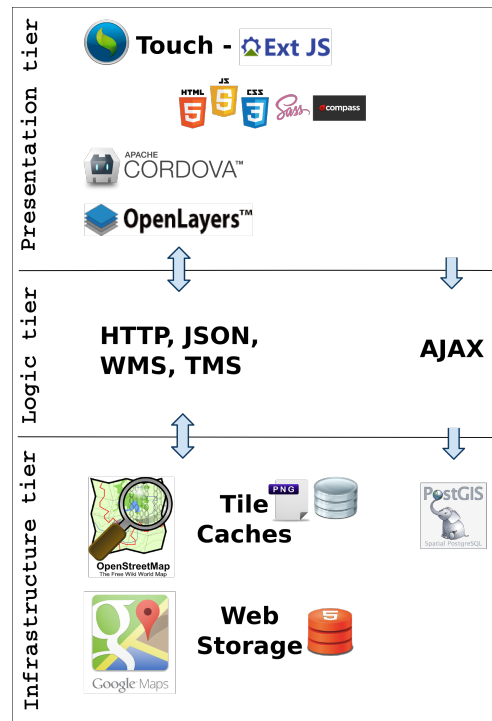


Figure 4.2: m-SportGIS Architecture.

The presentation tier represents what the users view (thin client). In the app, this is

represented by a Viewport with form panels, buttons, and a map panel. OpenLayers API, a popular and growing OS spatial library, was integrated to add spatial UI and functionalities to this tier. All non-spatial components were built employing the ST framework (mobile-oriented library based on Ext JS, which in turn relies on JS, HTML5, and CSS). Compass was used to compile Sass code to CSS. Cordova API (PhoneGap) was integrated in the app resources to get its functions and objects and reach the device capabilities (*e.g.* camera, File system).

The logic tier controls the routes and the methods by which system objects are accessed and updated. This tier works as a bridge from the presentation tier to the infrastructure tier, defining how users storage new data or send it to a database server (*e.g.* POST request to PNG file using HTTP), and how they request resources and get back their information (through a HTTP Uniform Resource Locator (URL)).

The infrastructure tier consists of data persistence mechanisms (*e.g.* file shares, database servers) and the data access layer which deals with data storage interactions. OpenStreetMap and Google Maps were integrated in the data app resources to deliver online services (WMS). A tile cached service (tiles stored on disk) was established running MetaCarta's TileCache script (v 2.11) over customized map files (.map) exported from a QGIS project (Figure 4.3). While users are working on field (usually offline), data inputs were stored and edited under a file-based mechanism; alphanumeric and spatial (coordinates) values were stored as items (key : value pairs) using HTML5 local storage; media data (mime type = image/jpeg) was stored on disk (memory card). Text data sent to the remote server was stored in a spatial database (PostGIS), including a referenced path that indexes a record to the correspondent pictures, which are sored in separately file-based structure.

4.4 Development and Implementation

4.4.1 Data Definition and Preparation

The heuristic development of the project started out searching and collecting information about sport data surveys and installations features to build a “navigation map” (hierarchical list) of the principal characteristics associated with sport facilities (Tabela A.4 in Appendix A). This list got several updates during ongoing meeting with stakeholders and future m-sportGIS releases. The last version has 150 fields.

Other necessary exploration was look at existing spatial data from Mozambique. Ensuing an analysis to different sources, such as National Cartography and Remote Sensing Center (CENACARTA)¹, National Disasters Management Institute (INGC)², and DIVA-GIS, data (shapefiles) designated to carry out a WMS-cached (offline service)

¹Centro Nacional de Cartografia e Teledeteção: cenacarta.com (accessed on 14 January 2013)

²Instituto Nacional de Gestão de Calamidades. Data source: moz-adapt.org (accessed on 18 January 2013)

was processed from CENACARTA and INGC. Selected data is listed in Tabela A.2 (in Appendix A). Original shapefiles projection/coordinate system was EPSG 4127 (Tete); using QGIS Desktop (v 1.8 Lisboa) (GNU GPL v3 license) they were reprojected to EPSG 3857 (Mercator projection) to be consistent with online map layers (OSM and Google Maps) being integrated. A project in QGIS Desktop was created to undertake analysis and edition (geometry corrections), customization, and cleanup (attributes inconsistencies) of selected shapefiles, and to study scales and zooms being implemented. The information presented on map (points, lines, polygons) was designed by composing the layers with proper symbology and style. The final result of the QGIS project (moz_3857.qgs) including all the layers is displayed in Figure 4.3.

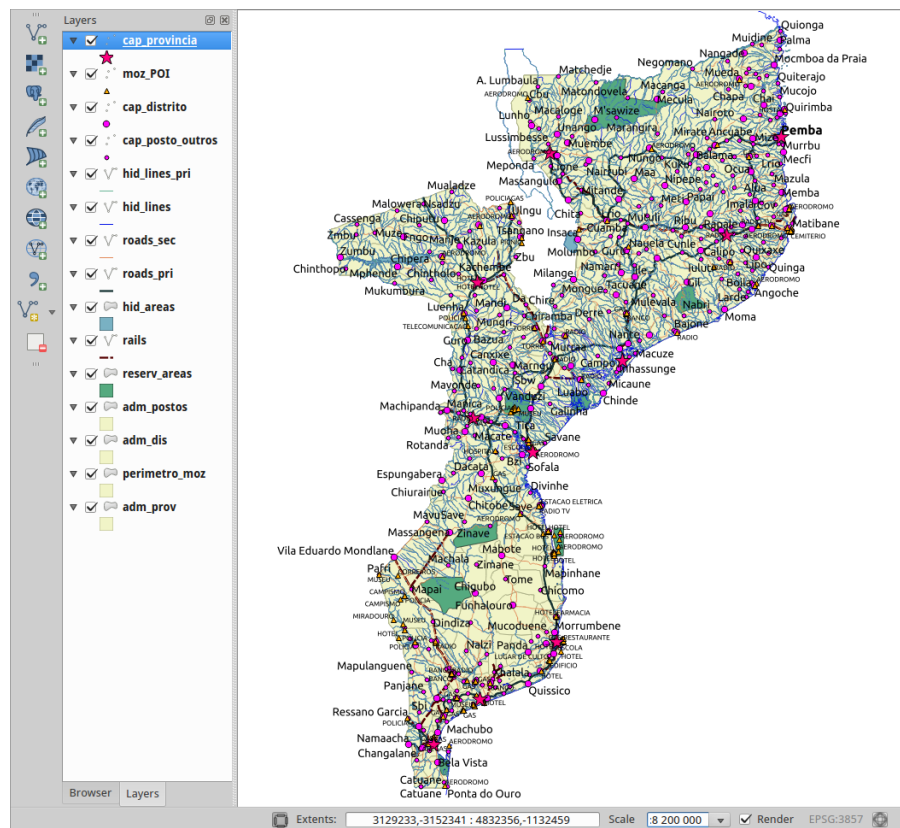


Figure 4.3: QGIS Project.

4.4.2 Creation of a Tiled Map Service

Map tiles of Mozambique were produced from the QGIS vectorial project (moz_3857.qgs). In order to handle the exportation of the moz_3857.qgs to moz_3857.map (map file), a QGIS plugin — PyQgis MapServer Export — was installed and enabled using the Plugin Manager. Required inputs to the export plugin were the path and name to the output map file, and the QGIS project file (data source) saved previously. Since that MapServer Export plugin only supports basic/old symbology style, the output file symbol.map needed some tweak to get the intended results. The main map file

configuration (common to all maps generated) is presented in Listing C.2 (in Appendix C). An example of one of layers (primary roads) configuration included inside the map object is shown in Listing C.3 (in Appendix C). MapServer app (CGI) (v 6.2.0), Apache HTTP Server (v 2.0), TileCache (v 2.11 under a CGI-script including the script `tilecache_seed.py`), and Python_installation were also set up to run on the operating system (Linux - Xubuntu 12.10). Figure 4.4 depicts the logic software schema of tiles production. The exported `moz_3857.map` containing all the vector layers (accordingly to data of Tabela A.2) as an object with layers and a variety of parameters, was broken-down into others map files, representing content accordingly to a sequence of zoom levels and scales that should act as the scale range principle (displaying certain map content at different zoom levels/scales).

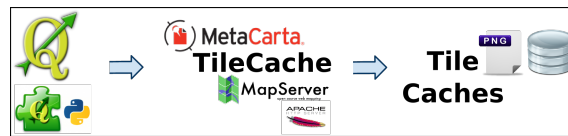


Figure 4.4: Tiles Production Schema.

4.4.2.1 Analysis of Zoom Levels, Scales, and Resolutions

Mozambique has an extension of 790 367 km² (INE, 2007). Its coastline extends from 26° 52' S to 10° 40' S, and main administrative divisions comprehend provinces (13), districts (146), and towns (411). After composing the spatial data in the QGIS project (`moz_3857.qgs`) (EPSG: 3857) was possible to extract directly the Mozambique's Bounding Box (BBox) (axis-aligned geometry) in [X,Y : X,Y] format — 3339584, -3123471 : 4564100, -1118890. Tiles of the basemap were defined to be PNG files of 256x256 pixels.

To create customized tile content being useful for users working in the field, TileCache script was executed and multiple outputs tests were performed, generating different zoom levels. The script settings were done in its configuration file — `tilecache.cfg`. The most important parameters to define in the file are BBOX (box of the layer), file EXTENSION (PNG), name of LAYERS to be rendered by MapServerLayer, number of zoom LEVELS, maximum resolution (MAXRESOLUTION) (how many units does a pixel on the screen represent), tile SIZE (256, 256 by default), SRS (EPSG: 3857), TYPE of layer (WMSLayer), and URL of layer (.map). Defining the type of cache (Disk) and base (path to tiles export folder) are also necessary. The relation used between number of zoom levels and layers (content) displayed is presented in Tabela A.3. Tile's MAXRESOLUTION (4783.265625 m) was calculated using the formula $MAXRESOLUTION = \frac{X_{MAX} - X_{MIN}}{WIDTH}$ where $X_{MAX} = 4564100$, $X_{MIN} = 3339584$, $WIDTH = 256$ (deduced from (Cothran, 2010) calculations). Giving the MAXRESOLUTION and the LEVEL number (integer), TileCache can calculate an

array of resolutions/scales for the outputs (tiles). Based on the zoom levels tests to get a more and less suitable detail in towns at the maximum scale level, and physical space required to build the cache, nine zoom levels (0 until 8) were defined for the cached tiles.

To generate tiles, `tilecache_seed.py` was executed in the console/terminal inside the TileCache directory — `tilecacheseed.py map0 0 1` where `map0` one of the map files (contains `adm_prov`, and `cap_provincia` layers) derived from `moz_3857.map`, where 0 is the start zoom level, and 1 is the stop zoom level. This operation was repeated until the ninth zoom level (`map0`, `map1`, `map2`, `map3`, `map4`, `map5`, `map6`, `map7`, `map8`). Tabela A.3 (in Appendix A) presents map content defined for each zoom level. Output files structure had the format `.../tilecache/ExportOutput/00/000/000/000/000/000/000.png`. Tiles directory was moved to an external physical card (32 GB capacity) being the data source of the offline map service.

4.4.3 System Features and User Interface

4.4.3.1 Prototype Conception

In attendance to an evolutionary and incremental software development philosophy, an app mock-up was designed using the OS Pencil Project (v 2.0.2) (GNU GPL v2 licence) to deliver a preliminary app prototype. Pencil project includes a collection of tools, elements, native UI widgets, and mobile Android ICS UI elements able to produce an outlook of a functional app concluded product with the general functionalities. The app was designed to include three windows (containers). A presentation window (or main view) with a Mozambique logotype and two buttons (Figure 4.5) — Create-New-Record, and View-Records. The Create-New-Record view was designed to be the data acquisition environment, including a sport-related-attributes form with different type of inputs (drop-down list, check box, alphanumeric input), and a map panel displaying a map object with WMS (OSM) and WMS-C (offline) layers, and a built-in-GPS-engaged functionality panel to get and show the current spatial location (Lon and Lat in degrees) and its accuracy (in meters). The container also contemplated in the header and bottom panels five action buttons — access to built-in camera, access to photo library, show a panel with current record taken photos, save current record, and back to main view (Figure 4.6, Figure 4.7). The View-Records container presented a list panel with records that have been saved, a map panel showing their spatial location, and a header panel with two buttons — submission all records to a remote database server (PostGIS), and back to main view. Associated with each record in the list there were a button to submit individually its data to a remote database server, and a edition mode button to modify its data (Figure 4.8). m-SportGIS mock-up was presented to the client and after acceptance of the outlook of functionalities, the development of a prototype version (v 1.0) took place.



Figure 4.5: Mock-up: Initial View.

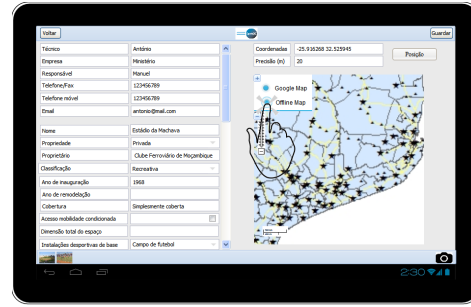


Figure 4.6: Mock-up: Collecting View with Map Panel (BaseLayers).

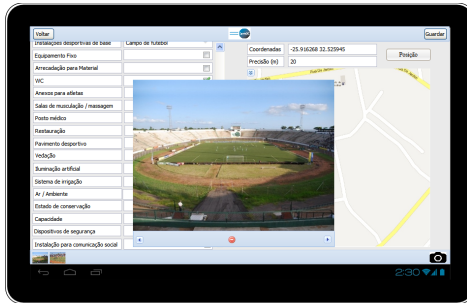


Figure 4.7: Mock-up: Displaying Picture Panel over Form View.

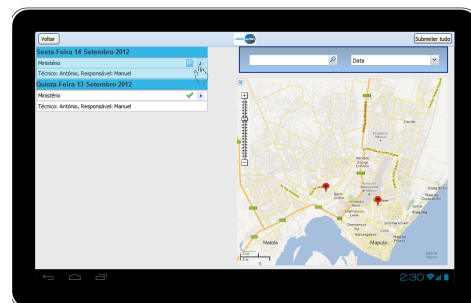


Figure 4.8: Mock-up: Displaying Saved Records List and Map Panel.

4.4.3.2 m-SportGIS Features and User Interface Implementation

m-SportGIS is a mobile web-based platform, targeting Android devices, developed under a hybrid code approach and MVC system architecture.

Different frameworks were employed to work with different components. UI components were built employing ST API (v 2.0.2) and customizing some classes. Sass (v 3.2.6) and Compass (v 0.13.alpha) libraries, together with RubyGems (v 2.0.0) were used to edit the style of UI components; ST resources include styling files — SCSS, CSS, themes — with a config.rb file, where Compass target action is defined. Spatial data interactions (*i.e.* WMS, WMS-C requests) were made out using OpenLayers Map Viewer Library (v 2.12). Apache Cordova/PhoneGap (v 2.5.0) was integrated to deal with native functionalities of the Tablet PC (*e.g.* camera, photo library, GPS sensor, and file system access).

Along m-SportGIS development, app UI and functionalities were tested on Chrome (WebKit-based browser), and in physical devices — Asus Transformer TF101G 10" (running Android v 4.0.3) and Samsung Galaxy Tab 2 7" (running Android v 4.1).

The heart of m-SportGIS was a function (Listing B.1 in Appendix B) that defined

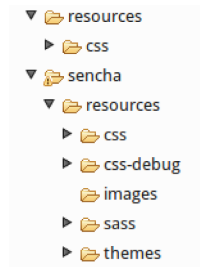


Figure 4.9: ST Resources Content.

the app MVC components and created an app instance — MainView — which loaded View panels (“startview”, “datacollectingview”, “facilitylistview”), Store, and Model components. Listing 4.1 shows portions of code that creates the MainView.

Listing 4.1: m-SportGIS MainView.

```

// SportApp.view.MainView :
Ext.define('SportApp.view.MainView', {
    extend : 'Ext.Panel',
    alias : "widget.mainview",
    config : {
        fullscreen : true,
        layout : {
            type : 'card',
            animation : {
                type : 'slide'
            }
        },
        items : [ {
            xtype : "startview"
        }, {
            xtype : "datacollectingview"
        }, {
            xtype : "facilitylistview"
        } ]
    },
    initialize : function() {
        this.callParent(arguments);
        /* painted: function renders map component with
        saved facility points
        */
        this.on('painted', function() {
            Ext.getStore("facilityStore").load();
            this.fireEvent("loadimages", this);
        });
    }
});

// Facility Store :
Ext.define('SportApp.store.FacilityStore', {

```

```

    extend: 'Ext.data.Store',
    config: {
        model: 'SportApp.model.FacilityModel',
        sorters: [{
            property: 'date',
            direction: 'DESC'
        }],
        proxy: {
            type: 'localstorage',
            id: 'facilities_app_store'
        },
        grouper: {
            // structured content (fields)
        }
    }
});

// Facility Model :
Ext.define('SportApp.model.FacilityModel', {
    extend: 'Ext.data.Model',
    config: {
        idProperty: 'id',
        fields: [
            // fields
        ]
    }
    // validations
});

```

Figure 4.10 presents the app StartView with the its logotype and name (“Atlas Desportivo de Moçambique” - Aplicação de Campo), and two buttons, to create a new record (“Criar Novo Registo”), and to view saved records (“Ver Registos”). Listing B.2 (in Appendix B) shows written code to build the StartView.

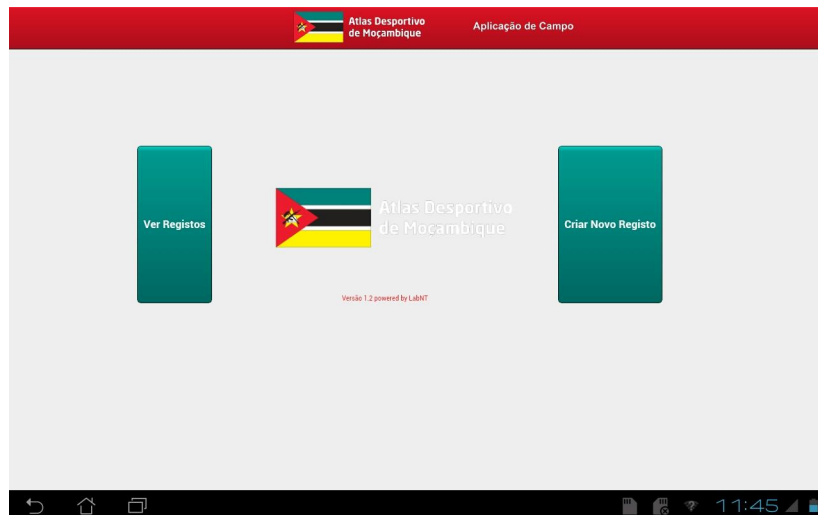


Figure 4.10: m-SportGIS: Start View.

Figure 4.11 displays the DataCollectionView, consisting of five different areas: two toolbars, a field data form, a coordinates data form, and a map panel. Toolbars (header and bottom) integrate a back to StartView button (“Voltar”), a save record button (“Guardar”), the app’s logotype and name, a view pictures button (“Ver Imagens”), and a button to access the device photo library (“Álbum”). Listing 4.2 shows written code to create a Toolbar and the Button (“Guardar”), including its functions definition.

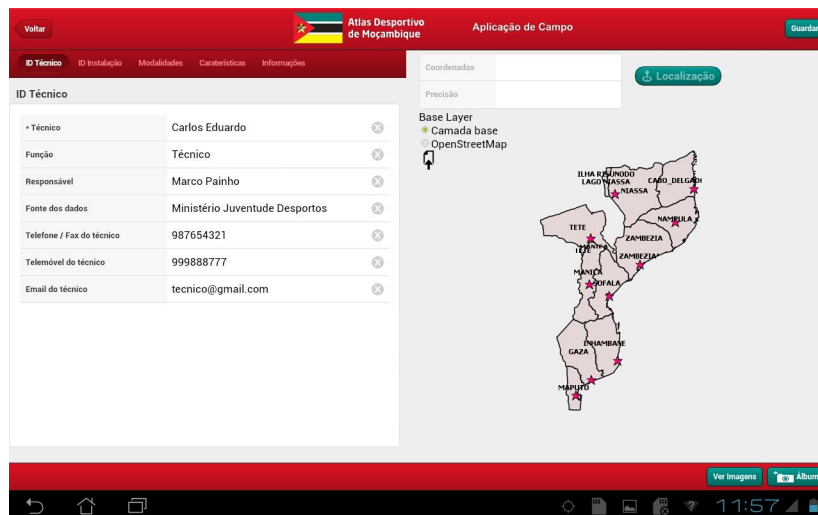


Figure 4.11: m-SportGIS: Technician ID (“ID Técnico”) View.

Listing 4.2: Instantiation of Toolbar and Save Button.

```
// DataCollectingView :
// Toolbar (header) :
topToolbar = Ext.create('Ext.Toolbar', {
    docked: "top",
    zIndex: 1,
```

```

        items: [{
            xtype: 'button',
            text: 'Guardar',
            ui: 'action',
            handler: this.saveButtonTap,
            scope: this
        }]
    });
    // Button event action :
    saveButtonTap: function() {
        var tmpThis = this;
        Ext.Msg.confirm("Guardar:", "Deseja guardar o registro?", function(
            value) {
            if (value == 'no') return;
            else {
                tmpThis.fireEvent("stopwatchingposition", tmpThis);
                tmpThis.fireEvent("savefacility", tmpThis);
            }
        });
    }

    // stops watching position function (MapController) :
    stopwatchingposition: function(options) {
        if (watchID != null) {
            navigator.geolocation.clearWatch(watchID);
            watchID = null;
        }
    }

    // FacilityController :
    savefacility: function(options) {
        var currentfacility = this.getDataCollectingView().getRecord();
        this.getDataCollectingView().updateRecord(currentfacility);
        currentfacility.set('submitted', false);
        var errors = currentfacility.validate(), message = '';
        if (!errors.isValid()) {
            errors.each(function(err) {
                message += err.getMessage() + '<br/>';
            }); // each()
            new Ext.MessageBox().alert('Registro não pode ser guardado',
                message);
        } else {
            // if valid = safe
            if (null == Ext.getStore('facilityStore').findRecord('id',
                currentfacility.get("id"))) {
                Ext.getStore('facilityStore').add(currentfacility);
            } else {
                currentfacility.setDirty();
            }
            Ext.getStore('facilityStore').sync();
        }
    }
}

```

```

        Ext.getStore('facilityStore').sort([
            property: 'date',
            direction: 'DESC'
        ]);
        // clear view
        this.getDataCollectingView().clearView();
        // update map
        SportApp.app.getController("MapController").showallfacilities();
        // return to last active view
        if (mode == 1) {
            this.showstartview('right');
        } else {
            this.showlistview();
        }
        // save images
        this.saveimages();
    }
}

```

The data form is divided into five field tabs (groups) — “ID Técnico”, “ID Instalação”, “Modalidades”, “Características”, “Informações” — accessed tapping the different tab buttons according to the type of data users want to fill up. The data form comprises 150 different fields among personnel and sports facility-related information; its structure can be consulted in Tabela A.4 (in Appendix A). This view shows “ID Técnico” FieldSet where data about personnel associated with the data collection operations is introduced. These items hold direct alphanumeric type inputs; Listing 3.1 shows a portion of written code to build two of those fields.

The coordinates form includes Lon, Lat and accuracy values, gathered from built-in GPS sensor when “Localização” button is pressed. This button event was created taking the geolocation object and methods from Cordova API; location-based instantiations declared in View and Controller can be seen in the portion of code in Listing 4.3.

Listing 4.3: Instantiations of Location Objects

```

// ---> DataCollectingView (creates button and defines button event) :
setPositionButton = Ext.create('Ext.Button', {
    text: 'Localizar',
    ui: 'action-round',
    iconCls: 'locate',
    iconMask: true,
    margin: "20 0 0 10",
    height: 30,
    disabled: true,
    handler: this.setPositionButtonTap,
    scope: this,
    style: 'font-size:13px;'
});
// button event action :
setPositionButtonTap: function() {

```

```

        this.fireEvent("setPosition", this);
// updates location fields in current record :
changeFormPosition: function(lon, lat, acc) {
    var currentFacility = this.getRecord();
    this.updateRecord(currentFacility);
    currentFacility.set("coordinates", lat + ' ' + lon);
    currentFacility.set("accuracy", acc);
    this.setRecord(currentFacility);
}

// ---> MapController (controls button action) :
Ext.define('SportApp.controller.MapController', {
    extend: 'Ext.app.Controller',
    config: {
        control: {
            dataCollectingView: {
                stopwatchingposition: "stopwatchingposition",
                setPosition: "setPosition"
            }
        }
    },
    setPosition: function(options) {
        this.getDataCollectingView().changeFormPosition(
            coordinates.longitude, coordinates.latitude, coordinates.
            accuracy);
        SportApp.app.getController("MapController").
            changeFacilityPosition(
                coordinates.longitude,
                coordinates.latitude,
                map.getZoom()
            );
    }
}
}

```

The map object was set to be centered in Mozambique extension (by default), and includes a layer switcher object (to change between BaseLayers — WMS-C (“Camada Base”) and OSM). Map is navigated (pan and zoom in/out) through device multi-touch capability. Listing 4.4 shows portion of written code to instantiate the `mapPanel`; fully functions to initialize objects, such as the layer switcher and layers, take OpenLayers library (*e.g.* `OpenLayers.Control.LayerSwitcher`), and are presented in Listing B.9 (in Appendix B).

Listing 4.4: m-SportGIS Map Instantiation.

```

// Data Collecting View :
mapPanel = Ext.create('Ext.Panel', {
    width: Ext.Viewport.getSize().width / 2.2,
    docked: 'right',
    margin: "5 0 0 5",
    zIndex: 0,

```

```

        items: [{
            // component where the Map is docked :
            xtype: "component",
            margin: "-20 0 0 0",
            width: Ext.Viewport.getSize().width / 2 - 100,
            height: Ext.Viewport.getSize().height - 220,
            scrollable: false,
            monitorResize: true,
            // Map component on Data Collecting View :
            id: "mapComponentDCV",
            listeners: {
                painted: function() {
                    SportApp.app.getController("MapController").
                        initializeMap(
                            this.id
                        );
                }
            }
        }]
    });

// Map Controller :
initializeMap : function(componentDIV) {
    // content/layers instantiations (online and offline/tiles)
    // note : function defined in MapController
}

// Facility Controller :
showdatacollectingview: function(options) {
    this.getView().animateActiveItem(
        this.getDataCollectingView(), {
            type: 'slide',
            direction: 'left'
        });
    SportApp.app.getController("MapController").dockmap("
        mapComponentDCV");
}

```

The button “Ver Imagens” fires a Panel (pictureView) showing pictures associate with the current record; record and image are linked through unique IDs; record ID is created each time a new record is instantiated, and picture ID when a record is saved with picture(s) attached to it. Listing 4.5 presents button and function code to build the pictureView.

Listing 4.5: m-SportGIS Show Picture Function.

```

// Data Collecting View :
// instantiating ToolBar and showPicture button :
bottomToolbar = Ext.create('Ext.Toolbar', {
    docked: 'bottom',
    items: [

```

```

        {
            xtype: "button",
            text: 'Ver Imagens',
            ui: 'action',
            handler: this.showPicture,
            scope: this
        }
    ];

    // record and picture indexed :
    var activePictureIndex;

    // showpicture function :
    showPicture: function() {
        activePictureIndex = 0;
        var data = Ext.getStore('ImageStore').getRange();
        if (data.length == 0) return;
        var item = data[0];
        // remove current image
        this.pictureView.removeAll();
        // show previous image
        this.pictureView.add({
            xtype: 'container',
            width: Ext.Viewport.getSize().width / 2 - 20,
            height: Ext.Viewport.getSize().height / 2 - 60,
            html: '<CENTER>'
        });
        // this.pictureView.doLayout();
        SportApp.view.DataCollectingView.prevPictureButton.enable();
        SportApp.view.DataCollectingView.nextPictureButton.enable();
        if (activePictureIndex == 0) {
            // disable prev button
            SportApp.view.DataCollectingView.prevPictureButton.disable();
        }
        if (activePictureIndex == data.length - 1) {
            // disable next button
            SportApp.view.DataCollectingView.nextPictureButton.disable();
        }
        // displays picture object on Viewport :
        Ext.Viewport.add(this.pictureView);
        this.pictureView.show();
    }
}

```

Listing 4.6 shows button and function code to handle the access to device photo album (using Cordova API).

Listing 4.6: m-SportGIS Show Camera Function.

```

// Data Collecting View :
// instantiating ToolBar and showcamera button :

```

```

bottomToolbar = Ext.create('Ext.Toolbar', {
    docked: 'bottom',
    items: [
        {
            xtype: "button",
            iconCls: 'btnAddPic',
            text: 'ÃAlbum',
            ui: 'action',
            handler: this.galleryButtonTap,
            scope: this
        }
    ]
});

// album button event :
galleryButtonTap: function() {
    this.fireEvent("showcamera");
}

// Facility Controller :
// showcamera function :
showcamera: function(options) {
    // number of pictures condition (max number = 5)
    if (Ext.getStore('ImageStore').getRange().length > 4) {
        Ext.Msg.alert('', 'MÃximo de 5 imagens permitidas!');
        return;
    } else {
        var sourceType = Camera.PictureSourceType.PHOTOLIBRARY;
        //var pictureSource=navigator.camera.PictureSourceType;
        //var sourceType = Camera.PictureSourceType.SAVEDPHOTOALBUM;
        // phoneGap API format : navigator.( cameraSuccess,
        cameraError, [ cameraOptions ] );
        navigator.camera.getPicture(onSuccess, onFail, {
            quality: 50,
            destinationType: Camera.DestinationType.FILE_URI,
            sourceType: sourceType,
            mediaType: Camera.MediaType.PICTURE
            //saveToPhotoAlbum: false
        });

        function onSuccess(imageURI) {
            //console.log("camera onSuccess");
            // creates image ID
            var now = new Date();
            var newId = now.getTime();
            // var imName = newId + '.jpg';
            var imagePath = imageURI;
            // add picture to database
            // get current record to get correspondent ID
            var currentfacility = SportApp.app.getController("
                facilityController").getDataCollectingView().

```

```

        getRecord();
        SportApp.app.getController("facilityController").
            getDataCollectingView().updateRecord(currentfacility)
        ;
        // create entry for the image
        var imageEntry = Ext.create("SportApp.model.ImageModel",
            {
                imageID: newId,
                facilityID: currentfacility.get("id"),
                path: imagePath,
                saved: false
            });
        // save entry
        Ext.getStore('ImageStore').add(imageEntry);
        Ext.getStore('ImageStore').sort([{
            property: 'facilityID',
            direction: 'DESC'
        }]);
    }

    function onFail(message) {
        // alert('Failed because: ' + message);
        //console.log("camera onFail");
    }
}
}

```

Figure 4.12 presents “ID Instalação” field group to insert data related to sports facilities; some of these fields involved special structures — nested data lists, *i.e.* drop-down selection menus with a hierarchical tree structure. This was employed to the fields tree Province - has many - District - has many - Town. Long data (field : value) structures are better organized in stores, which handle data in JSON data-interchange format to feed nested lists. Listing D.1 (in Appendix D) presents the built DistrictsStore.

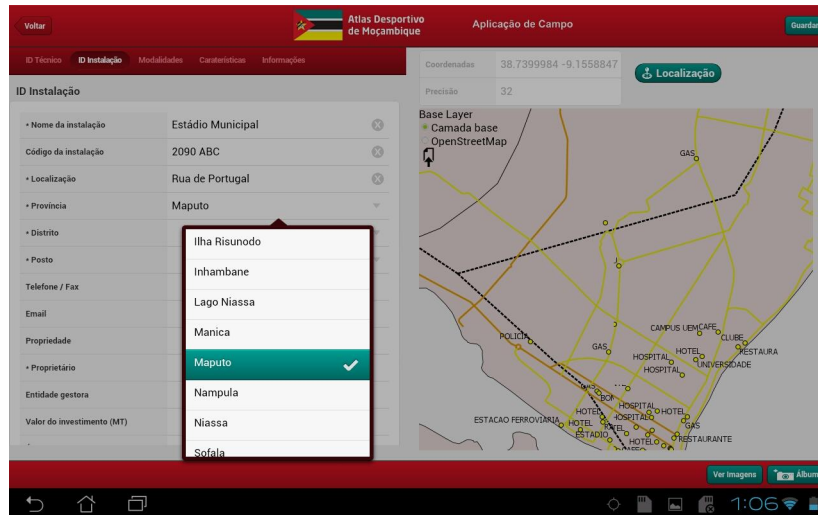


Figure 4.12: m-SportGIS: Facility ID (“ID Instalação”) View and Select Field.

This view highlights on the Viewport the Province item which include a selectfield list of Provinces (values). Its Districts and Towns values were organized in DistrictsStore and TownsStore, respectively. When the user chooses a Province (e.g. value : Maputo), an internal built filter is applied over the selected value, and Districts-child-values are filtered; consequently, the filter acts over these Districts-child-values, and Towns-child-values are filtered. In the items District and Town only filtered values from the stores are returned. Listing B.3 (in Appendix B) presents written code to build this functionality.

The vectorial map presented shows zoomed tiled data in a Maputo area.

Figure 4.13 shows over the Viewport an example of an item with a multi-list-overlay class defined to handle nested content (field : value) on child panels. Listing B.5 (in Appendix B) shows code written to create this functionality. Overlays panels were set up to be fired up when user taps on form fields containing nested data. Ok and “Cancel” buttons confirm or cancel values inputed, respectively, and close their overlay panel. When the user taps out of the overlay panels, they are also folded. To control the values selected by the user on the different panels, a multi-list-overlay-item class was defined (Listing B.6 in Appendix B).

Figure 4.14 depicts “Modalidades” field group related with sports activities. The two overlay form panels highlighted contain, in the first, a selectfield list of sports activities (Checkbox property declared in Listing B.6), and in the second, information of their practitioners. Sports activities shown on the first overlay panel are able to be multi-selected (Listing B.4 in Appendix B), and for each one of these, a form panel is fired to input its number of practitioners (textfield). The on-screen keyboard is popped-up whenever users touch within a text entry field.

Figure 4.15 shows a field group — “Características” — with technical information about sports facilities, such as field dimensions. Every field (activities shown on first

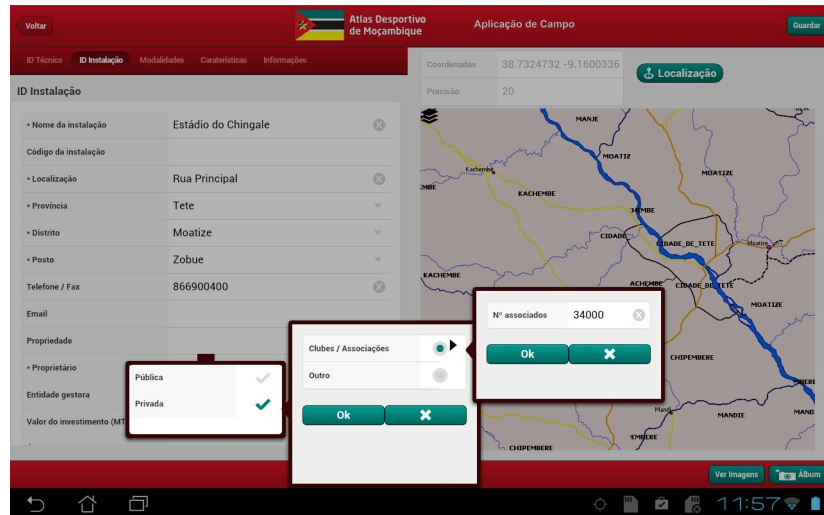


Figure 4.13: m-SportGIS: MultiListOverlay and MultiListOverlayItem Classes.

form panel highlighted) that holds nested data fires an event when the user taps on Checkbox. The second overlay panel that is displayed includes a form field (textfield and numberfield) with facility dimensions and category.

Figure 4.16 displays the last form tab (“Informações”) with general information about the sports facility, such as the public access (infrastructures) conditions, and its capacity (seats). It includes items with selectfield lists, Checkbox and numberfield formats.

m-SportGIS holds a FacilityModel with a list of allowed fields to be stored and their formats, and validations (rules) to ensure data is saved according to designated specifications (*e.g.* numberfield). Listing B.7 depicts the code that addresses these conditions. Figure 4.17 highlights a data validation alert fired up at the moment user saves the current record, listing the inserted data that is not according to defined validation rules in the Model.

Figure 4.18 depicts the data list view UI with four different areas: a toolbar, a panel with a list of saved data records, an order-by record field, and a map panel displaying saved data records locations. The header Toolbar contains a back to start view button (“Voltar”), the app logotype and name, and a submit all data button (“Submeter tudo”) handling a function (submitAll) that was created to push all data records to the server database. This function calls two others created main functions (sendAlphaAll and sendImgAll) which push to the database alphanumeric data and images (jpg), respectively. Each of these functions has an onsuccess and onfailure submission function, which shows a message to the user accordingly to that event status. These functions can be perceived in the code written in the FacilityController (Listing B.8 in Appendix B).

The list of saved records displays every records by order of the attribute chosen by the user which is done in the order-by record field. This list of attributes was established based on the sports field set created in the DataCollectingView component. When the

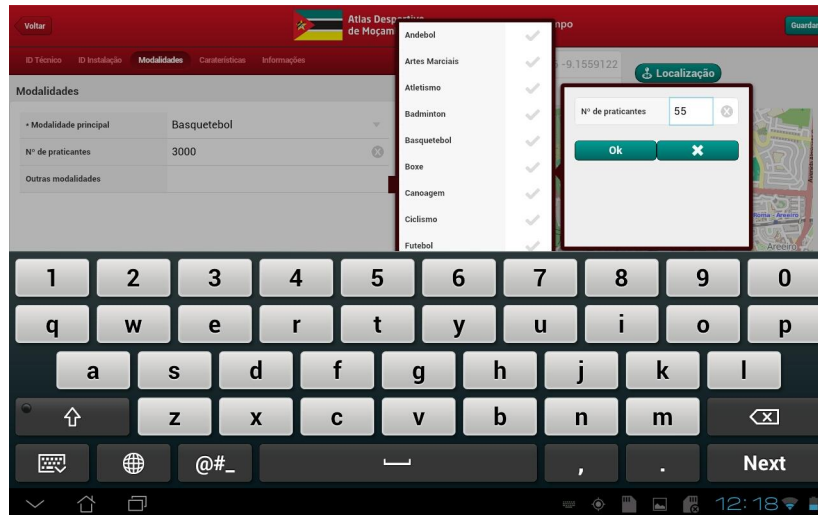


Figure 4.14: m-SportGIS: MultiListOverlay Class in Sports Activities (“Modalidades”) Field Set.

user taps on a record, its location is centered and highlighted (colored cross) on the map. If there is more than one facility on the same location, the cross discloses the number of facilities on that point. Each record contains an individual-record submit to server database button, and an edit data button (go to [DataCollectingView](#)). This individually submission button handles an event similar to the [submitAll](#) function, as can be seen in [FacilityController](#). After a succeeded submission, the button icon changes to a “check” button icon.

Listing 4.7: Submit all data records function

```
submitAll : function(options) {
    Ext.Viewport.setMasked({
        xtype: 'loadmask',
        message: 'A enviar registos... Por favor aguarde.'
    });

    fileStore = Ext.getStore('facilityStore'),
    allData = []; // var holding picture ID in records
    imgURI = []; // var holding path to the picture
    allData2 = []; // var holding all alphanumeric data
    imagID = []; // var holding picture ID
    fileStore.filter('submitted', false);
    fileStore.each(function(record) {
        allData.push(record.get('id'));
        var currentImage = Ext.getStore('ImageStore');
        currentImage.filter('facilityID', allData);
        var rangeImage = Ext.getStore('ImageStore').getRange();
        numImage = rangeImage.length;
        uploadedImages = 0;
        allData2.push(record.getData());
        if (numImage > 0) {
```

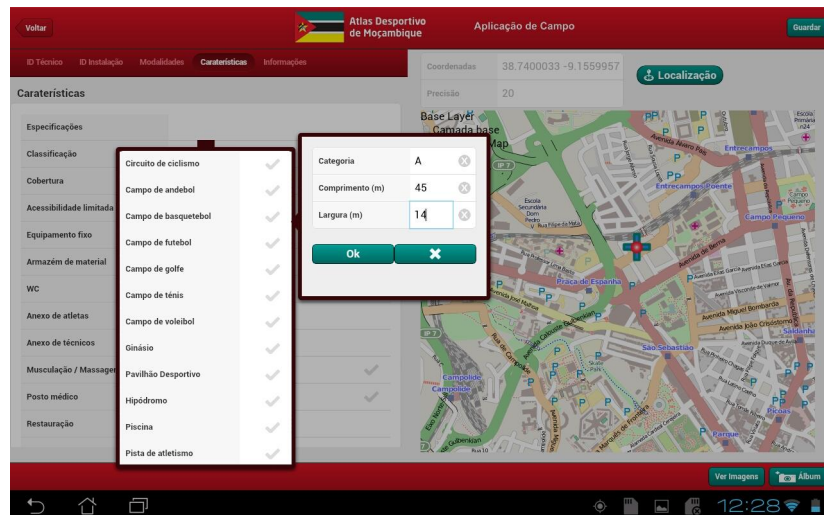


Figure 4.15: m-SportGIS: Technical Features (“Características”) View.

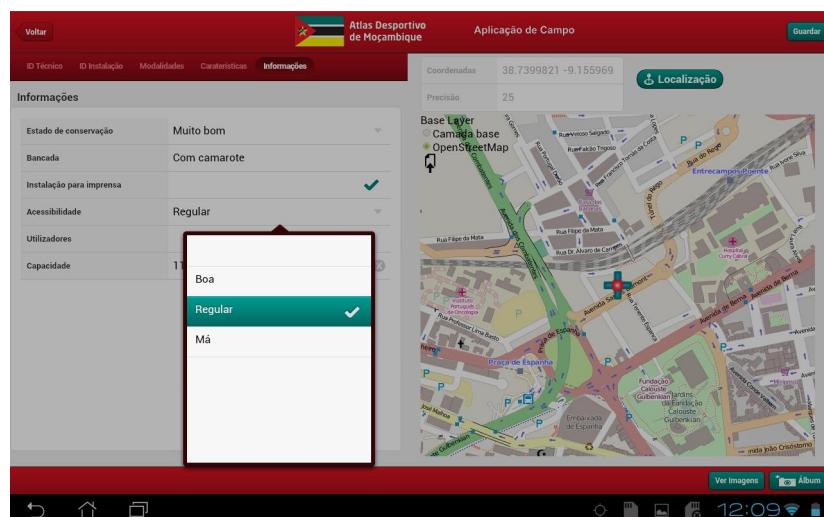


Figure 4.16: m-SportGIS: Informations (“Informações”) View.

```
// --- > start pictures sending cycle
currentImage.each(function(record) {
    me.sendImgAll(record);
});
} // ends cycle
if (numImage == 0) {
    me.sendAlphaAll();
    Ext.Msg.alert('Envio de dados', 'Dados submetidos para o
        servidor.');
```

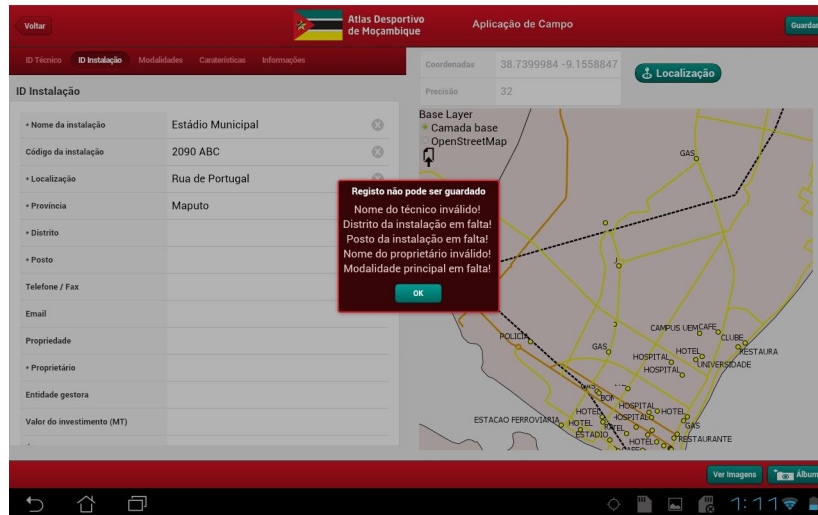


Figure 4.17: m-SportGIS: Example of Record Validation Alert.

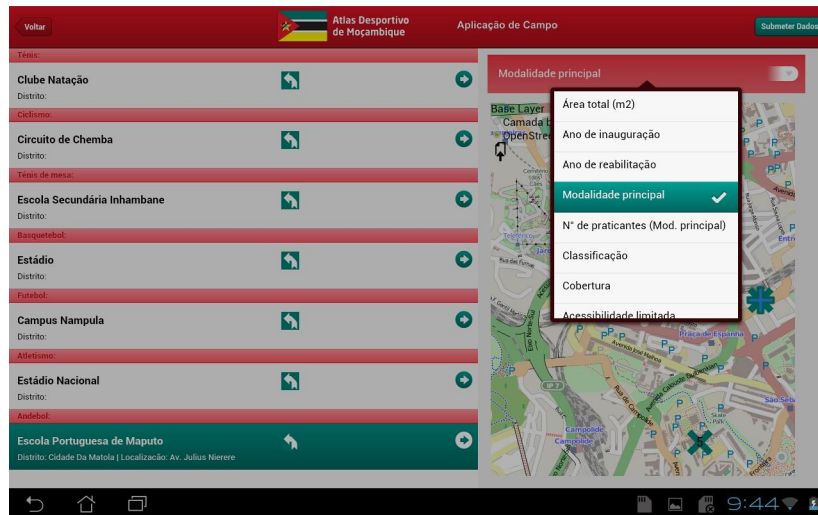


Figure 4.18: m-SportGIS: List of Saved Records and Spatial Location View.

Figure 4.19 shows the view exhibited to users when access to previously saved data. This view is very alike to the `DataCollectingView`, and users can edit all kind of information, except the location data. The coordinates data form comes out with no GPS functionality.

Figure 4.20 displays the on top of the data edition view a container including the pictures saved in the ImageStore (local storage). The `Toolbar` at the bottom of this container holds tree buttons — a previous picture button (`previousPictureButtonTap`), a delete picture button (`deletePictureButtonTap`), and a next picture button (`nextPictureButtonTap`). Listing B.10 shows code written to create those buttons events functions.

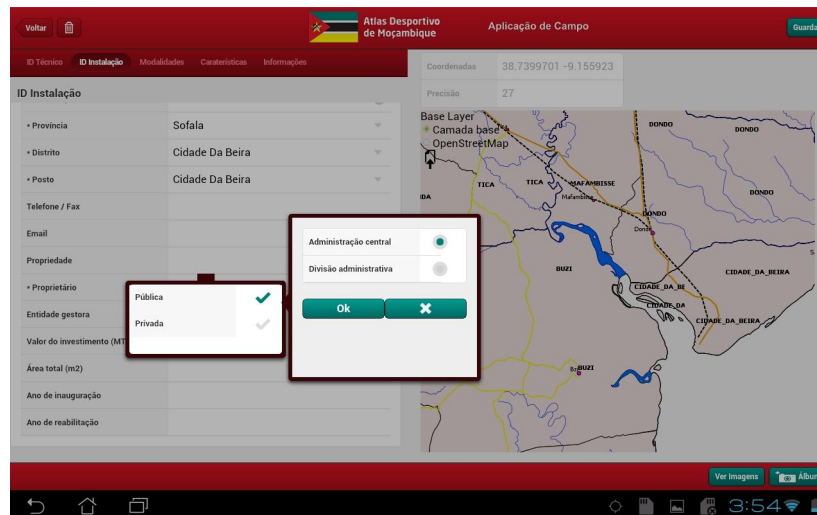


Figure 4.19: m-SportGIS: Data Edition (Saved Record) View.

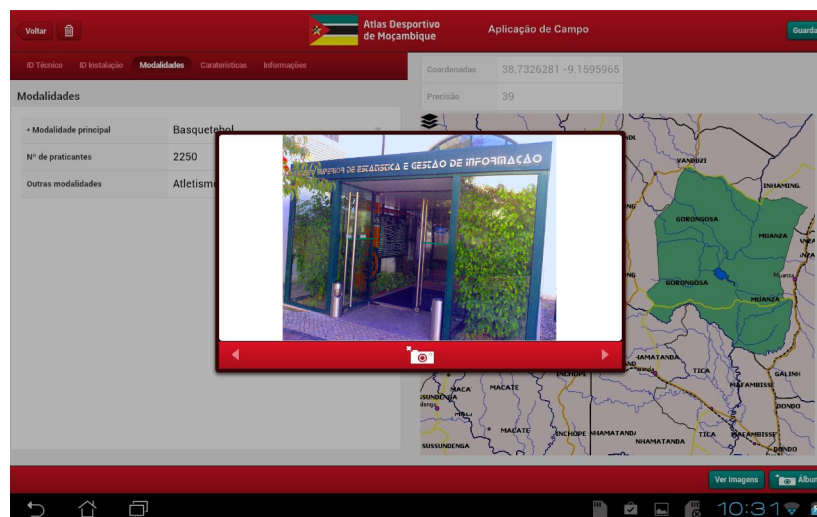


Figure 4.20: m-SportGIS: Saved Photo Indexed to a Record View.

4.4.3.3 m-SportGIS as an Android Application

To implement m-SportGIS as a natively-installed Android app, the IDE (Eclipse v 4.2.2) was configured with Android SDK Tools (including Platform-tools and Android 4.0.3 - API 15), Android Development Tools (ADT) plugin (v 21), and Cordova/PhoneGap (v 2.5.0). ADT extends the IDE's capabilities, creating an app UI, adding API-based packages, app debugging tools, and exporting native files (.apk) in order to distribute the app.

Android Project Structure

A new Android OS project was created in Workspace, configuring the compiler builder to target the Android 4.0.3 and later versions. m-SportGIS and pt.unl.isegi.labnt

.m-sportgis were the given app name and package name (namespace), respectively. To configure the project to use Cordova library, assets/www directory and libs directory were created inside the Android project; Cordova HTML and JS files will reside within assets/www folder (such as OpenLayers.js compiled JS file) as shown in Figure 4.22. cordova-2.5.0.js was copied to the assets/www folder, cordova-2.5.0.jar to the libs directory and xml into the res directory, as depicted in Figure 4.21.

index.html was created inside assets/www folder with the content shown in Listing B.11, and was the main entry point for the Cordova app's interface (run on Chrome). cordova-2.5.0.jar library was added to the Java build path of the Android project (Android dependencies).

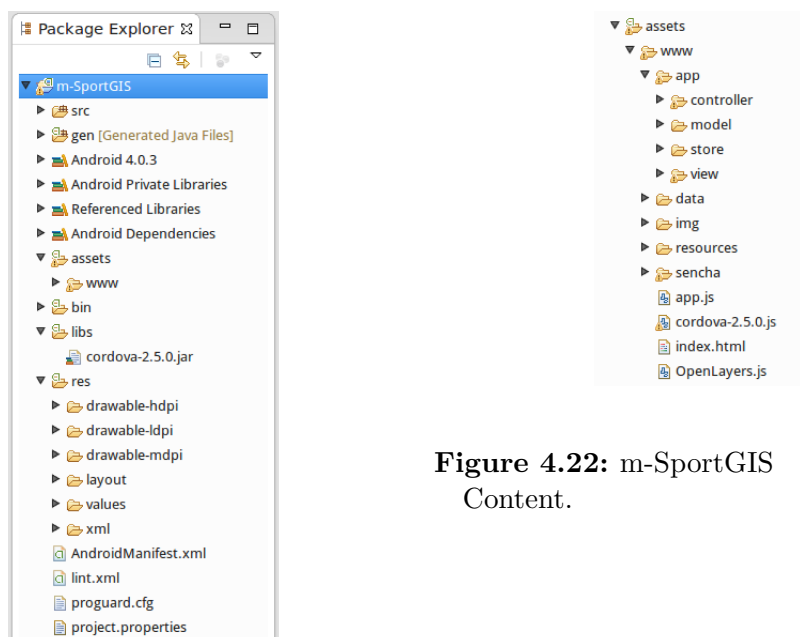


Figure 4.22: m-SportGIS assets/www Content.

Figure 4.21: m-SportGIS Package Folder Structure.

Cordova Configuration

In order to set up Cordova to be launched in the project, a Java statement (Listing B.12) was declared in the Activity file inside the project package pt.unl.isegi.labnt.m-sportgis.

Project Metadata

Finally, Cordova required metadata in order to run on the Android project. This was done by adding XML content — screen sizes, Android features permissions, activity configuration changes, DroidGap class declaration — into the AndroidManifest.xml file, as shown in Listing B.13.

Running m-SportGIS

In the development environment, m-SportGIS.apk was launch/created running the project under the option “Run As Android App”. The file produced was located inside bin directory. The fully functional app was tested on the physical devices available (Asus and Samsung) before being deployed into the production/delivery environment.

4.5 WebGIS Platform

SportGIS WebGIS constitutes an e-government platform and is being created to distribute and process spatial information. Its development and implementations are not scope of this project, although examples of the platform functionalities are presented bellow, showing a navigation over sports facilities spatial distribution (Figure 4.23), a statistical/demographical analysis (Figure 4.24), spatial analysis (Figure 4.25), and a type of report output (PDF) (Figure 4.26).

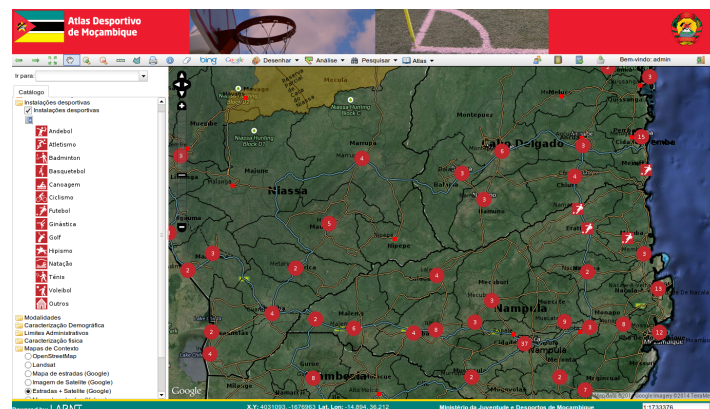


Figure 4.23: WebGIS: Displaying Sports Facilities Spatial Distribution.

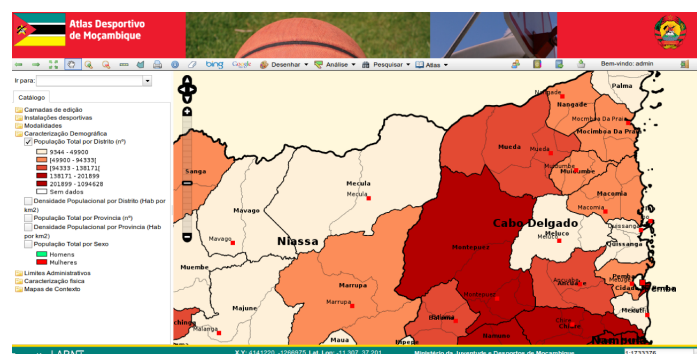


Figure 4.24: Displaying Statistical/Demographical Analysis.

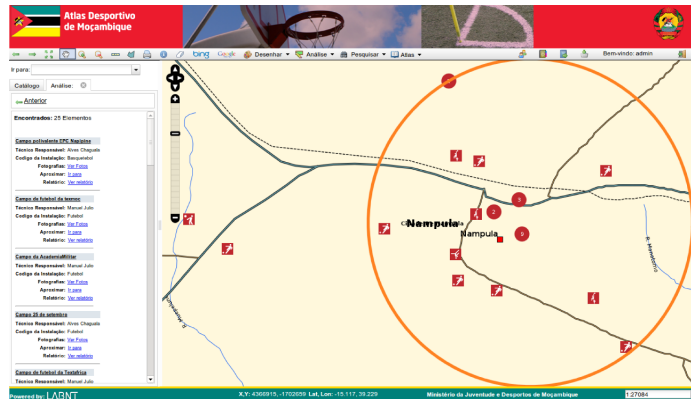


Figure 4.25: Displaying Sports Facilities Spatial Analysis.

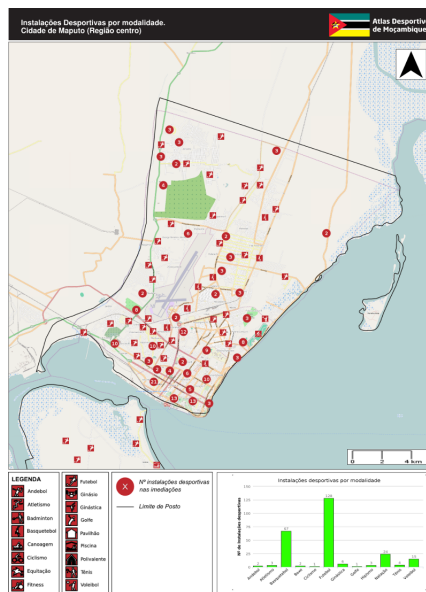


Figure 4.26: Displaying Outputted Report.

4.6 Considerations

Taking advantage of existing OS resources ST, OpenLayers, and Cordova, and programming languages (*e.g.* JS) it was produced a low-cost, customized, and scalable mobile web-based mapping app for real-time collection of sports field data. Along the development and implementation were produced tree versions of m-SportGIS being v 1.2 the last stable version. After testing phases under the Android devices, and regarding the app's responsiveness, the functionality to access the device camera inside the app Viewport was disabled; it is desirable users grab pictures out of the app's environment, and call/select them later inside the app. Attending to Google Maps license and client final intentions (possibility to print maps), on the MapPanel view only OSM and WMS-C (tiles) base layers were made visible. At this moment, Mozambican Gov-

ernment is being exploring successfully m-SportGIS in the creation of sports facilities data to produce and share very useful information to sports stakeholders and global communities.

4.7 Usage of m-SportGIS and Statistical Results

m-SportGIS implementation constitutes a successful exploration of a low-cost mapping solution. 28 members of the governmental staff are taking advantage of its capabilities to build a national sports inventory. Figure 4.27 depicts the number of collected facilities by month (in the period considered were collected 1062 facilities) gathered by that team. This statistical information was produced from the data in the spatial database (server). After the field works the sports information was made available on the WebGIS almost instantly. This technology permits to have access to “near” real-time information about sports resources. This capacity makes possible visualize, analyze, and produce information very efficiently. Figure 4.28 shows a more detailed distribution of sports facilities per type of main activity (*e.g.*, football) by province. This software exploration represents a great benefit inside Mozambican Government in terms of resources (capital and human) savings, when the organization undertakes operations of data collection and post-processing.

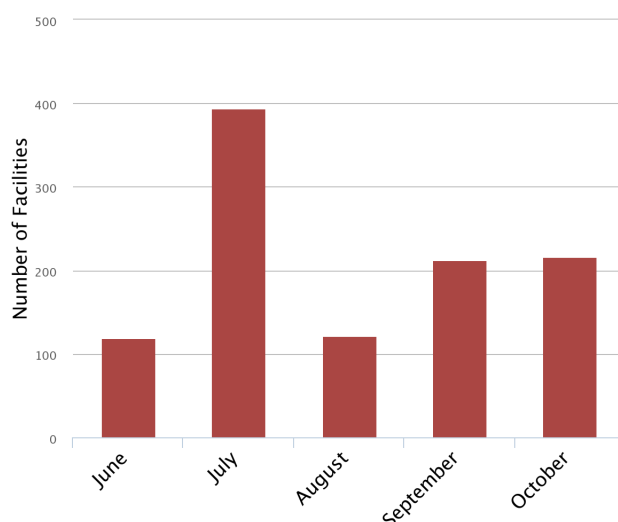


Figure 4.27: Number of Collected Facilities by Month.

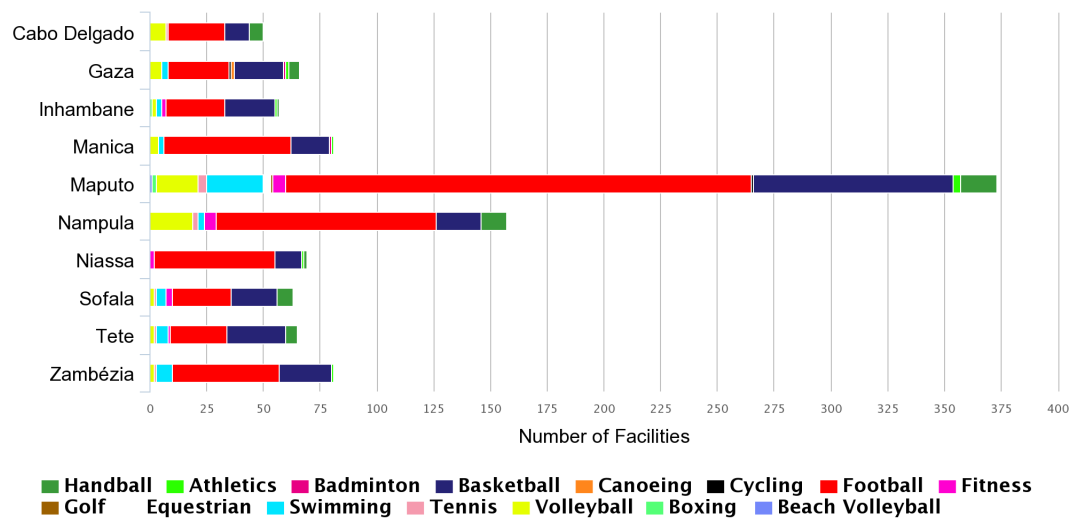


Figure 4.28: Number of Collected Facilities per Type of (Main) Sports Activity by Province.

5 Conclusions

5.1 Synthesis

Mobile mapping tools represent, certainly, very convenient vehicles to overcome the substantial bottleneck of available spatial and attribute data in developing countries. Sports data production in Mozambique, and in other developing countries contribute substantially to improve public and private services, and governmental resources management; consequently, this fact improves the well-being of communities in those countries. A low-cost solution to produce an inventory with very rich data constitutes a crucial resource to Mozambican Government, particularly to MJD staff, whom are looking to improve their participation in the sustainable development and management of sports infrastructures, providing the right facilities in the right places. m-SportGIS is creating a great impact on the way MJD staff works, since it is improving and optimizing human and capital resources in the production of data. OS solutions inside the governmental sector are justified and beneficial, since governments can use OS software as a vehicle for promoting economic development, based on the fact that governmental staff community enthusiasts can contribute to the OS solution development and maintenance.

This work documents a succeeded analysis, design, development and implementation of an enterprise low-cost location-aware mobile software app for geospatial data collection. This mapping solution was developed using low-cost resources, and was created in response to the need of valuable georeferenced data to be an instrument in the planing and management of sports facilities and infrastructures in Mozambique. m-SportGIS offers a customized UI with a complete set of sports attributes and map interaction. Beyond its online functionalities (online map and data transmission), it provides offline capabilities to work network-communication-independently (local storage and cached map service).

m-SportGIS development carried out a previous analysis of software development in organizations, which addressed to the adoption of a combined methodology based on a prototyping ISDM. Its architecture was build under a MVC pattern to provide a more scalable solution. Regarding the mobile system specifications, available resources, and the author technical background, the app's construction was taken under a hybrid development, *i.e.* using web programming languages and frameworks, and a native API to control mobile device native functionalities/behaviors. The success of its im-

plementation demonstrates that mobile UX and UI can take advantage of OS hybrid developments. Current JS APIs proved to be powerful means to reach device capabilities such as File system, Geolocation, and Camera. Android devices appealed to be very suitable (and low-cost) tools for GIS mapping. Tile caching mechanisms proved to be valuable means to improve UX in all scenarios (online, offline) with customized vectorial data.

The app is currently available to the MJD and is being exploited by its staff; its success can be noticeable in the large amount of data already available to be analyzed on the SDI.

5.2 Contributions

To achieve the objects of m-SportGIS project, features have been developed as depicted along chapter 4. The app development phases implicated the production of new features (UI components) and data (customized sports facilities and activities form with a tree structure, and map tiles). The outputted IT product is available to Mozambican Government staff.

5.3 Further Developments

Over the project development some approaches were taken: Android operating system, MVC hybrid development with Touch (including HTML5 storage feature) and Cordova, and WMS-C service implementation with TileCache.

Being a software app, it is flexible and expandable to accommodate new functionalities or reformulations, in order to improve its performance and capabilities. Although the implemented solution already presents the necessary functionalities, future work can include research of other OS platforms and tools, and plausible features discerned are the extending spatial capability to edit geometry such as polygons and lines, and the implementation of a client-side spatial database (*e.g.* SpatiaLite) capable of store and update data between client-side and a server-side. Regarding the caching mechanism adopted, others approaches can be undertaken to create other content with different details.

5.4 Dissemination of Project's Research and Results

At this moment the development of m-SportGIS was documented in a full paper accepted to be presented in the 2014 Conference on ENTERprise Information Systems, and published as an article in *Procedia Technology Journal*, Elsevier.

References

- Aanensen, D. M., Huntley, D. M., Feil, E. J., Al-Own, F., & Spratt, B. G. (2009, January). EpiCollect: linking smartphones to web applications for epidemiology, ecology and community data collection. *PloS one*, 4(9). Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2735776&tool=pmcentrez&rendertype=abstract> doi: 10.1371/journal.pone.0006968
- Alavi, M., & Leidner, D. (2001). Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly*, 25(1), 107–136.
- Anokwa, Y., Hartung, C., & Brunette, W. (2009). Open Source Data Collection in the Developing World. *Computer*, 42(10), 97–99. doi: <http://doi.ieeecomputersociety.org/10.1109/MC.2009.328>
- Apache Cordova. (2013). *Apache Cordova*. Retrieved 26 October 2013, from <http://cordova.apache.org>
- Appcelerator. (2013). *Case Study: imgZine*. Retrieved 18 October 2013, from <http://www.appcelerator.com/titanium/>
- AppFoundation. (2013). *HTML5*. Retrieved 07 September 2013, from www.appfoundation.com/html5
- Avison, D., & Fitzgerald, G. (2006a). *Information Systems Development: Methodologies, Techniques and Tools* (4nd Editio ed.). McGraw-Hill.
- Avison, D., & Fitzgerald, G. (2006b). Methodologies for Developing Information Systems: A historical perspective. In (Vol. 214, pp. 27–38). Boston Springer.
- Balaji, S., & Murugaiyan, M. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26–30.
- Battersby, S. E. (2012). *Serving Your Map*. Retrieved 5 December 2013, from <http://giscollective.org/tutorials/web-mapping/wmsseven/>
- Berger, S., Mcfaddin, S., Narayanaswami, C., & Raghunath, M. (2003). Web Services on Mobile Devices: Implementation and Experience. In *Fifth ieee workshop on mobile computing systems & applications*. IEEE Computer Society.
- Bertot, J. C., Jaeger, P. T., Simmons, S. N., Grimes, J. M., & Shuler, J. A. (2009). Reconciling government documents and e-government: Government information in policy, librarianship and education. *Government Information Quarterly*, 26(3), 433–436.

- Bohicchio, M., & Fiore, N. (2004). WARP: Web Application Rapid Prototyping. In *Proceedings of the 2004 acm symposium on applied computing* (pp. 1670–1676). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/967900.968232>
- Boston Technology Corporation. (2013). *Native vs. Hybrid vs. Mobile Web Apps* (Tech. Rep.). Marlborough, MA, USA: Boston Technology Corporation. Retrieved from www.boston-technology.com
- Boston Technology Corporation. (2014). *Hybrid Apps: The Future of Enterprise Apps* (Tech. Rep.). Marlborough, MA, USA: Boston Technology Corporation. Retrieved from www.boston-technology.com
- Buckingham, R. A., Hirschheim, R. A., Land, F. F., & Tully, C. J. (1986). Information Systems Education: Recommendations and Implementation. In (pp. 14–133). Cambridge University Press.
- Budiu, R. (2013). *Mobile: Native Apps, Web Apps, and Hybrid Apps*. Retrieved 7 December 2013, from <http://www.nngroup.com/articles/mobile-native-apps/>
- Cadima, J., Fernandes, H., Viseu, J., Parente, F., & Brito, M. (2002). *Planeamento e Desenvolvimento Desportivo Municipal* (Tech. Rep.). Universidade do Minho. Retrieved from <http://hdl.handle.net/1822/2780>
- Casademont, J., Lopez-Aguilera, E., Paradells, J., Rojas, A., Calveras, A., Barceló, F., & Cotrina, J. (2004, July). Wireless technology applied to GIS. *Computers & Geosciences*, 30(6), 671–682. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0098300404000524> doi: 10.1016/j.cageo.2004.02.004
- Charland, B. A., & Leroux, B. (2011). Mobile Application Development : Web vs . Native. *ACM*, 54(5), 49–53. doi: 10.1145/1941487
- Chen, H., & Xiao, K. (2011, June). The design and implementation of the geological data acquisition system based on mobile GIS. In *19th international conference on geoinformatics* (pp. 1–6). IEEE Xplore. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980829> doi: 10.1109/GeoInformatics.2011.5980829
- Clegg, P., Bruciatelli, L., Domingos, F., Jones, R., De Donatis, M., & Wilson, R. (2006, December). Digital geological mapping with tablet PC and PDA: A comparison. *Computers & Geosciences*, 32(10), 1682–1698. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0098300406000604> doi: 10.1016/j.cageo.2006.03.007
- Cothran, J. (2010). *TileCache Notes*. Retrieved 11 March 2013, from <https://code.google.com/p/xenia/wiki/TileCache>
- Cowart, J. (2013). *When to Go Native, Mobile Web or Cross-Platform/Hybrid*. Retrieved 30 November 2013, from <http://tech.pro/blog/1355/when-to-go-native-mobile-web-or-cross-platformhybrid>
- Cunha, L. (1997). *O Espaço e o Desporto: Estudo da Acessibilidade ao Desporto na*

- Sub-Região do Vale do Tejo: Constituição de um Modelo de Análise* (Unpublished doctoral dissertation). Faculty of Human Kinetics, Technical University of Lisbon.
- Dahiya, D., & Dahiya, S. (2010). Role of Software Process Models in Enterprise System Development. In *9th roedunet iee international conference* (pp. 134–138). IEEE Xplore.
- Dangermond, J. (1997). Opening Lecture Synergy of Photogrammetry , Remote Sensing , and GIS. In D. Hobbie and D. Fritsch (Ed.), (pp. 11–16). Heidelberg.
- Davenport, T. H., & Prusak, L. (2000). Working knowledge: How organizations manage what they know. *Ubiquity*(August). Retrieved from <http://doi.acm.org/10.1145/347634.348775>
- David, M. (2012). *HTML5 Mobile Websites*. Elsevier.
- de Abreu Freire, C. E., & Painho, M. (2014). Development of a Mobile Mapping Solution for Spatial Data Collection using Open-Source Technologies. *Procedia Technology*, 00(00), 000 (in press).
- De Donatis, M., & Bruciatelli, L. (2006, June). MAP IT: The GIS software for field mapping with tablet pc. *Computers and Geosciences Geosciences*, 32(5), 673–680. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0098300405002001> doi: 10.1016/j.cageo.2005.09.003
- Devisch, O., & Veestraeten, D. (2013, April). From Sharing to Experimenting: How Mobile Technologies Are Helping Ordinary Citizens Regain Their Positions as Scientists. *Journal of Urban Technology*, 20(2), 63–76. Retrieved from <http://www.tandfonline.com/doi/abs/10.1080/10630732.2013.769313> doi: 10.1080/10630732.2013.769313
- Döner, F., & Yomral, T. (2008, July). Examination and Comparison of Mobile GIS Technology for Real Time Geo-Data Acquisition in the Field. *Survey Review*, 40(309), 221–234.
- Dong, C., Liu, F., Wang, H., & Chen, F. (2010, August). Application research of mobile GIS in forestry informatization. In *5th international conference on computer science & education* (pp. 351–355). IEEE. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5593615> doi: 10.1109/ICCSE.2010.5593615
- Donk, W., & Taylor, J. (2000). Geographic Information Systems (GIS) in public administration: An introduction to a series of articles. *Information Infrastructure and Policy*, 6, 127–129.
- Drucker, P. (1966). *The Effective Executive*. HarperBusiness Essentials.
- England, S. (2013). *Planning for Sport: Aims & Objectives* (Tech. Rep.). England: Sport England.
- ESRI. (2006). *What is a cached map service?* Retrieved 23 February 2013, from http://webhelp.esri.com/arcgisserver/9.2/dotnet/manager/publishing/static_map_svcs.htm

- ESRI. (2008). *What is map caching?* Retrieved 7 January 2013, from http://webhelp.esri.com/arcgisserver/9.3/dotNet/index.htm#what_is_map_caching.htm
- Firtman, M. (2013). *Programming the Mobile Web* (2nd Editio ed.; Simon Laurent, Ed.). O'Reilly Media.
- Fowler, M. (2012). *Developing Software for Multiple Mobile Devices* (Tech. Rep.). ThoughtWorks. Retrieved from <http://martinfowler.com/articles/multiMobile/>
- Gaffney, P. (2002). Developing a Statewide Emission Inventory Using Geographic Information Systems (GIS). In *U.s. epa annual emission inventory conference* (pp. 1–12). Atlanta, Georgia.
- Garcia, J., Moss, A. D., & Simoens, M. (2013). *Sencha Touch in Action* (F. Pohlmann, Ed.). Manning Publications Co.
- Gkatzoflias, D., Mellios, G., & Samaras, Z. (2013, March). Development of a web GIS application for emissions inventory spatial allocation based on open source software tools. *Computers & Geosciences*, 52, 21–33. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0098300412003512> doi: 10.1016/j.cageo.2012.10.011
- Goldstein, A. and Lazaris, L. and Weyl, E. (2011). *HTML5 & CSS3 for the Real World* (First Edit ed.; K. Steele, Ed.). SitePoint Pty. Ltd.
- Goodchild, M., Fu, P., & Rich, P. (2007). Sharing Geographic Information: An Assessment of the Geospatial One-Stop. *Annals of the Association of American Geograhers*, 97(2), 250–266.
- Google Developers. (2012). *Optimize for mobile*. Retrieved 10 November 2013, from <https://developers.google.com/speed/docs/best-practices/mobile>
- Google Developers. (2013a). *Google Maps API Web Services*. Retrieved 10 September 2013, from <https://developers.google.com/maps/documentation/webservices/#WebServices>
- Google Developers. (2013b). *MVC Architecture*. Retrieved 27 October 2013, from http://developer.chrome.com/apps/app_frameworks.html
- Google Developers. (2013c). *Web Performance Best Practices*. Retrieved 10 November 2013, from https://developers.google.com/speed/docs/best-practices/rules_intro
- Green, D. R., & King, S. D. (2004). *Pencil Out, Stylus In: Geospatial Technologies Give Coastal Fieldwork a New Dimension*. Aberdeen, UK.
- GSDI Association. (2012). *The SDI Cookbook*. Retrieved 31 May 2013, from http://www.gsddocs.org/GSDIWiki/index.php/Main_Page
- Guadagno, L., VandeWeerd, C., Stevens, D., Abraham, I., Paveza, G. J., & Fulmer, T. (2004, November). Using PDAs for data collection. *Applied Nursing Research*, 17(4), 283–291. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0897189704000783> doi: 10.1016/j.apnr.2004.09.008

- Hales, W. (2013). *HTML5 and JavaScript Web Apps* (1st Editio ed.; Simon Laurent, Ed.). OâÃOtildeReilly Media.
- Hazzard, E. (2011). *OpenLayers 2.10 Beginner's Guide* (A. Kumar, Ed.). Birmingham, UK: Packt Publishing.
- Hirschheim, R., Klein, H., & Lyytinen, K. (1996). Exploring the Intellectual Structures of Information Systems Development: a social action theoretic analysis. *Accounting, Management and Information Technologies*, 6.
- Hochdörfer, S. (2013). *Offline strategies for HTML5 mobile applications*. Retrieved 24 November 2013, from <http://talks.bitexpert.de/zendcone13-html5-offline-apps/#/>
- Howcroft, D., & Carroll, J. (2000). A Proposed Methodology for Web Development. In *Ecis 2000 a cyberspace odyssey: Proceedings of the 8th conference on information systems*. University of Salford.
- IBM. (2010). *Creek Watch: Snap a picture. Save a stream*. Retrieved 29 June 2013, from http://www.ibm.com/smarterplanet/us/en/water_management/article/creek_watch.html
- Instituto Português do Desporto e Juventude. (2012). *Censo Geral de Instalações Desportivos*. Retrieved 06 October 2012, from http://www.municipio-portodemos.pt/userfiles/File/Menu/ServicosMunicipais/Gabinetedesporto/CandidaturaApoios/Censo_Instalacoes_Desportivas.pdf
- J. Drummond, R. Billen, E. João, D. F. (2007). *Dynamic and Mobile GIS: Investigating Changes in Space and Time* (CRC Press, Ed.). Taylor & Francis.
- John Wilson and Stewart Fotheringham. (2008). *The Handbook of Geographic Information Science*. Blackwell Publishing Ltd.
- Jones, R. R., McCaffrey, K. J. W., Wilson, R. W., & Holdsworth, R. E. (2004, January). Digital field data acquisition: towards increased quantification of uncertainty during geological mapping. *Geological Society, London, Special Publications*, 239(1), 43–56. Retrieved from <http://sp.lyellcollection.org/cgi/doi/10.1144/GSL.SP.2004.239.01.04> doi: 10.1144/GSL.SP.2004.239.01.04
- Kanhere, S. S. (2011, June). Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces. *12th International Conference on Mobile Data Management*, 3–6. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6068482> doi: 10.1109/MDM.2011.16
- Keith, J., Cederholm, D., Kissane, E., Design, R. W., & Marcotte, E. (2013). *Sass for Web Designers* (E. Kissane, Ed.). A Book Apart.
- Kienberger, S. (2001). Assessing the Vulnerability to Natural Hazards on the Provincial / Community Level in Mozambique: The Contribution of GIScience and Remote Sensing.
- Kim, J., Lee, S., Ahn, H., Seo, D., Seo, D., Lee, J., & Choi, C. (2013, January). Accuracy evaluation of a smartphone-based technology for coastal monitoring. *Measurement*, 46(1), 233–248. Retrieved from <http://linkinghub.elsevier>

- .com/retrieve/pii/S0263224112002515 doi: 10.1016/j.measurement.2012.06.010
- Klokkan Technologies. (2008). *Tiles à la Google Maps: Coordinates, Tile Bounds and Projection*. Retrieved 1 September 2013, from <http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/>
- Kosmaczewski, A. (2012). *Mobile JavaScript Application Development: Bringing Web Programming to Mobile Devices* (1st Editio ed.; S. Laurent, Ed.). O'Reilly Media.
- Kosmaczewski, A. (2013). *Sencha Touch 2 - Up and Running* (1st Editio ed.; S. Laurent and M. Blanchette, Ed.). O'Reilly Media.
- Kumar, A. (2011). *Sencha Touch Cookbook* (1st Editio ed.; M. Rajani & A. Sheikh, Eds.). Packt Publishing.
- Kumar, A. (2012). *Sencha MVC Architecture* (1st Editio ed.; R. Rajgor, N. Shetty, & A. Paiva, Eds.). Birmingham, UK: Packt Publishing.
- Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., & Choudhury, T. (2010). A Survey of Mobile Phone Sensing. *IEEE Communications Magazine*(September), 140–150.
- Li, C.-g., & Jiang, Y.-y. (2011, May). Development of mobile GIS system for forest resources second-class inventory. *Journal of Forestry Research*, 22(2), 263–268. Retrieved from <http://link.springer.com/10.1007/s11676-011-0160-1> doi: 10.1007/s11676-011-0160-1
- Li, H., Fei, L., Wang, H., & Li, Y. (2009). An Efficient Mechanism for Organizing and Indexing Tile Caches on Map Server. *IEEE Communications Magazine*(200724).
- Li, R. (1997). Mobile Mapping: An Emerging Technology for Spatial Data Acquisition. *Photogrammetric Engineering and Remote Sensing*, 63(9), 1085–1092.
- Lohr, S. (2011). *In a New Web World, No Application Is an Island*. New York Times. Retrieved 10 November 2013, from www.nytimes.com/2011/03/27/business/27unboxed.html?_r=0
- Longley, P., Maguire, D., Goodchild, M., Rhind, D. (2011). *Geographical Information Systems and Science* (3rd Editio ed.; John Wiley and Sons, Ed.). UK: Wiley.
- Luke Wroblewski. (2013). *Breaking Development: Beyond Mobile*. Retrieved 11 November 2013, from www.lukew.com/ff/entry.asp?1813
- Lwin, K. K. (2011). Web-based GIS System for Real-time Field Data Collection Using Personal Mobile Phone. *Journal of Geographic Information System*, 03(04), 382–389. Retrieved from <http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/jgis.2011.34037> doi: 10.4236/jgis.2011.34037
- Mahemoff, M. (2010). *Introduction: The Meaning of Offline*. Retrieved 29 October 2013, from <http://www.html5rocks.com/en/tutorials/offline/storage/>
- Matsudaira, K. (2013). Making the Mobile Web Faster. *ACM*, 1–9.
- MetaCarta Labs. (2010). *TileCache 2.11 Documentation*. Retrieved 13 January 2013, from <http://tilecache.org/docs/index.html>
- MetaCarta Labs. (2013). *TileCache - Web Map Tile Caching*. Retrieved 12 December 2012, from <http://tilecache.org>

- Middleton, P. (1999). Managing information system development in bureaucracies. *Information and Software Technology*, 41, 473–482.
- Montoya, L. (2003). Geo-data acquisition through mobile GIS and digital video : an urban disaster management perspective. *Environmental Modelling and Software*, 18, 869–876. doi: 10.1016/S1364-8152(03)00105-1
- Munassar, N. M., & Govardhan, A. (2010). A Comparison Between Five Models of Software Engineering. *International Journal of Computer Science Issues*, 7(5), 94–101.
- OpenGeo. (2013a). *Geoserver*.
- OpenGeo. (2013b). *GeoWebCache*. Retrieved 12 September 2013, from <http://geowebcache.org/docs/1.5.1/introduction/whatis.html>
- OpenLayers. (2013). *OpenLayers Resources*. Retrieved 23 March 2013, from <http://dev.openlayers.org/docs/index/General.html>
- Orchard, L. (2013). *Using geolocation*. Retrieved 27 July 2013, from <https://developer.mozilla.org/en-US/docs/Web/API/Geolocation>
- OSGeo. (2012). *Tile Map Service Specification*. Retrieved 7 March 2013, from http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification
- Osmani, A. (2013). *Learning JavaScript Design Patterns*. Retrieved 28 September 2013, from <http://addyosmani.com/resources/essentialjsdesignpatterns/book/#detailmvc>
- Papatsoutsos, D. (2001). *Information Systems Development Methodologies in the Age of Digital Economy*. University Campus, TYPA Buildings, 15771 Athens.
- Pascu, P. (2010). The Stages of Implementation of the SSADM System in the Government Institutions. *Journal of Applied Computer Science and Mathematics*, 8(4), 31–33.
- Perez, A. S. (2012). *OpenLayers Cookbook* (P. Dsouza, J. and Das, M. and Dalvi, P. and Salvi, Ed.). Birmingham, UK: Packt Publishing.
- Piccoli, G. (2012). *Information Systems for Managers: Text and Cases* (2nd Editio ed.). John Wiley and Sons, Inc.
- Pilgrim, M. (2010). *HTML5 - Up and Running* (1st Editio ed.; M. Loukides, Ed.). O'Reilly Media.
- Pollitt, C., Thiel, S. V., & Homburg, V. (2007, January). New Public Management in Europe. Retrieved from <http://www.palgraveconnect.com/doi/10.1057/9780230625365> doi: 10.1057/9780230625365
- Potmesil, M. (1997, September). Maps alive: viewing geospatial information on the WWW. *Computer Networks and ISDN Systems*, 29(8-13), 1327–1342. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0169755297000135> doi: 10.1016/S0169-7552(97)00013-5
- Prakash, S., Kumar, A., & Mishra, R. B. (2013). MVC Architecture Driven Design and Agile Implementation of a Web-Based Software System. *International Journal of Software Engineering & Applications (IJSEA)*, 4(6), 13–28.

- Quantum GIS. (2013). *QGIS Server*. Retrieved 15 August 2013, from https://qgis.readthedocs.org/en/latest/docs/user_manual/working_with_ogc/ogc_server_support.html
- R. Kwok. (2009). Personal technology: Phoning in data. *Nature*, 458(April), 959–961.
- Raento, M., Oulasvirta, A., & Eagle, N. (2009, February). Smartphones: An Emerging Tool for Social Scientists. *Sociological Methods & Research*, 37(3), 426–454. Retrieved from <http://smr.sagepub.com/cgi/doi/10.1177/0049124108330005> doi: 10.1177/0049124108330005
- Ravula, V. (2009). *Multi-tier Enterprise Application Architecture*. Retrieved 24 November 2013, from <http://www.codeproject.com/Articles/32653/Multi-tier-Enterprise-Application-Architecture>
- Répubblica de Moçambique. (2002). Lei do Desporto - n. 11/2002. In *Boletim da república* (chap. I Série -). Imprensa Nacional de Moçambique.
- S. Nusser and L. Miller and K. Clarke and M. Goodchild. (2003). Geospatial IT for mobile field data collection. *Communications of the ACM*, 46(1), 1–2. doi: 10.1145/602421.602446
- Satrom, B. (2011). *Integrating Geolocation into Web Applications*. Retrieved 29 July 2013, from <http://msdn.microsoft.com/en-us/magazine/hh580735.aspx>
- Seebregts, C. J., Zwarenstein, M., Mathews, C., Fairall, L., Flisher, A. J., Seebregts, C., ... Klepp, K.-I. (2009, November). Handheld computers for survey and trial data collection in resource-poor settings: development and evaluation of PDACT, a Palm Pilot interviewing system. *International journal of medical informatics*, 78(11), 721–31. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/19157967> doi: 10.1016/j.ijmedinf.2008.10.006
- Sencha Inc. (2011). *Sencha Blog*. Retrieved 16 November 2012, from <http://www.sencha.com/blog/>
- Sencha Inc. (2013a). *Ext.Component*. Retrieved 02 August 2013, from <http://docs.sencha.com/touch/2.0.2/#!/api/Ext.Component>
- Sencha Inc. (2013b). *Types of Proxy*. Retrieved 3 August 2013, from <http://docs.sencha.com/touch/2.3.1/#!/api/Ext.data.proxy.Proxy>
- Soares, D., & Amaral, L. (2011). Information systems interoperability in public administration: identifying the major acting forces through a Delphi study. *Journal of Theoretical and Applied Electronic Commerce Research*, 6(1), 61–94.
- Souders, S. (2011). *Storager case study: Bing, Google*. Retrieved 1 December 2013, from <http://www.stevesouders.com/blog/2011/03/28/storager-case-study-bing-google/>
- Sport England. (2005). *Spatial Planning for Sport and Active Recreation* (Tech. Rep.). England: Sport England. Retrieved from <http://www.sportengland.org/media/121927/document-8-spatial-planning-for-sport-guidance-on-se-aspirations-and-experience-5-.pdf>
- Sportbuchhandlung. (2000). *Schriftenreihe des Bundesinstitut für Sportwissenschaft*:

- Band 103* (Sportstättenentwicklungsplanung, Ed.). Köln: Hofmann.
- Stark, J. (2012). *Building Android Apps with HTML, CSS, and JavaScript* (2nd Edition ed.; Brian Jepson, Ed.). O'Reilly.
- Sumrada, R. (2002). The Shift from Stand-Alone to Web and Mobile GIS Solutions: Towards Distributed Application of GIS Technology. *GIM International*, 16, 40–43.
- Takada, M. (2012). *Single page apps in depth*. Retrieved 15 December 2013, from <http://singlepageappbook.com/goal.html>
- Tamelen, P. G. (2004, August). A comparison of obtaining field data using electronic and written methods. *Fisheries Research*, 69(1), 123–130. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0165783604000773> doi: 10.1016/j.fishres.2004.01.006
- The Center for Embedded Networked Sensing. (2010). *What's Invasive! Community Data Collection*. Retrieved 30 June 2013, from <http://www.whatsinvasive.org/index.html>
- Tolkach, Y. (2012). *Mobile Web App Frameworks Review: Sencha Touch vs jQuery Mobile*. Retrieved 3 February 2013, from <http://altabel.wordpress.com/2012/12/14/>
- Ubuntu. (2008). *WebKit*. Retrieved 28 September 2013, from wiki.ubuntu.com/WebKit
- University of Minnesota. (2013). *MapServer Documentation* (6.2.0 ed.). MapServer. Retrieved 4 April 2013, from <http://mapserver.org/documentation.html>
- Vidgen, R. (2002). Constructing a web information system development methodology. *Information Systems*, 12, 247–261.
- W3C. (2013a). *Geolocation API Specification*. Retrieved from http://dev.w3.org/geo/api/spec-source.html#geolocation_interface
- W3C. (2013b). *Web Storage*. Retrieved 25 October 2013, from <http://www.w3.org/TR/webstorage/#storage>
- Wagtendonk, A. J., & Jeu, R. A. M. D. (2007). Sensible Field Computing: Evaluating the Use of Mobile GIS Methods in Scientific Fieldwork. *Photogrammetric Engineering & Remote Sensing*, 73(6), 651–662.
- Wayner, P. (2013). *Review: Mobile Web development frameworks face off*. Retrieved 9 November 2013, from <http://www.javaworld.com/article/2078908/mobile-java/review--mobile-web-development-frameworks-face-off.html>
- Williams, G. (2012). *Learn HTML5 and JavaScript for Android*. Retrieved from <http://link.springer.com/10.1007/978-1-4302-4348-9> doi: 10.1007/978-1-4302-4348-9
- Willison, S. (2013). *A re-introduction to JavaScript (JS Tutorial)*. Retrieved 30 November 2013, from https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript
- Wroblewski, L. (2011). *Mobile First* (Mandy Brown, Ed.). Jeffrey Zeldman. Retrieved

from <http://abookapart.com>

- Xu, D., Yuan, Z., Yu, T., Xie, D., Zheng, F., & Zheng, G. (2012, October). The research of remote sensing image segmentation and release which are based on Tile Map Service. *2012 International Symposium on Geomatics for Integrated Water Resource Management*, 1–4. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6349564> doi: 10.1109/GIWRM.2012.6349564
- Yan, Y., Yu, J., Wu, J., & Ma, S. (2009). Design and Implementation of a Mobile GIS for Field Data Collection. *2009 WRI World Congress on Computer Science and Information Engineering*, 235–241. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5171169> doi: 10.1109/CSIE.2009.538
- Yu, P., de Courten, M., Pan, E., Galea, G., & Pryor, J. (2009, August). The development and evaluation of a PDA-based method for public health surveillance data collection in developing countries. *International journal of medical informatics*, 78(8), 532–42. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/19369114> doi: 10.1016/j.ijmedinf.2009.03.002
- Zaied, A. N., Aal, S. I., & Hassan, M. M. (2013). Rule-based Expert Systems for Selecting Information Systems Development Methodologies. *International Journal of Intelligent Systems and Applications*, 5(9), 19–26.
- Zakas, N. C. (2011). *localStorage Read Performance*. Retrieved 1 December 2013, from <http://calendar.perfplanet.com/2011/localstorage-read-performance/>
- Zakas, N. C. (2013, April). The evolution of web development for mobile devices. *Communications of the ACM*, 56(4), 42. Retrieved from <http://dl.acm.org/citation.cfm?doid=2436256.2436269> doi: 10.1145/2436256.2436269
- Zandbergen, P. a. (2009, June). Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning. *Transactions in GIS*, 13, 5–25. Retrieved from <http://doi.wiley.com/10.1111/j.1467-9671.2009.01152.x> doi: 10.1111/j.1467-9671.2009.01152.x
- Zhang, J., & Chung, J.-Y. (2003). Mockup-driven fast-prototyping methodology for web application development. *Software - Practice and Experience*, 33(1).
- Zhang, J., Liu, J., & Wang, B. (2010, May). Spatial Data Infrastructure for E-Government Based on the Geospatial Services. *2010 International Conference on E-Business and E-Government*, 646–650. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5590716> doi: 10.1109/ICEE.2010.169
- Zhang, Y., Li, D., & Zhu, Z. (2008, July). A Server Side Caching System for Efficient Web Map Services. *2008 International Conference on Embedded Software and Systems Symposia*, 32–37. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4627127> doi: 10.1109/ICESS.Symposia.2008.39

Appendix A

Tables

	App Cache	Web Storage	WebSQL	IndexedDB	File API
iOS 5.1	10 MB	5 MB	5 MB	5 MB	
Android 4	unlimited	5 MB	?	?	
Safari 5.2	unlimited	5 MB	5 MB	5 MB	
Chrome 18	5 MB	5 MB	unlimited	unlimited	unlimited
IE 10	50 MB	10 MB	500 MB	500 MB	
Opera 11	50 MB	5 MB	5 MB	5 MB	
Firefox 11	unlimited	10 MB	50 MB	50 MB	

Table A.1: HTML5 offline storage capacity (adapted from Hochdörfer (2013)).

Class	Subclass	Geometry type	Source
Administrative divisions	Province	Polygon	CENACARTA
	District	Polygon	CENACARTA
	Town	Point	CENACARTA
	Province Capital	Point	CENACARTA
	District Capital	Point	CENACARTA
	Town Capital	Point	CENACARTA
Hydrography	Lake	Polygon	CENACARTA
	River	Polygon, Line	CENACARTA
Communication	Road	Line	CENACARTA
	Railway	Line	CENACARTA
Green spaces	Park	Polygon	CENACARTA
Point of interest		Point	INGC

Table A.2: Classes of Shapefiles Selected for QGIS Project.

Table A.4: m-SportGIS Tree Structure (Fields Form Navigation Panel).

1° level	2° level	3° level	4° level
Técnico			
Função			
Responsável			
Fonte dos dados			

Telefone/Fax do técnico			
Telemóvel do técnico			
Email do técnico			
Nome da instalação			
Código da instalação			
Localização			
Província	Seleccionar Província		
Distrito	Seleccionar Distrito		
Posto	Seleccionar Posto		
Telefone/Fax			
Email			
Propriedade	Pública	Administração central	
		Divisão administrativa	
	Privada	Clubes/Associações	Nº associados
		Outro	
Proprietário			
Entidade gestora			
Valor do investimento (MT)			
Área total (m²)			
Ano de inauguração			
Ano de reabilitação			
Modalidade principal	Andebol		
	Artes Marciais		
	Atletismo		
	Badminton		
	Basquetebol		
	Boxe		
	Canoagem		
	Ciclismo		
	Futebol		
	Ginástica		
	Golfe		
	Hipismo		
	Natação		
	Ténis		
	Ténis de mesa		
	Triatlo		
	Vela		
	Voleibol		
	Voleibol de praia		
Nº de praticantes			

Outras modalidades	Andebol	Nº de praticantes
	Artes Marciais	Nº de praticantes
	Atletismo	Nº de praticantes
	Badminton	Nº de praticantes
	Basquetebol	Nº de praticantes
	Boxe	Nº de praticantes
	Canoagem	Nº de praticantes
	Ciclismo	Nº de praticantes
	Futebol	Nº de praticantes
	Ginástica	Nº de praticantes
	Golfe	Nº de praticantes
	Hipismo	Nº de praticantes
	Natação	Nº de praticantes
	Ténis	Nº de praticantes
	Ténis de mesa	Nº de praticantes
	Triatlo	Nº de praticantes
	Vela	Nº de praticantes
	Voleibol	Nº de praticantes
	Voleibol de praia	Nº de praticantes
Especificações	Campo de andebol	Categoria
		Comprimento (m)
		Largura (m)
	Campo de basquetebol	Categoria
		Comprimento (m)
		Largura (m)
	Campo de futebol	Categoria
		Comprimento (m)
		Largura (m)
	Campo de golfe	
	Campo de ténis	Categoria
		Largura (m)
	Campo de voleibol	Categoria
		Comprimento (m)
		Largura (m)
	Circuito de ciclismo	
	Ginásio	
	Hipódromo	
	Pavilhão Desportivo	
	Piscina	Não olímpica
		Semi-olímpica
		Olímpica

Classificação	Polivalente	
	Competição	
	Recreativa	
	Formativa	
Cobertura	Ar livre	
	Simplesmente coberta	
	Coberta + Encerrada	Altura (m)
	Combinada/Mista	Altura (m)
Acessibilidade limitada		
Equipamento fixo		
Armazém de material		
WC		
Anexo de atletas	Balneários	
	Sanitários	
	Vestiários	
Anexo de técnicos	Balneários	
	Sanitários	
	Vestiários	
Musculação/Massagem		
Restauração	Bares/Cafés	
	Lojas	
	Restaurantes	
Pavimento desportivo	Betuminoso	
	Cimento	
	Sintético	
	Cinza	
	Madeira	
	Relva	
	Relva com irrigação	
	Solo	
	Outro	
Vedação exterior	Parcial	
	Completa	
Iluminação artificial		
Ar/Ambiente	Climatização	
	Ventilação mecânica	
	Ventilação natural	
Dispositivos de segurança	Controlo de entrada	
	Video-vigilância	
Estado de conservação	Fora de Uso	
	Muito bom	

	Mau
	Regular
Bancada	Simples
	Com camarote
	Com tribuna
Instalação para imprensa	
Acessibilidade	Boa
	Má
	Regular
Utilizadores	
Capacidade	

Zoom level	Layers
0	adm_prov, cap_provincia
1	adm_prov, cap_provincia, reserv_areas, hid_areas
2	adm_prov, adm_dis, cap_provincia, cap_distrito, reserv_areas, hid_areas
3	adm_prov, adm_dis, cap_provincia, cap_distrito, reserv_areas, hid_areas, rails, roads_pri
4	adm_dis, cap_distrito, adm_postos, cap_posto_outros, reserv_areas, hid_areas, rails, roads_pri, roads_sec
5	adm_dis, cap_distrito, adm_postos, cap_posto_outros, reserv_areas, hid_areas, rails, roads_pri, roads_sec
6	adm_dis, cap_distrito, adm_postos, cap_posto_outros, reserv_areas, hid_areas, rails, roads_pri, roads_sec, moz_POI
7	adm_dis, adm_postos, cap_posto_outros, reserv_areas, hid_areas, hid_lines, rails, roads_pri, roads_sec, moz_POI
8	adm_postos, cap_posto_outros, reserv_areas, hid_areas, hid_lines_pri, rails, roads_pri, roads_sec, moz_POI

Table A.3: Tile content (layers) according to zoom level.

Appendix B

Program Listings

Listing B.1 m-SportGIS Loader Function (Ext.Application).

```
// ux: multiselectfield extension of selectfieldselect (API ST2)
Ext.Loader.setPath({
    'Ux.field': 'app/view/ux' });
// loading of SportApp components
SportApp = Ext.application({
    name: 'SportApp',
    // master namespace - global app JS object
    useLoadMask: true,
    requires: [
        'Ux.field.Multiselect',
        'Ux.field.Time',
        'Ux.field.TimePicker'
    ],
    controllers: [ 'MapController', 'SportController' ],
    views: [ 'MainView', 'DataCollectingView', 'SportListView', 'StartView' ],
    models: [ 'ImageModel', 'SportModel', 'UploadSportModel' ],
    stores: [ 'ImageStore', 'SportStore', 'DistritosStore', 'PostosStore', 'ProvinciasStore' ],
    launch: function() {
        Ext.create(
            "SportApp.view.MainView");
    }
});
```

Listing B.2 m-SportGIS StartView

```
Ext.define(
    'SportApp.view.StartView',
    {
        extend : "Ext.form.Panel",
        alias : "widget.startview",
        config : {
            scrollable : false
        },
        initialize : function() {
            this.callParent(arguments);
            // console.log('initialize startview');
            this.newButton = Ext.create(
                'Ext.Button', {
                    text : 'Criar Novo Registo',
                    ui : 'action',
                    width : '300',
                    handler : this.onNewNote,
                    scope : this
                });
            this.viewDatasetsButton = Ext.create(
                'Ext.Button', {
                    text : 'Ver Registos',
                    ui : 'action',
                    width : '300',
                    handler : this.onShowList,
                    scope : this
                });
            this.topToolbar = Ext.create(
                'Ext.Toolbar', {
                    docked : 'top',
                    items : [
                        { xtype : 'spacer' },
                        { xtype : "container",
                            docked : 'top',
                            { xtype : 'spacer' }
                        }
                    ]
                });
            this.root = Ext.create(
                'Ext.Panel', {
                    margin : "150 0 0 200",
                    layout : 'hbox',
                    items : [
                        this.viewDatasetsButton,
                        { xtype : 'panel',
                            width : '50%',
                            items : [
                                {html : "<CENTER><img src='blank.gif' style='width:10%'>"},
                                {html : "<CENTER><img src='logo_cor01_transparent.png' style='width:105%'>"},
                                {html : "<CENTER><img src='blank.gif' style='width:10%'>"},
                                {html : "<CENTER><font size='1' color='red'>Prot\xF3tipo powered by LabNT</font></html>"}
                            ]
                        },
                        this.newButton
                    ]
                });
            this.add(
                [this.topToolbar,
                 this.root]
            );
        },
        onNewNote : function() {
            this.fireEvent("newfacility", this);
            this.fireEvent("showdatacollectingview", this);
            // start Cordova Geolocation capability
            this.fireEvent("startwatchingposition", this);
        },
        onShowList : function() {
            this.fireEvent("showlistview", this);
        },
    });
```

Listing B.3 Tree List of Mozambique Administrative Divisions.

```
// Nested data – feeding of province–district–town
this.dataForm2 = Ext.create(
    'Ext.form.FieldSet', {
        title: 'ID Empresa',
        zIndex: 0,
        width: Ext.Viewport.getSize().width / 2 + 70,
        scrollable: {
            direction: 'vertical',
            directionLock: true
        },
        defaults: {
            labelWidth: '44%',
        },
        items: [{
            // provinces :
            xtype: 'selectfield',
            name: 'idProvince',
            label: '* Província',
            itemId: 'idProvíncia',
            options: [{
                text: '',
                value: null
            }, {
                text: 'Cabo Delgado',
                value: '1'
            }, {
                text: 'Gaza',
                value: '2'
            }, {
                text: 'Ilha Licom',
                value: '3'
            }, {
                text: 'Ilha Risunodo',
                value: '4'
            }, {
                text: 'Inhambane',
                value: '5'
            }, {
                text: 'Lago Niassa',
                value: '6'
            }, {
                text: 'Manica',
                value: '7'
            }, {
                text: 'Maputo',
                value: '8'
            }, {
                text: 'Nampula',
                value: '9'
            }, {
                text: 'Niassa',
                value: '10'
            }, {
                text: 'Sofala',
                value: '11'
            }, {
                text: 'Tete',
                value: '12'
            }, {
                text: 'Zambézia',
                value: '13'
            }
        ]], //end of options
        listeners: {
            change: function(provList, newValue, oldValue) {
                var targetSelect = Ext.ComponentQuery.query(
                    'datacollectingview selectfield[id=
                    selectfieldDistrict]')[0];
                targetSelect.disable();
                var distritosStore = Ext.getStore('DistritosStore');
                distritosStore.clearFilter(true);
                distritosStore.filter(Ext.create('Ext.util.Filter', {
                    filterFn: function(item) {
                        return (item.get('idProvince') == newValue);
                    }
                }));
                targetSelect.enable(); // unlocks dropdown
            }
        }
    } // end of listeners
),
// districts : {
xtype: 'selectfield',
itemId: 'selectfieldDistrict',
name: 'idDistrict',
label: '* Distrito',
disabled: true,
store: 'DistritosStore',
displayField: 'district',
valueField: 'idDistrict',
listeners: {
    change: function(distList, newValue, oldValue) {
        var target2Select = Ext.ComponentQuery.query(
            'datacollectingview selectfield[id=selectfieldTown
            ]')[0];
    }
}
```

```

        target2Select.disable();
        var storePosto = Ext.getStore('PostosStore');
        storePosto.clearFilter(true);
        storePosto.filter(Ext.create('Ext.util.Filter', {
            filterFn: function(item) {
                return (item.get('idDistrict') == newValue);
            }
        }));
        target2Select.enable(); // unlocks dropdown
    } // end of listeners
},
// towns : {
    xtype: 'selectfield',
    name: 'idTown',
    label: '* Posto',
    itemId: 'selectfieldTown',
    store: 'PostosStore',
    displayField: 'town',
    valueField: 'idTown'
}] // end of items
});

```

Listing B.4 m-SportGIS MultiList class.

```

Ext.define(
    'SportApp.view.MultiList', {
        extend: 'Ext.Panel',
        xtype: 'multilist',
        config: {
            single: false,
                // allows multi selection
            useComponents: true,
            defaultType: 'multilistoverlayitem',
            width: '100%',
            height: '100%',
            listeners: {
                erased: function() {...}
                painted: function() {...}
            }
        }
    });

```

Listing B.5 m-SportGIS MultiListOverlay Class.

```

// define the overlay for the nested list
Ext.define(
    'SportApp.view.MultiListOverlay', {
        xtype: 'multilistoverlay',
        extend: 'Ext.Panel',
        config: {
            ref: null,
            hideOnMaskTap: true,
            modal: true
        },
        constructor: function(config) {
            this.callParent(arguments);
            this.initConfig(config);
        }
    });

```

Listing B.6 m-SportGIS MultiListOverlayItem Class.

```
Ext.define(
    'MultiListOverlayItem', {
        extend: 'Ext.dataview.component.DataItem',
        requires: ['Ext.field.Checkbox'],
        xtype: 'multilistoverlayitem',
        config: {
            labelCheckbox: true,
            dataMap: {
                getLabelCheckbox: {
                    setLabel: 'label'
                }
            },
            applyLabelCheckbox: function(config) {
                return Ext.factory(config, Ext.field.Checkbox, this.getLabelCheckbox());
            },
            updateLabelCheckbox: function(newLabelCheckbox, oldLabelCheckbox) {
                if (oldLabelCheckbox) {
                    this.remove(oldLabelCheckbox);
                }
                if (newLabelCheckbox) {
                    // sets if checked or not
                    var formRecord = SportApp.view.DataCollectingView.getRecord();
                    var contentRecord = this.getRecord();
                    if (formRecord.get(contentRecord.get("name")) != null && formRecord.get(
                        contentRecord.get("name"))) {
                        newLabelCheckbox.check();
                    }
                    newLabelCheckbox.setLabelWidth("70%");
                    newLabelCheckbox.on('check', this.onLabelCheckboxTab, this, { check:
                        true });
                    newLabelCheckbox.on('uncheck', this.onLabelCheckboxTab, this, { check:
                        false });
                    this.add(newLabelCheckbox);
                }
            },
            onLabelCheckboxTab: function(currentCheckbox, e, options) {
                var contentRecord = this.getRecord();
                // updates checkbox state
                SportApp.view.DataCollectingView.formPanelOverlay.getRecord().set(contentRecord
                    .get("name"), options.check);
                if (!e) {
                    var dataview = this.getParent().getParent();
                    // access dataview
                    if (dataview.getSingle() && options.check) {
                        var checkboxContainers = dataview.getActiveItem().getInnerItems()
                            ;
                        for (var i = 0; i < checkboxContainers.length; i++) {
                            if (checkboxContainers[i].getRecord().get("name") !=
                                contentRecord.get("name")) {
                                checkboxContainers[i].getLabelCheckbox().uncheck();
                                // updates checkbox state
                                SportApp.view.DataCollectingView.formPanelOverlay
                                    .getRecord().set(checkboxContainers[i].
                                        getRecord().get("name"), false);
                                // deletes selection of selectfield
                                SportApp.view.DataCollectingView.formPanelOverlay
                                    .getRecord().set(checkboxContainers[i].
                                        getRecord().get("selectfieldname"), null);
                            }
                        }
                    }
                }
                return;
            }
        },
        SportApp.view.DataCollectingView.formPanelOverlay.hide();
        // updates record
        var record = SportApp.view.DataCollectingView.formPanelOverlay.getRecord();
        SportApp.view.DataCollectingView.formPanelOverlay.updateRecord(record);
        SportApp.view.DataCollectingView.formPanelOverlay.setRecord(record);
        SportApp.view.DataCollectingView.formPanelOverlay.removeAll();
        // creates content for the overlay
        var fieldSet = Ext.create(
            'Ext.form.FieldSet', {
                width: "100%", defaults: {
                    labelWidth: "60%", width: "100%" } });
        // if no selectfield --> creates textfield etc...
        if (contentRecord.get("selectfieldname") == null) {
            for (var i = 1; i <= 3; i++) {
                if (contentRecord.get("xtype" + i) != null) {
                    fieldSet.add({
                        xtype: contentRecord.get("xtype" + i),
                        name: contentRecord.get("field" + i),
                        label: contentRecord.get("label" + i),
                        listeners: {
                            change: function(textfield, newValue,
                                oldValue, eOpts) {
                                    currentCheckbox.check();
                                    for (var j = 0; j < fieldSet.getItems
                                        ().items.length; j++) {
                                        var value = fieldSet.getItems
                                            ().items[j].getValue();
                                        if ((value == null || value ==
                                            "")) {
                                            currentCheckbox.uncheck
                                                ();
                                        }
                                    }
                                }
                        }
                    });
                }
            }
        }
        else {
            fieldSet.setDefaults({
```

```

        labelWidth: "70%",
        width: "100%"
    });
    // creates radiofield and defines which field is select
    var selection = SportApp.view.DataCollectingView.formPanelOverlay.getRecord().get(
        contentRecord.get("selectfieldname"));
    for (var i = 1; i <= 3; i++) {
    if (contentRecord.get("field" + i) != null) {
        fieldSet.add({
            xtype: "panel",
            layout: "hbox",
            defaults: {
                labelWidth: "85%" },
            items: [{
                xtype: 'radiofield',
                name: contentRecord.get("selectfieldname"),
                value: i,
                label: contentRecord.get("field" + i),
                checked: i == selection,
                width: "85%",
                id: "radio" + i,
                listeners: {
                    uncheck: function(radioField, e, eOpts) {
                        // hide button
                        radioField.getParent().getItems().items[1].setHidden(true); },
                    check: function(radioField, e, eOpts) {
                        // SportApp.view.DataCollectingView.formPanelOverlay.getRecord().set(
                            contentRecord.get("selectfieldname"), i);
                        currentCheckbox.check();
                        // defines 4th overlay level
                        if (contentRecord.get("field" + this.getValue() + "subfield1") != null
                            && SportApp.view.DataCollectingView.formPanelOverlay2 == null) {
                            SportApp.view.DataCollectingView.formPanelOverlay2 = Ext.create(
                                'Ext.form.Panel', {
                                    height: 180,
                                    width: 300, });
                            // defines overlay content of 4th level
                            var fieldSet2 = Ext.create(
                                'Ext.form.FieldSet', {
                                    width: "100%",
                                    defaults: {
                                        labelWidth: "50%",
                                        width: "100%" } },
                                for (var j = 1; j <= 3; j++) {
                                    if (contentRecord.get("field" + this.getValue() + "subtype" + j) !=
                                        null) {
                                        fieldSet2.add({ xtype: contentRecord.get("field" + this.getValue()
                                            + "subtype" + j),
                                            name: contentRecord.get("field" + this.getValue() + "subfield" +
                                                j),
                                            label: contentRecord.get("field" + this.getValue() + "sublabel" +
                                                j) });
                                    };
                                };
                            SportApp.view.DataCollectingView.formPanelOverlay2.add([
                                fieldSet2, {
                                    xtype: "panel",
                                    layout: "hbox",
                                    defaults: {
                                        flex: 1 },
                                    items: [{
                                        xtype: "button",
                                        text: 'Ok',
                                        ui: 'action',
                                        handler: function() {
                                            var record = SportApp.view.DataCollectingView.formPanelOverlay.getRecord();
                                            SportApp.view.DataCollectingView.formPanelOverlay2.updateRecord(record);
                                            SportApp.view.DataCollectingView.formPanelOverlay.setRecord(record);
                                            SportApp.view.DataCollectingView.formPanelOverlay2.hide();
                                            SportApp.view.DataCollectingView.formPanelOverlay2.destroy();
                                            SportApp.view.DataCollectingView.formPanelOverlay2 = null;
                                        }},
                                    { xtype: "button",
                                        iconMask: true,
                                        iconCls: 'delete',
                                        ui: 'action',
                                        // deletes handler function
                                        handler: function() {
                                            SportApp.view.DataCollectingView.formPanelOverlay2.reset();

```

```

SportApp.view.DataCollectingView.formPanelOverlay.reset();
// save record
var record = SportApp.view.DataCollectingView.formPanelOverlay.getRecord();
SportApp.view.DataCollectingView.formPanelOverlay2.updateRecord(record);
SportApp.view.DataCollectingView.formPanelOverlay.setRecord(record);
SportApp.view.DataCollectingView.formPanelOverlay2.hide();
SportApp.view.DataCollectingView.formPanelOverlay2.destroy();
SportApp.view.DataCollectingView.formPanelOverlay2 = null;
} }}
});
// loads record
var record = SportApp.view.DataCollectingView.formPanelOverlay.getRecord();
SportApp.view.DataCollectingView.formPanelOverlay.updateRecord(record);
SportApp.view.DataCollectingView.formPanelOverlay2.setRecord(record);
// shows button
var button = radioField.getParent().getItems().items[1];
button.setHidden(false);
// show
SportApp.view.DataCollectingView.formPanelOverlay2.showBy(button, "cl-cr");
} } } },
// end of radio field
{
xtype: "button",
iconCls: 'arrow_right',
iconMask: true,
ui: 'plain',
hidden: true,
handler: function() {
var radioField = this.getParent().getItems().items[0];
radioField.fireEvent("check", radioField, null, null);
} } } });
};
};
};
if (fieldSet.getItems().length > 0) {
SportApp.view.DataCollectingView.formPanelOverlay.add(
[
fieldSet, {
xtype: "panel",
layout: "hbox",
defaults: {
flex: 1 },
items: [ {
xtype: "button",
text: 'Ok',
ui: 'action',
handler: function() {
SportApp.view.DataCollectingView.formPanelOverlay.hide();
// update record
var record = SportApp.view.DataCollectingView.formPanelOverlay.getRecord();
SportApp.view.DataCollectingView.formPanelOverlay.updateRecord(record);
SportApp.view.DataCollectingView.formPanelOverlay.setRecord(record);
} },
{
xtype: "button",
iconMask: true,
iconCls: 'delete',
ui: 'action',
handler: function() {
// deletes handler function
SportApp.view.DataCollectingView.formPanelOverlay.reset();
SportApp.view.DataCollectingView.formPanelOverlay.hide();
// updates record
var record = SportApp.view.DataCollectingView.formPanelOverlay.getRecord();
SportApp.view.DataCollectingView.formPanelOverlay.updateRecord(record);
SportApp.view.DataCollectingView.formPanelOverlay.setRecord(record);
},
} ]
} ] });
}
// shows overlay if content exists
if (SportApp.view.DataCollectingView.formPanelOverlay.getItems().length > 0) {
// updates record
var record = SportApp.view.DataCollectingView.formPanelOverlay.getRecord();
SportApp.view.DataCollectingView.formPanelOverlay.setRecord(record);
if (options.check) {
currentCheckbox.uncheck();
}
}
}

```

```

        else {
            currentCheckbox.check();
        }
        SportApp.view.DataCollectingView.formPanelOverlay.showBy(this, "
            cl-cr");
    };
} });

```

Listing B.7 Validations (rules) in Model.

```

Ext.define(
'SportApp.model.UploadFacilityModel', {
    extend: 'Ext.data.Model',
    config: {
        idProperty: 'imageID',
        fields: [
            { name: 'id', type: 'int' },
            { name: 'date', type: 'date', dateFormat: 'c' },
            { name: 'technician', type: 'string' },
            { name: 'role', type: 'string' },
            { name: 'responsable', type: 'string' },
            { name: 'source', type: 'string' },
            { name: 'telephone', type: 'string' },
            { name: 'mobile', type: 'string' },
            { name: 'email', type: 'string' },
            { name: 'installationName', type: 'string' },
            { name: 'installationCode', type: 'int' },
            { name: 'address', type: 'string' },
            { name: 'idProvince', type: 'int' },
            { name: 'idDistrict', type: 'int' },
            { name: 'idTown', type: 'int' },
            { name: 'telephone2', type: 'string' },
            { name: 'email2', type: 'string' },
            { name: 'propertyPublic', type: 'boolean' },
            { name: 'propertyPublicCategory', type: 'string' },
            { name: 'propertyPrivate', type: 'boolean' },
            { name: 'numberOfClubs', type: 'int' },
            { name: 'propertyPrivateCategory', type: 'string' },
            { name: 'owner', type: 'string' },
            { name: 'manager', type: 'string' },
            { name: 'investment', type: 'string' },
            { name: 'areaTotal', type: 'string' },
            { name: 'inauguration', type: 'string' },
            { name: 'rehabilitation', type: 'string' },
            { name: 'activityMain', type: 'int' },
            { name: 'handballMod', type: 'boolean' },
            { name: 'artsMod', type: 'boolean' },
            { name: 'athleteMod', type: 'boolean' },
            { name: 'badmintonMod', type: 'boolean' },
            { name: 'basketballMod', type: 'boolean' },
            { name: 'boxingMod', type: 'boolean' },
            { name: 'canoeingMod', type: 'boolean' },
            { name: 'cyclingMod', type: 'boolean' },
            { name: 'footballMod', type: 'boolean' },
            { name: 'golfMod', type: 'boolean' },
            { name: 'poolMod', type: 'boolean' },
            { name: 'tennisMod', type: 'boolean' },
            { name: 'tabletennisMod', type: 'boolean' },
            { name: 'triathlonMod', type: 'boolean' },
            { name: 'sailingMod', type: 'boolean' },
            { name: 'volleyballMod', type: 'boolean' },
            { name: 'beachvolleyballMod', type: 'boolean' },
            { name: 'cycling', type: 'boolean' },
            { name: 'handballField', type: 'boolean' },
            { name: 'handballFieldCategory', type: 'string' },
            { name: 'handballFieldWidth', type: 'int' },
            { name: 'handballFieldLength', type: 'int' },
            { name: 'handballField', type: 'boolean' },
            { name: 'basketballFieldCategory', type: 'string' },
            { name: 'basketballFieldWidth', type: 'int' },
            { name: 'basketballFieldLength', type: 'int' },
            { name: 'footballField', type: 'boolean' },
            { name: 'footballFieldCategory', type: 'string' },
            { name: 'footballFieldWidth', type: 'int' },
            { name: 'footballFieldLength', type: 'int' },
            { name: 'golf', type: 'boolean' },
            { name: 'tennisField', type: 'boolean' },
            { name: 'tennisFieldCategory', type: 'string' },
            { name: 'tennisFieldWidth', type: 'int' },
            { name: 'tennisFieldLength', type: 'int' },
            { name: 'volleyballField', type: 'boolean' },
            { name: 'volleyballFieldCategory', type: 'string' },
            { name: 'volleyballFieldWidth', type: 'int' },
            { name: 'volleyballFieldLength', type: 'int' },
            { name: 'fitness', type: 'boolean' },
            { name: 'gym', type: 'boolean' },
            { name: 'hall', type: 'boolean' },
            { name: 'equestrian', type: 'boolean' },

```

```

    { name: 'pool', type: 'boolean' },
    { name: 'poolCategory', type: 'string' },
    { name: 'athleteField', type: 'boolean' },
    { name: 'athleteFieldCategory', type: 'string' },
    { name: 'athleteFieldWidth', type: 'int' },
    { name: 'athleteFieldLength', type: 'int' },
    { name: 'multifield', type: 'boolean' },
    { name: 'classification', type: 'int' },
    { name: 'openRoof', type: 'boolean' },
    { name: 'cover', type: 'boolean' },
    { name: 'coverClosed', type: 'boolean' },
    { name: 'coverClosedHeight', type: 'int' },
    { name: 'roofMix', type: 'boolean' },
    { name: 'roofMixHeight', type: 'int' },
    { name: 'accessLimited', type: 'boolean' },
    { name: 'equipment', type: 'boolean' },
    { name: 'storageMaterial', type: 'boolean' },
    { name: 'wc', type: 'boolean' },
    { name: 'outbuildingAthlete', type: 'int' },
    { name: 'outbuildingTechnician', type: 'int' },
    { name: 'musclesRoom', type: 'boolean' },
    { name: 'aid', type: 'boolean' },
    { name: 'bars', type: 'int' },
    { name: 'ground', type: 'int' },
    { name: 'fenceParcial', type: 'boolean' },
    { name: 'fenceParcialType', type: 'string' },
    { name: 'fenceComplete', type: 'boolean' },
    { name: 'fenceCompleteType', type: 'string' },
    { name: 'light', type: 'boolean' },
    { name: 'air', type: 'int' },
    { name: 'entranceControl', type: 'boolean' },
    { name: 'videoControl', type: 'boolean' },
    { name: 'condition', type: 'int' },
    { name: 'bencheSimple', type: 'boolean' },
    { name: 'bencheCabin', type: 'boolean' },
    { name: 'bencheTribune', type: 'boolean' },
    { name: 'press', type: 'boolean' },
    { name: 'accessibility', type: 'int' },
    { name: 'userClub', type: 'boolean' },
    { name: 'userSchool', type: 'boolean' },
    { name: 'userPublic', type: 'boolean' },
    { name: 'userMunicipality', type: 'boolean' },
    { name: 'userOther', type: 'boolean' },
    { name: 'capacity', type: 'string' },
    { name: 'coordinates', type: 'string' },
    { name: 'accuracy', type: 'int', defaultValue: '' },
    { name: 'imageID1', type: 'int' },
    { name: 'imageID2', type: 'int' },
    { name: 'imageID3', type: 'int' },
    { name: 'imageID4', type: 'int' },
    { name: 'imageID5', type: 'int' },
    { name: 'image1', type: 'string' },
    { name: 'image2', type: 'string' },
    { name: 'image3', type: 'string' },
    { name: 'image4', type: 'string' },
    { name: 'image5', type: 'string' }
  ],

  validations: [
    // presence : mandatory field
    // format : data type // \d --> digit: [0-9]
    // \d* --> number or empty
    // \D --> non-digit : [^0-9]
    {type: 'presence', field: 'id' },
    {type: 'format', name: 'technician', matcher: /\D/, message: 'Nome do técnico inválido!'},
    {type: 'format', name: 'installationName', matcher: /\D/, message: 'Nome da instalação inválido!'},
    //{type: 'format', name: 'installationCode', matcher: /\d*/, message: 'Código inválido!'},
    {type: 'presence', name: 'address', message: 'Verifique a localização!'},
    {type: 'presence', name: 'idProvince', message: 'Província da instalação em falta!'},
    {type: 'presence', name: 'idDistrict', message: 'Distrito da instalação em falta!'},
    {type: 'presence', name: 'idTown', message: 'Posto da instalação em falta!'},
    {type: 'format', name: 'owner', matcher: /\D/, message: 'Nome do proprietário inválido!'},
    {type: 'presence', name: 'activityMain', message: 'Modalidade principal em falta!'},
    {type: 'presence', name: 'coordinates', message: 'Posição desconhecida!'},
    {type: 'format', name: 'mobile', matcher: /\d*/, message: "Telefone inválido"}],

```

```

{type: 'format', name: 'telephone', matcher: /\d*/, message: "Telefone inválido"},
{type: 'format', name: 'telephone2', matcher: /\d*/, message: "Telefone inválido"},
{type: 'format', name: 'email', matcher: /^(^([a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z])$)
?/, message: "Email inválido"},
{type: 'format', name: 'email2', matcher: /^(^([a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z])$)
?/, message: "Email inválido"},
{type: 'format', name: 'numberOfClubs', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'investment', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'inauguration', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'rehabilitation', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'coverClosedHeight', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'roofMixHeight', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'areaTotal', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'athleteFieldWidth', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'athleteFieldLength', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'footballFieldWidth', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'footballFieldLength', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'basketballFieldWidth', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'basketballFieldLength', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'tennisFieldWidth', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'tennisFieldLength', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'capacity', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'practitioners', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'practitionersHandball', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersArts', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'practitionersAthlete', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersBadminton', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersBasketball', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersBoxing', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersCanoeing', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersCycling', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersFootball', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersGolf', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'practitionersEquestrian', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersPool', matcher: /\d*/, message: "Valor inválido"},
{type: 'format', name: 'practitionersTennis', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersTabletennis', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersTriathlon', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersSailing', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersVolleyball', matcher: /\d*/, message: "Valor inválido"},
},
{type: 'format', name: 'practitionersBeachvolleyball', matcher: /\d*/, message: "Valor inválido"},
},
}
});

```

Listing B.8 Facility Controller.

```
Ext.define('SportApp.controller.facilityController', {
    extend: 'Ext.app.Controller',
    init : function() {
        facilityController = this;
        me = this;
    },
    config : {
        refs: {
            dataCollectingView: "datacollectingview",
            mainView: "mainview",
            startView: "startview",
            facilityListView: "facilitylistview",
        },
        control: {
            dataCollectingView: {
                savefacility: "savefacility",
                showstartview: "showstartview",
                showlistview: "showlistview",
                submit: "submit",
                showcamera: "showcamera",
                deletepicture: "deletepicture",
                deletepicturephysically: "deletepicturephysically",
                deletefacility: "deletefacility",
            },
            mainView: {
                loadimages: "loadimages"
            },
            startView: {
                newfacility: "newfacility",
                showdatacollectingview: "showdatacollectingview",
                showlistview: "showlistview",
            },
            facilityListView: {
                showstartview: "showstartview",
                submitall: "submitall",
                editfacility: "editfacility",
                showdatacollectingview: "showdatacollectingview",
                submit: "submit",
                //updatePic : "updatePic",
            }
        },
        // loads stores - default field null in dropdown field option:
        nullDropdown : function() {
            var divisaoStore = Ext.getStore('DivisaoStore');
            var grupoStore = Ext.getStore('GrupoStore');
            var classeStore = Ext.getStore('ClasseStore');
            var subclasseStore = Ext.getStore('SubclasseStore');
            divisaoStore.clearFilter(true);
            divisaoStore.filter(Ext.create('Ext.util.Filter', {
                filterFn: function(item) {
                    return false;
                }
            }));
            grupoStore.clearFilter(true);
            grupoStore.filter(Ext.create('Ext.util.Filter', {
                filterFn: function(item) {
                    return false;
                }
            }));
            classeStore.clearFilter(true);
            classeStore.filter(Ext.create('Ext.util.Filter', {
                filterFn: function(item) {
                    return false;
                }
            }));
            subclasseStore.clearFilter(true);
            subclasseStore.filter(Ext.create('Ext.util.Filter', {
                filterFn: function(item) {
                    return false;
                }
            }));
        },
        // called on click new record (start view)
        newfacility : function(options) {

            /*
            creates a nomedia extension - ensures device does not run over all tiles images :
            window.resolveLocalFileSystemURI("file:///mnt/extSdCard/tilecache/.nomedia", fileSuccess ,
                fileError);
            function fileSuccess(){
                return
            }
            function fileError(){
            window.requestFileSystem(LocalFileSystem.PERSISTENT, 0, gotFS, fail);
            function gotFS(fileSystem){
            console.log('---> nomedia nao existe');
            fileSystem.root.getFile("/mnt/extSdCard/tilecache/.nomedia", {create: true, exclusive: false
                }, gotFileEntry, fail);
            };
            function gotFileEntry() {
            console.log('--->.nomedia criado');
            }
            */
        }
    }
});
```

```

function fail(evt) {
console.log('nomedia nao criado');
}
};
*/

this.getDataCollectingView().switch2NewfacilityMode();
SportApp.app.getController("MapController").switch2newmode();
var now = new Date();
var facilityId = now.getTime();
// loads technician to keep its data in new records
var technician = Ext.getStore('facilityStore').getTechnician();
var facility = Ext.create("SportApp.model.facilityModel", {
id: facilityId,
date: now,
technician: technician
});
// set filter for image store
Ext.getStore('ImageStore').clearFilter();
Ext.getStore('ImageStore').filter('facilityID', facilityId);
this.getDataCollectingView().setRecord(facility);
},

// called on click edit record
editfacility : function(placeholder, record) {
this.getDataCollectingView().switch2EditMode();
var controller = SportApp.app.getController("MapController");
controller.switch2editmode();
//console.log(record.get("id"));
// set filter for image store
Ext.getStore('ImageStore').clearFilter();
Ext.getStore('ImageStore').filter('facilityID', record.get("id")); // facility.data.
this.getDataCollectingView().setRecord(record);
},

// called on save data (data collecting view)
savefacility : function(options) {
var currentfacility = this.getDataCollectingView().getRecord();
this.getDataCollectingView().updateRecord(currentfacility);
currentfacility.set('submitted', false);
var errors = currentfacility.validate(),
message = '';
if (!errors.isValid()) {
errors.each(function(err) {
message += err.getMessage() + '<br/>';
}); // each()
new Ext.MessageBox().alert('Registro não pode ser guardado', message);
} else {
// if valid, safe
if (null == Ext.getStore('facilityStore').findRecord('id', currentfacility.get("id"))) {
Ext.getStore('facilityStore').add(currentfacility);
} else {
currentfacility.setDirty();
}
Ext.getStore('facilityStore').sync();
Ext.getStore('facilityStore').sort([{
property: 'date',
direction: 'DESC'
}]);
// clear view
this.getDataCollectingView().clearView();
// update map
SportApp.app.getController("MapController").showallfacilities();
// return to last active view
if (mode == 1) {
this.showstartview('right');
} else {
this.showlistview();
}
// save images
this.saveimages();
}
},

loadPic : function(imageURI) {
//var imageURI = "file:///mnt/sdcard/sport_pictures/1365782308912/1365782373988.jpg";
var options = new FileUploadOptions();
options.fileKey = "file";
options.fileName = imageURI.substr(imageURI.lastIndexOf('/') + 1);

```

```

options.mimeType = "image/jpeg";
var params = {};
params.value1 = "test";
params.value2 = "param";
options.params = params;
var ft = new FileTransfer();
ft.upload(imageURI, encodeURI(""), this.win, this.fail, options);
//encodeURI("http://41.220.166.111/php/load/load_img.php")
},

// function picture sent with success
win: function(r) {
//
// console.log("Code = " + r.modificationTime);
// console.log("Response = " + r.response);
// console.log("Sent = " + r.bytesSent);
uploadedImages++;
// console.log('--num img-->' + numImage);
// console.log('--up img-->' + uploadedImages);
// console.log('--[imagID]-->' + imagID);
if (uploadedImages == numImage) {
me.sendAlpha();
}
},
fail: function(error) {
Ext.Msg.alert('Erro de envio!', 'Verifique serviço de dados. Código=' + error.code);
},

// function sending each data record
/* legend for server response messages :
0 envio + erro
1 falha
2 envio + sucesso
*/
sendAlpha : function() {
//me.uploadedImages == numImage
// console.log(allData);
allData.push(imagID);
imagID = [];
var dataToBeSentToServer = Ext.JSON.encode(allData);
Ext.Ajax.request({
//url: 'http://41.220.166.111/php/load/load.php', // real server address
url: '',
method: 'POST',
jsonData: dataToBeSentToServer,
success : function(response) {
//console.log('-->success ');
var text = 0;
text = response.responseText;
//console.log(text);
if (text == 2) {
currentfacilitySub.set('submitted', true);
currentfacilitySub.setDirty();
SportApp.app.getController("facilityController").getfacilityListView().facilityList.refresh();
}
// sync with proxy
Ext.getStore('facilityStore').sync();
Ext.Msg.show({
title: 'Registro submetido com sucesso.'
});
} else {
Ext.Msg.show({
title: 'Erro no servidor.'
});
}
Ext.Viewport.setMasked(false);
},
failure : function() {
var text = 1;
//console.log(text);
//console.log('--> failure ');
Ext.Msg.show({
title: 'Registro não submetido.'
});
Ext.Viewport.setMasked(false);
}
});
// console.log(text);
// box.hide();
},

```

```

sendImg : function(record) {
// ----> start pictures sending cycle
imgURI.push(record.get('path'));
imagID.push(record.get('imageID'));
var imageURI = imgURI[0];
var options = new FileUploadOptions();
options.fileKey = "file ";
options.fileName = imageURI.substr(imageURI.lastIndexOf('/') + 1);
options.mimeType = "image/jpeg ";
var params = {};
params.regID = identif;
options.params = params;
var ft = new FileTransfer();
ft.upload(imageURI, encodeURI("http://41.220.166.111/php/load/load_img.php"), me.win(imagID),
// ends cycle
me.fail, options);
imgURI = []; // cleans picture path
},

// submit records 1 by 1 :
submit : function(options, id) {
Ext.Viewport.setMasked({
xtype: 'loadmask',
message: 'A enviar registo... Por favor aguarde.'
});
currentfacilitySub = Ext.getStore('facilityStore').findRecord('id', id);
var currentImage = Ext.getStore('ImageStore');
currentImage.filter('facilityID', id);
var rangeImage = Ext.getStore('ImageStore').getRange();
numImage = rangeImage.length;
// console.log("----> "+numImage+" <---");
// console.log(currentfacility);
// console.log(currentImage);
allData = [];
imgURI = [];
identif = [];
uploadedImages = 0;
//var teste = 0;
identif.push(currentfacilitySub.getData());
imagID = [];
allData.push(currentfacilitySub.getData());
if (numImage > 0) {
currentImage.each(function(record) {
me.sendImg(record);
});
//Ext.Msg.alert('Envio de dados', 'Dados submetidos para o servidor.');
// ----> ends pictures sending cycle
} else {
me.sendAlpha();
// Ext.Msg.alert('Envio de dados', 'Dados submetidos para o servidor.');
}
//imagID=[]; //limpa o id das fotografias
currentImage.clearFilter();
},

// submit all data records (checked as submitted = false)
winAll : function(r) {
//
// console.log("Code = " + r.modificationTime);
// console.log("Response = " + r.response);
// console.log("Sent = " + r.bytesSent);
uploadedImages++;
console.log('--num img---->' + numImage);
console.log('--up img-->' + uploadedImages);
console.log('[imagID] ' + imagID);
if (uploadedImages == numImage) {
//me.sendAlphaAll();
Ext.Msg.alert('Envio de dados', 'Dados submetidos para o servidor.');
}
},

sendAlphaAll : function() {
// ----> sending alphanumeric data if send pictures = success :
allData2.push(imagID);
var dataToBeSentToServer = Ext.JSON.encode(allData2);
//console.log('----> allData2 ' + dataToBeSentToServer);
Ext.Ajax.request({
//url: 'http://41.220.166.111/php/load/load.php', // real server address

```

```

url: '',
    method: 'POST',
jsonData: dataToBeSentToServer,
success : function(response) {
var text = response.responseText;
if (text != 2) {
//console.log('---> '+text+' <---');
Ext.Msg.show({
title: 'Erro no servidor.'
});
} else {
Ext.Msg.show({
title: 'Registos submetidos com sucesso.'
});
//console.log('--->alldata submitted');
fileStore.each(function(record) {
record.set('submitted', true);
record.setDirty();
Ext.getStore('facilityStore').sync();
//record.sync();
});
//Ext.getStore('facilityStore').setDirty();
SportApp.app.getController("facilityController").getfacilityListView().facilityList.refresh();
;
// sync with proxy
//console.log('--->data stored');
}
Ext.Viewport.setMasked(false);
},
failure : function() {
Ext.Msg.show({
title: 'Registos não submetidos.'
});
//console.log('---> failure all data');
Ext.Viewport.setMasked(false);
}
}); // ---> ends of cycle
allData2 = []; // clean IDs
imagID = []; // clean pictures IDs
},

sendImgAll : function(record) {
// ---> start pictures sending cycle
imgURI.push(record.get('path'));
imagID.push(record.get('imageID'));
var imageURI = imgURI[0];
var options = new FileUploadOptions();
options.fileKey = "file";
options.fileName = imageURI.substr(imageURI.lastIndexOf('/') + 1);
options.mimeType = "image/jpeg";
var params = {};
params.regID = allData;
options.params = params;
var ft = new FileTransfer();
ft.upload(imageURI, encodeURIComponent("http://41.220.166.111/php/load/load_img.php"), me.winAll(), //
    ends cycle
me.fail, options);
imgURI = []; // clean picture path
},

submitAll : function(options) {
Ext.Viewport.setMasked({
xtype: 'loadmask',
message: 'A enviar registos... Por favor aguarde.'
});
fileStore = Ext.getStore('facilityStore'),
allData = []; // var holding picture ID in records
imgURI = []; // var holding path to the picture
allData2 = []; // var holding all alphanumeric data
imagID = []; // var holding picture ID
fileStore.filter('submitted', false);
fileStore.each(function(record) {
allData.push(record.get('id'));
//console.log(allData);
var currentImage = Ext.getStore('ImageStore');
currentImage.filter('facilityID', allData);
var rangeImage = Ext.getStore('ImageStore').getRange();
numImage = rangeImage.length;
//console.log("---> "+numImage+" <---");

```

```

uploadedImages = 0;
allData2.push(record.getData());
if (numImage > 0) {
// -----> start pictures sending cycle
currentImage.each(function(record) {
me.sendImgAll(record);
});
} // ends cycle
if (numImage == 0) {
// console.log('---> numI : ' + numImage);
me.sendAlphaAll();
Ext.Msg.alert('Envio de dados', 'Dados submetidos para o servidor.');
```

```

}
});
fileStore.clearFilter();
Ext.getStore('facilityStore').clearFilter();
},

showstartview : function(direction) {
this.getMainView().setActiveItem(
this.getStartView());
},

showcamera : function(options) {
// number of pictures condition (max number = 5)
if (Ext.getStore('ImageStore').getRange().length > 4) {
Ext.Msg.alert('', 'Máximo de 5 imagens permitidas!');
return;
} else {
var sourceType = Camera.PictureSourceType.PHOTOLIBRARY;
//var pictureSource=navigator.camera.PictureSourceType;
//var sourceType = Camera.PictureSourceType.SAVEDPHOTOALBUM;
// phoneGap API format : navigator.( cameraSuccess , cameraError , [ cameraOptions ] );
navigator.camera.getPicture(onSuccess, onFail, {
quality: 50,
destinationType: Camera.DestinationType.FILE_URI,
sourceType: sourceType,
mediaType: Camera.MediaType.PICTURE
//saveToPhotoAlbum: false
});

function onSuccess(imageURI) {
//console.log("camera onSuccess");
// creates image ID
var now = new Date();
var newId = now.getTime();
// var imName = newId + '.jpg';
var imagePath = imageURI;
// add picture to database
// get current record to get correspondent ID
var currentfacility = SportApp.app.getController("facilityController").getDataCollectingView()
().getRecord();
SportApp.app.getController("facilityController").getDataCollectingView().updateRecord(
currentfacility);
// create entry for the image
var imageEntry = Ext.create("SportApp.model.ImageModel", {
imageID: newId,
facilityID: currentfacility.get("id"),
path: imagePath,
saved: false
});
// save entry
Ext.getStore('ImageStore').add(imageEntry);
Ext.getStore('ImageStore').sort([
{
property: 'facilityID',
direction: 'DESC'
}
]);
}

function onFail(message) {
// alert('Failed because: ' + message);
//console.log("camera onFail");
}
},

showlistview : function(options) {
/*

```

```

creates a nomedia extension – ensures device does not run over all tiles images :
window.resolveLocalFileSystemURI("file:///mnt/extSdCard/tilecache/.nomedia", fileSuccess,
    fileError);
function fileSuccess(){
    return
}
function fileError(){
    window.requestFileSystem(LocalFileSystem.PERSISTENT, 0, gotFS, fail);
function gotFS(fileSystem){
    console.log('---> nomedia nao existe');
    fileSystem.root.getFile("/mnt/extSdCard/tilecache/.nomedia", {create: true, exclusive: false
        }, gotFileEntry, fail);
};
function gotFileEntry() {
    console.log('--->.nomedia criado');
}
function fail(evt) {
    console.log('.nomedia nao criado');
}
};
*/

this.getMainView().animateActiveItem(
this.getfacilityListView(), {
type: 'slide',
direction: 'right'
});
SportApp.app.getController("MapController").dockmap("mapComponentLV");
SportApp.app.getController("MapController").switch2listmode();
};
showdatacollectingview : function(options) {
    console.log("Número de imagens: " + Ext.getStore('ImageStore').getRange().length);
    this.getMainView().animateActiveItem(
    this.getDataCollectingView(), {
type: 'slide',
direction: 'left'
});
};
SportApp.app.getController("MapController").dockmap("mapComponentDCV");
},
deletefacility : function(options) {
    var currentfacility = this.getDataCollectingView().getRecord();
    if (Ext.getStore('facilityStore').findRecord('id', currentfacility.get("id"))) { // data.
    Ext.getStore('facilityStore').remove(currentfacility);
    }
    Ext.getStore('facilityStore').sync();
    this.getfacilityListView().refreshList();
    var controller = SportApp.app.getController("MapController");
    controller.showallfacilitys();
    },
deletepicture : function(options, item) {
    if (Ext.getStore('ImageStore').findRecord('imageID', item.get("imageID"))) {
    Ext.getStore('ImageStore').remove(item);
    }
    },

// removes picture permanently (hard drive)
deletepicturephysically : function(options, item) {
    var imageURI = item.data.path;
    window.resolveLocalFileSystemURI(imageURI, resOnSuccess, resOnError);

    function resOnSuccess(entry) {
        entry.remove(success, resOnError);
    };
    function success(entry) {
        // alert('succesfully deleted');
    }
    function resOnError(entry) {
        alert("Verifique operação!");
    };
    },

// checks image store for unsaved images and saves them permanently
saveimages : function(options) {
    var storedImages = Ext.getStore('ImageStore').getRange();
    console.log(storedImages.length);
    for (var i = 0; i < storedImages.length; i++) {
    var curlImage = storedImages[i];
    if (curlImage.get("saved") == false) {
    var imageURI = curlImage.get("path");

```

```

var facilityID = curImage.get("facilityID");
var facilityStr = facilityID.toString();
var imName = curImage.get("imageID") + '.jpg';
console.log(imageURI + ' - ' + imName);
window.resolveLocalFileSystemURI(imageURI, resOnSuccess, resOnError);

function resOnSuccess(entry) {
window.requestFileSystem(LocalFileSystem.PERSISTENT, 0, function(fileSys) {
// creates pictures folder
fileSys.root.getDirectory("atlasPics", {
create: true,
exclusive: false
},
function(directory) {
// creates folder for this facility
directory.getDirectory(facilityStr, {
create: true,
exclusive: false
},
function(directory) {entry.copyTo(
directory,
imName,
success,
resOnError);
},
resOnError);
},
resOnError);
},
resOnError);
};

function success(entry) {
curImage.data.path = entry.toURL();
curImage.data.saved = true;
};
function resOnError(entry) {
alert("Verifique operação!");
};
}
},
},
/*
checks image folder for images and saves locations and IDs in image store
(not permanently, only for the session)
*/
loadimages : function(options) {
// console.log("load images");
window.requestFileSystem(LocalFileSystem.PERSISTENT, 0, function(fileSys) {
fileSys.root.getDirectory("atlasPics", {
create: true,
exclusive: false
},
function(dirEntry) {
// gets all folders in this directory
var directoryReader = dirEntry.createReader();
directoryReader.readEntries(dirSuccess, fail);

function dirSuccess(dirEntries) {
var i;
for (i = 0; i < dirEntries.length; i++) {
if (dirEntries[i].isDirectory) {
var imageReader = dirEntries[i].createReader();
imageReader.readEntries(imSuccess, fail);

function imSuccess(imEntries) {
var j;
for (j = 0; j < imEntries.length; j++) {
if (imEntries[j].isFile) {
// adds to store
var imagePath = imEntries[j].toURL();
var curImageID = imEntries[j].name.split('.')[0];
// extracts facilityID from path
var splittedImagePath = imagePath.split('/');
var curfacilityID = splittedImagePath[splittedImagePath.length - 2];
// creates entry for the image
var imageEntry = Ext.create("SportApp.model.ImageModel", {
imageID: curImageID,
facilityID: curfacilityID,

```

```

path: imagePath,
saved: true
});
// add picture to database
Ext.getStore('ImageStore').add(imageEntry);
//console.log(imageEntry);
}
}
}
}
}
function fail(error) {
alert("Erro na visualização de conteúdos: " + error.code);
}
},
resOnError);
},
resOnError);
function resOnError(entry) {
alert("alerta 3 " + entry.code);
};
},
canceledit : function(options) {
this.getMainView().animateActiveItem(
this.getStartView(), {
type: 'slide',
direction: 'right'
});
},
});
});

```

Listing B.9 Map Controller.

```
var watchID = null;
var coordinates;

var map;
var currentFacilityMarker;
var buffer;
var positionMarker;

// Layers
var accuracyLayer;
var allFacilitysLayer;
var currentFacilityLayer;
var currentPositionLayer;

// Projections
var projLonLat = new OpenLayers.Projection("EPSG:4326"); // WGS 1984
var projMercator = new OpenLayers.Projection("EPSG:3857"); // Spherical Mercator

var deviceProjection = projLonLat; // the device's Projection
var mapProjection = projMercator; //projMercator; // the base layers projection

Ext.define(
'SportApp.controller.MapController',
{
extend : 'Ext.app.Controller',
config : {
refs : {
dataCollectingView : "datacollectingview",
mainView : "mainview",
startView : "startview",
facilityListView : "facilitylistview"
},
control : {
dataCollectingView : {
stopwatchingposition : "stopwatchingposition",
setPosition : "setPosition",
},
mainView : {},
startView : {
startwatchingposition : "startwatchingposition",
},
facilityListView : {
changelocation : "changelocation",
switch2listmode : "switch2listmode",
}
},
},
/* starts cordova watchposition function and handles
successful localization or errors
*/
startwatchingposition : function(options) {
// maximumAge: object value in milliseconds
var options = {
maximumAge : 30000,
timeout : 10000000,
enableHighAccuracy : true
};
watchID = navigator.geolocation.watchPosition(onSuccess, onError, options);
// onSuccess Geolocation
function onSuccess(p) {
// save position
coordinates = {
longitude : p.coords.longitude,
latitude : p.coords.latitude,
accuracy : p.coords.accuracy
};
// update map
SportApp.app.getController("MapController").changeMapPosition(
p.coords.longitude,
p.coords.latitude,
p.coords.accuracy
);
// show set position button
SportApp.app.getController("MapController").getDataCollectingView()
.setPositionButton
.setDisabled(false);
}
// onError callback receives position Error object
function onError(error) {
if (watchID != null) {
navigator.geolocation.clearWatch(watchID);
watchID = null;
}
switch (error.code) {
case error.PERMISSION_DENIED:
Ext.Msg.alert('Imposs\xE9vel obter localiza\xE7\xE3o',
'Servi\xE7o de localiza\xE7\xE3o desativado.');
```



```

}
else {
var extentOfMozambique = new OpenLayers.Bounds(
3339584, -3123471, 4564100, -1118890
);
// if no records stored —> zoom to Mozambique extension
map.zoomToExtent(extentOfMozambique);
}
},
// docks to map component to another container
dockmap : function(componentDIV) {
if (map) {
map.render(componentDIV);
}
},
// initializes map
initializeMap : function(componentDIV) {
// console.log("initialize map");
if (map) {
this.dockmap(componentDIV);
return;
}
// initialize Layers
accuracyLayer = new OpenLayers.Layer.Vector("Accuracy",
{
displayInLayerSwitcher : false
});
// all facilities layer gets special cluster strategy
var strategy = new OpenLayers.Strategy.Cluster({
distance : 20,
threshold : 2
});
var style = new OpenLayers.Style({
graphicName : "cross",
strokeColor : "#00807a",
fillColor : "#00807a",
graphicName : "cross",
pointRadius : "${size}",
strokeWidth : 1,
rotation : 45,
strokeLinecap : "butt",
label : "${label}",
fontSize : '20'
},
{
context : {
size : function(feature) {
return (feature.cluster) ? 23 : 14;
},
label : function(feature) {
if (feature.cluster) {
return feature.attributes.count;
}
}
}
});
allFacilitysLayer = new OpenLayers.Layer.Vector(
"All Facilitys", {
displayInLayerSwitcher : false ,
strategies : [ strategy ],
styleMap : new OpenLayers.StyleMap(
{
"default" : style
})
});
currentFacilityLayer = new OpenLayers.Layer.Vector(
"Current Facility", {
displayInLayerSwitcher : false
});
currentPositionLayer = new OpenLayers.Layer.Vector(
"Current Position", {
displayInLayerSwitcher : false
});

/*
-----> TILES tilecache layer <-----
--> path configuration :
in offlinelayer :
path to sdcard SAMSUNG : /mnt/extSdCard/tilecache

```

```

path to sdcard ASUS : /Removable/MicroSD/tilecache
*/

// get external files directory
// var retrievedObject = localStorage.getItem('testObject');

var mercatorOfflineExtent = new OpenLayers.Bounds(
3339584, -3123471, 4564100, -1118890);
// tiles layer :
var offlineLayer = new OpenLayers.Layer.TileCache(
"Camada base", [ "/mnt/extSdCard/tilecache" ],
"Export_Output", {
maxResolution : mercatorOfflineExtent.getWidth() / 256,
maxExtent : mercatorOfflineExtent,
numZoomLevels : 9,
//projection: "EPSG:3857",
});
// osm layer :
var oSMLayer = new OpenLayers.Layer.OSM("OpenStreetMap",
null, {
numZoomLevels : 20,
});
// google layer :
/*
var googleLayer = new OpenLayers.Layer.Google(
"Google Streets",
{
numZoomLevels : 20,
sphericalMercator: true,
});
*/
map = new OpenLayers.Map({
div : componentDIV,
theme : null,
projection : mapProjection,
controls : [
new OpenLayers.Control.TouchNavigation({
dragPanOptions : {
// non zero interval cause performance prob on some devices
interval : 0,
enableKinetic : true
}
}),
new OpenLayers.Control.LayerSwitcher(),
],
layers : [
offlineLayer,
//googleLayer,
oSMLayer,
accuracyLayer,
allFacilitysLayer,
currentFacilityLayer,
currentPositionLayer
]
});

SportApp.app.getController("MapController").showallfacilitys();
// map.setCenter(new OpenLayers.LonLat(35.994, -16.135).transform(deviceProjection,
mapProjection), 5);
map.zoomToExtent(mercatorOfflineExtent);
},

//changes position of marker for current location and accuracy
changeMapPosition : function(lon, lat, acc) {
if (buffer)
accuracyLayer.removeFeatures([ buffer ]);
if (positionMarker)
currentPositionLayer.removeFeatures([ positionMarker ]);
var newPosition = new OpenLayers.LonLat(lon, lat).transform(
deviceProjection,
mapProjection
);
var point = new OpenLayers.Geometry.Point(
newPosition.lon,
newPosition.lat
);
// update map

```

```

// map.setCenter(newPosition, 10);

// set marker
positionMarker = new OpenLayers.Feature.Vector(
point,
null, {
externalGraphic : 'circle20.png',
graphicWidth : 20,
graphicHeight : 20,
graphicXOffset : -10,
graphicYOffset : -10,
graphicOpacity : 1,
});

// set buffer
var bufferStyle = OpenLayers.Util.extend({});
bufferStyle.fillOpacity = 0.2;
bufferStyle.opacity = 0.9;
bufferStyle.strokeWidth = 1;
bufferStyle.strokeStyle = "grey";
bufferStyle.fillColor = "grey";
var bufferGeometry = OpenLayers.Geometry.Polygon.createRegularPolygon(
point,
acc,
40,
0
);
buffer = new OpenLayers.Feature.Vector(bufferGeometry,
null,
bufferStyle
);
accuracyLayer.setVisibility(true);
currentPositionLayer.setVisibility(true);
accuracyLayer.addFeatures([ buffer ]);
currentPositionLayer.addFeatures([ positionMarker ]);
},

/*
changes the position of marker for current selected facility and zooms to fixed level
*/
changelocation : function(options, latitude, longitude) {
this.changeFacilityPosition(
longitude,
latitude,
10
);
},

/*
changes the position of marker for currently selected facility
*/
changeFacilityPosition : function(lon, lat, zoom) {
// add marker
if (currentFacilityMarker) {
currentFacilityLayer.removeFeatures([ currentFacilityMarker ]);
}

// Center of the map
var newPosition = new OpenLayers.LonLat(lon, lat).transform(
deviceProjection,
mapProjection
);

// transform from WGS84 to Spherical Mercator Projection :
var point = new OpenLayers.Geometry.Point(
newPosition.lon,
newPosition.lat
);

// cross marker
var style_cross = OpenLayers.Util.extend(
{},
OpenLayers.Feature.Vector.style['default']
);

style_cross.fillOpacity = 1,
style_cross.strokeStyle = "#00807a";
style_cross.fillColor = "3385EB";
style_cross.graphicName = "cross";

```

```

style_cross.pointRadius = 20;
style_cross.strokeWidth = 4;
style_cross.rotation = 0;
style_cross.strokeLinecap = "butt";

currentFacilityMarker = new OpenLayers.Feature.Vector(
point, null, style_cross);
currentFacilityLayer.setVisibility(true);
currentFacilityLayer.addFeatures([ currentFacilityMarker ]);
map.setCenter(newPosition, zoom);
},

switch2listmode : function(options) {
if (accuracyLayer)
accuracyLayer.setVisibility(false);
if (allFacilitysLayer)
allFacilitysLayer.setVisibility(true);
if (currentFacilityLayer)
currentFacilityLayer.setVisibility(false);
if (currentPositionLayer)
currentPositionLayer.setVisibility(false);
},

switch2editmode : function(options) {
if (accuracyLayer)
accuracyLayer.setVisibility(false);
if (allFacilitysLayer)
allFacilitysLayer.setVisibility(true);
if (currentFacilityLayer)
currentFacilityLayer.setVisibility(true);
if (currentPositionLayer)
currentPositionLayer.setVisibility(false);
},

switch2newmode : function(options) {
if (accuracyLayer)
accuracyLayer.setVisibility(false);
if (allFacilitysLayer)
allFacilitysLayer.setVisibility(true);
if (currentFacilityLayer)
currentFacilityLayer.setVisibility(false);
if (currentPositionLayer)
currentPositionLayer.setVisibility(false);
},

zoomto : function(options, longitude, latitude, zoom) {
// update map
var newPosition = new OpenLayers.LonLat(
latitude,
longitude
)
.transform(
deviceProjection,
mapProjection
);
map.setCenter(
newPosition,
zoom
);
}
});

```

Listing B.10 PictureBox Buttons Functions.

```
// previous picture event function :
previousPictureButtonTap: function() {
    activePictureIndex--;
    var data = Ext.getStore('ImageStore').getRange();
    if (activePictureIndex < 0) {
        activePictureIndex = 0;
        return;
    }
    if (activePictureIndex == 0) {
        SportApp.view.DataCollectingView.prevPictureButton.disable();
    }
    if (activePictureIndex == data.length - 2) {
        SportApp.view.DataCollectingView.nextPictureButton.enable();
    }
    var item = data[activePictureIndex];
    SportApp.view.DataCollectingView.pictureView.removeAll();
    SportApp.view.DataCollectingView.pictureView.add({
        xtype: 'container',
        width: Ext.Viewport.getSize().width / 2 - 20,
        height: Ext.Viewport.getSize().height / 2 - 60,
        html: '<CENTER>'
    });
}

// next picture event function :
nextPictureButtonTap: function() {
    activePictureIndex++;
    var data = Ext.getStore('ImageStore').getRange();
    if (activePictureIndex >= data.length) {
        activePictureIndex--;
        return;
    }
    if (activePictureIndex == 1) {
        SaltApp.view.DataCollectingView.prevPictureButton.enable();
    }
    if (activePictureIndex == data.length - 1) {
        SaltApp.view.DataCollectingView.nextPictureButton.disable();
    }
    var item = data[activePictureIndex];
    SaltApp.view.DataCollectingView.pictureView.removeAll();
    SaltApp.view.DataCollectingView.pictureView.add({
        xtype: 'container',
        width: Ext.Viewport.getSize().width / 2 - 20,
        height: Ext.Viewport.getSize().height / 2 - 60,
        html: '<CENTER>'
    });
}

// delete picture event function :
deletePictureButtonTap: function() {
    SportApp.view.DataCollectingView.pictureView.hide();
    Ext.Msg.confirm("Apagar imagem.", "Tem a certeza?", function(value) {
        if (value == 'no') {
            SportApp.view.DataCollectingView.pictureView.show();
            return;
        } else {
            var data = Ext.getStore('ImageStore').getRange();
            var item = data[activePictureIndex];
            SportApp.app.getController("MineController").getDataCollectingView().fireEvent("
                deletpicture", this, item);
            SportApp.view.DataCollectingView.fireEvent("deletpicturephysically", this, item)
                ;
        }
    });
}
}
```

Listing B.11 m-SportGIS index.html.

```
<!DOCTYPE HTML>
<html>
<head>
<title>m-SportGIS</title>

<script src='http://maps.google.com/maps?file=api&v=2&key=<google-maps-key>'></script>
</script>
<link href="resources/css/google.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="OpenLayers.js"></script>
<link rel="stylesheet" href="sencha/resources/css/sencha-touch.css" type="text/css">
<link href="resources/css/app.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="sencha/sencha-touch-all-debug.js"></script>
<!-- <script type="text/javascript" src="sencha/sencha-touch-all-compat.js"></script> -->
<script type="text/javascript" charset="utf-8" src="cordova-2.5.0.js"></script>
<script type="text/javascript" charset="utf-8" src="app.js"></script>
</script>
<!--localStorage.clear();-->
Ext.Loader.setConfig({ disableCaching: false });
Ext.Ajax.setDisableCaching(false);
</script>
</head>
<body></body>
</html>
```

Listing B.12 Java Statement to Launch Cordova API.

```
package pt.unl.isegi.labnt.m-sportgis;

import org.apache.cordova.DroidGap;

import android.os.Bundle;

public class m-SportGIS extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.setIntegerProperty("loadUrlTimeoutValue", 60000);
        // file:///android_asset URI maps to assets directory
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

Listing B.13 Android Manifest.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pt.unl.isegi.labnt.m-sportgis"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="14" />

    <supports-screens
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:resizeable="true"
        android:anyDensity="true"
    />

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />

    <application android:debuggable="true" android:icon="@drawable/icon" android:label="@string/app_name" android:hardwareAccelerated="true">
        <activity android:configChanges="orientation|keyboardHidden" android:screenOrientation="sensorLandscape" android:name="pt.unl.isegi.labnt.m-sportgis.m-SportGIS"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Appendix C

MapServer Program Listings

Listing C.1 Example of TileCache Configuration File.

```
# Configuration for TileCache (tilecache.cfg)

[cache]
type= Disk
base= /home/abreu/tilecache/Export_Output

[map9] # zoom level 10
type= WMSLayer
url= http://localhost/cgi-bin/mapserv?
map=/var/www/projects/sportgis/mapdata/poi2.map

layers= adm_postos,reserv_areas,hid_areas,hid_lines_pri,
rails,roads_pri,roads_sec,moz_POI,cap_posto_outros

#Mozambique extent (Lon : Lat): 3339584 -3123471 : 4564100 -1118890
bbox= 3339584 -3123471 4564100 -1118890
srs= EPSG:3857

#maxresolution= (MAX(X)-MIN(x))/256 (tile size in pixels)
#(4564100-3339584)/256
maxresolution= 4783.265625

levels= 10
extent_type= loose
extension= png
mime_type= image/png
```

Listing C.2 Header Settings for Map Files.

```
# Map file created from QGIS project file /home/abreu/Documents/Trabalho/SportGIS/moz_3857.
# qgs
# (Created with PyQgis MapServer Export plugin)
# MAP object definitions
MAP
    NAME "QGIS-MAP"
    # Map image size
    SIZE 800 800
    UNITS meters

# Symbology
# SYMBOLSET './symbols/symbols.txt'
SYMBOL
    NAME 'star'
    TYPE VECTOR
    FILLED TRUE
    POINTS # [X Y]
        0 .375
        .35 .375
        .5 0
        .65 .375
        1 .375
        .75 .625
        .875 1
        .5 .75
        .125 1
        .25 .625
    END # end of POINTS
END # end of SYMBOL
SYMBOL
    NAME 'circle'
    TYPE ELLIPSE
    FILLED TRUE
    POINTS
        1 1
    END # end of POINTS
END # end of SYMBOL

FONTSET 'font.txt' # path './fonts/fonts.txt'

EXTENT 3339584 -3123471 4564100 -1118890

PROJECTION
    'proj=merc'
    'a=6378137'
    'b=6378137'
    'lat_ts=0.0'
    'lon_0=0.0'
    'x_0=0.0'
    'y_0=0'
    'k=1.0'
    'units=m'
    'nadgrids=@null'
    'wktext'
    ''
    'no_defs'
END # end of PROJECTION

# Background color for the map canvas
IMAGECOLOR 255 255 255
IMAGEQUALITY 95
IMAGETYPE agg
OUTPUTFORMAT
    NAME agg
    DRIVER AGG/PNG
    IMAGEMODE RGB
    FORMATOPTION "INTERLACE=OFF"
    TRANSPARENT ON
END # end of OUTPUTFORMAT

# Legend
LEGEND
    IMAGECOLOR 255 255 255
    STATUS ON
    KEYSIZE 18 12
    LABEL
        TYPE BITMAP
        SIZE MEDIUM
        COLOR 0 0 89
    END # end of LABEL
END # end of LEGEND

# WEB interface definition. Output folder (IMAGEPATH). URL to IMAGEPATH (IMAGEURL)
WEB
    IMAGEPATH '/tmp/'
    IMAGEURL '/tmp/'

# WMS server settings
METADATA
    'ows_title' 'QGIS-MAP'
```

```

        'ows_onlineresource' 'http://localhost/cgi-bin/mapserv?map=/home/abreu/mapdata/
        poi.map'
        'ows_srs' 'EPSG:3857'
        'ows_enable_request' '*'
    END # end of METADATA

#Scale range for web interface
    TEMPLATE 'fooOnlyForWMSGetFeatureInfo'
END # end of WEB

```

Listing C.3 Map Object Code Fragment in moz_3857.map

```

# LAYER object definitions (Primary road ways)
LAYER
    NAME 'roads_pri'
    TYPE LINE
    DUMP true
    TEMPLATE fooOnlyForWMSGetFeatureInfo
    EXTENT 3339584 -3123471 4564100 -1118890
    DATA '/home/abreu/Documents/Trabalho/SportGIS/Shapefiles/epsg3857/roads_pri.shp'

    METADATA
        'ows_title' 'roads_pri'
    END # end of METADATA
    STATUS OFF
    TRANSPARENCY 100
    PROJECTION
        'proj=merc'
        'a=6378137'
        'b=6378137'
        'lat_ts=0.0'
        'lon_0=0.0'
        'x_0=0.0'
        'y_0=0'
        'k=1.0'
        'units=m'
        'nadgrids=@null'
        'wktext'
        ''
        'no_defs'
    END # end of PROJECTION
    CLASS
        NAME 'roads_pri'
        STYLE
            WIDTH 2.8
            COLOR 220 147 0
        END # end of STYLE
    END # end of CLASS
END # end of LAYER

```

Appendix D

Stores

Listing D.1 Districts ("Distritos") Store.

```
Ext.define('SportApp.store.DistritosStore', {
    extend : 'Ext.data.Store',
    autoLoad: true,
    config : {
        fields:[
            { name: 'district', type: 'string' },
            { name: 'idDistrict', type: 'int' },
            { name: 'idProvince', type: 'int' }
        ],
        data :[
            { district : 'Ancuabe', idDistrict : '1', idProvince : '1' },
            { district : 'Balama', idDistrict : '2', idProvince : '1' },
            { district : 'Chiure', idDistrict : '3', idProvince : '1' },
            { district : 'Cidade De Pemba', idDistrict : '4', idProvince : '1' },
            { district : 'Ibo', idDistrict : '5', idProvince : '1' },
            { district : 'Macomia', idDistrict : '6', idProvince : '1' },
            { district : 'Mecufi', idDistrict : '7', idProvince : '1' },
            { district : 'Meluco', idDistrict : '8', idProvince : '1' },
            { district : 'Mocimboa Da Praia', idDistrict : '9', idProvince : '1' },
            { district : 'Montepuez', idDistrict : '10', idProvince : '1' },
            { district : 'Mueda', idDistrict : '11', idProvince : '1' },
            { district : 'Muidumbe', idDistrict : '12', idProvince : '1' },
            { district : 'Namuno', idDistrict : '13', idProvince : '1' },
            { district : 'Nangade', idDistrict : '14', idProvince : '1' },
            { district : 'Palma', idDistrict : '15', idProvince : '1' },
            { district : 'Pemba', idDistrict : '16', idProvince : '1' },
            { district : 'Quissanga', idDistrict : '17', idProvince : '1' },
            { district : 'Bilene', idDistrict : '18', idProvince : '2' },
            { district : 'Chibuto', idDistrict : '19', idProvince : '2' },
            { district : 'Chicualacuala', idDistrict : '20', idProvince : '2' },
            { district : 'Chigubo', idDistrict : '21', idProvince : '2' },
            { district : 'Chokwe', idDistrict : '22', idProvince : '2' },
            { district : 'Cidade De Xai-Xai', idDistrict : '23', idProvince : '2' },
            { district : 'Guija', idDistrict : '24', idProvince : '2' },
            { district : 'Mabalane', idDistrict : '25', idProvince : '2' },
            { district : 'Mandlakaze', idDistrict : '26', idProvince : '2' },
            { district : 'Massangena', idDistrict : '27', idProvince : '2' },
            { district : 'Massingir', idDistrict : '28', idProvince : '2' },
            { district : 'Xai-Xai', idDistrict : '29', idProvince : '2' },
            { district : 'Ilha Licom', idDistrict : '30', idProvince : '3' },
            { district : 'Ilha Risunodo', idDistrict : '31', idProvince : '4' },
            { district : 'Cidade De Inhambane', idDistrict : '32', idProvince : '5' },
            { district : 'Funhalouro', idDistrict : '33', idProvince : '5' },
            { district : 'Govuro', idDistrict : '34', idProvince : '5' },
            { district : 'Homoine', idDistrict : '35', idProvince : '5' },
            { district : 'Inharrime', idDistrict : '36', idProvince : '5' },
            { district : 'Inhassoro', idDistrict : '37', idProvince : '5' },
            { district : 'Jangamo', idDistrict : '38', idProvince : '5' },
            { district : 'Mabote', idDistrict : '39', idProvince : '5' },
            { district : 'Massinga', idDistrict : '40', idProvince : '5' },
            { district : 'Maxixe', idDistrict : '41', idProvince : '5' },
            { district : 'Morrumbene', idDistrict : '42', idProvince : '5' },
            { district : 'Panda', idDistrict : '43', idProvince : '5' },
            { district : 'Vilankulo', idDistrict : '44', idProvince : '5' },
            { district : 'Zavala', idDistrict : '45', idProvince : '5' },
            { district : 'Lago Niassa', idDistrict : '46', idProvince : '6' },
            { district : 'Barue', idDistrict : '47', idProvince : '7' },
            { district : 'Cidade De Chimoio', idDistrict : '48', idProvince : '7' },
            { district : 'Gondola', idDistrict : '49', idProvince : '7' },
            { district : 'Guro', idDistrict : '50', idProvince : '7' },
            { district : 'Machaze', idDistrict : '51', idProvince : '7' },
            { district : 'Macossa', idDistrict : '52', idProvince : '7' },
            { district : 'Manica', idDistrict : '53', idProvince : '7' },
            { district : 'Mossurize', idDistrict : '54', idProvince : '7' },
            { district : 'Sussundenga', idDistrict : '55', idProvince : '7' },
            { district : 'Tambara', idDistrict : '56', idProvince : '7' },
            { district : 'Boane', idDistrict : '57', idProvince : '8' },
            { district : 'Cidade Da Matola', idDistrict : '58', idProvince : '8' },
            { district : 'Cidade De Maputo', idDistrict : '59', idProvince : '8' },
        ]
    }
});
```

```

{ district : 'Magude' , idDistrict : '60' , idProvince : '8' },
{ district : 'Manhia' , idDistrict : '61' , idProvince : '8' },
{ district : 'Marracuene' , idDistrict : '62' , idProvince : '8' },
{ district : 'Matutuine' , idDistrict : '63' , idProvince : '8' },
{ district : 'Moamba' , idDistrict : '64' , idProvince : '8' },
{ district : 'Namaacha' , idDistrict : '65' , idProvince : '8' },
{ district : 'Angoche' , idDistrict : '66' , idProvince : '9' },
{ district : 'Cidade De Nampula' , idDistrict : '67' , idProvince : '9' },
{ district : 'Erati' , idDistrict : '68' , idProvince : '9' },
{ district : 'Ilha De Mocambique' , idDistrict : '69' , idProvince : '9' },
{ district : 'Lalaua' , idDistrict : '70' , idProvince : '9' },
{ district : 'Malema' , idDistrict : '71' , idProvince : '9' },
{ district : 'Mecenta' , idDistrict : '72' , idProvince : '9' },
{ district : 'Mecuburi' , idDistrict : '73' , idProvince : '9' },
{ district : 'Memba' , idDistrict : '74' , idProvince : '9' },
{ district : 'Mogincual' , idDistrict : '75' , idProvince : '9' },
{ district : 'Mogovolas' , idDistrict : '76' , idProvince : '9' },
{ district : 'Moma' , idDistrict : '77' , idProvince : '9' },
{ district : 'Monapo' , idDistrict : '78' , idProvince : '9' },
{ district : 'Mossuril' , idDistrict : '79' , idProvince : '9' },
{ district : 'Muecate' , idDistrict : '80' , idProvince : '9' },
{ district : 'Murrupula' , idDistrict : '81' , idProvince : '9' },
{ district : 'Nacala' , idDistrict : '82' , idProvince : '9' },
{ district : 'Nacala-A-Velha' , idDistrict : '83' , idProvince : '9' },
{ district : 'Nacaraoa' , idDistrict : '84' , idProvince : '9' },
{ district : 'Nampula' , idDistrict : '85' , idProvince : '9' },
{ district : 'Ribaua' , idDistrict : '86' , idProvince : '9' },
{ district : 'Cidade De Lichinga' , idDistrict : '87' , idProvince : '10' },
{ district : 'Cuamba' , idDistrict : '88' , idProvince : '10' },
{ district : 'Lago' , idDistrict : '89' , idProvince : '10' },
{ district : 'Lichinga' , idDistrict : '90' , idProvince : '10' },
{ district : 'Majune' , idDistrict : '91' , idProvince : '10' },
{ district : 'Mandimba' , idDistrict : '92' , idProvince : '10' },
{ district : 'Marrupa' , idDistrict : '93' , idProvince : '10' },
{ district : 'Maua' , idDistrict : '94' , idProvince : '10' },
{ district : 'Mavago' , idDistrict : '95' , idProvince : '10' },
{ district : 'Mecanhelas' , idDistrict : '96' , idProvince : '10' },
{ district : 'Mecula' , idDistrict : '97' , idProvince : '10' },
{ district : 'Metarica' , idDistrict : '98' , idProvince : '10' },
{ district : 'Muembe' , idDistrict : '99' , idProvince : '10' },
{ district : 'Ngauma' , idDistrict : '100' , idProvince : '10' },
{ district : 'Nipepe' , idDistrict : '101' , idProvince : '10' },
{ district : 'Sanga' , idDistrict : '102' , idProvince : '10' },
{ district : 'Buzi' , idDistrict : '103' , idProvince : '11' },
{ district : 'Caia' , idDistrict : '104' , idProvince : '11' },
{ district : 'Chemba' , idDistrict : '105' , idProvince : '11' },
{ district : 'Cheringoma' , idDistrict : '106' , idProvince : '11' },
{ district : 'Chibabava' , idDistrict : '107' , idProvince : '11' },
{ district : 'Cidade Da Beira' , idDistrict : '108' , idProvince : '11' },
{ district : 'Dondo' , idDistrict : '109' , idProvince : '11' },
{ district : 'Gorongosa' , idDistrict : '110' , idProvince : '11' },
{ district : 'Machanga' , idDistrict : '111' , idProvince : '11' },
{ district : 'Maringue' , idDistrict : '112' , idProvince : '11' },
{ district : 'Marromeu' , idDistrict : '113' , idProvince : '11' },
{ district : 'Muanza' , idDistrict : '114' , idProvince : '11' },
{ district : 'Nhamatanda' , idDistrict : '115' , idProvince : '11' },
{ district : 'Angonia' , idDistrict : '116' , idProvince : '12' },
{ district : 'Cahora Bassa' , idDistrict : '117' , idProvince : '12' },
{ district : 'Changara' , idDistrict : '118' , idProvince : '12' },
{ district : 'Chifunde' , idDistrict : '119' , idProvince : '12' },
{ district : 'Chiuta' , idDistrict : '120' , idProvince : '12' },
{ district : 'Cidade De Tete' , idDistrict : '121' , idProvince : '12' },
{ district : 'Luenha' , idDistrict : '122' , idProvince : '12' },
{ district : 'Macanga' , idDistrict : '123' , idProvince : '12' },
{ district : 'Magoe' , idDistrict : '124' , idProvince : '12' },
{ district : 'Maravia' , idDistrict : '125' , idProvince : '12' },
{ district : 'Moatize' , idDistrict : '126' , idProvince : '12' },
{ district : 'Mutarara' , idDistrict : '127' , idProvince : '12' },
{ district : 'Tsangano' , idDistrict : '128' , idProvince : '12' },
{ district : 'Zumbu' , idDistrict : '129' , idProvince : '12' },
{ district : 'Alto Molocue' , idDistrict : '130' , idProvince : '13' },
{ district : 'Chinde' , idDistrict : '131' , idProvince : '13' },
{ district : 'Cidade De Quelimane' , idDistrict : '132' , idProvince : '13' },
{ district : 'Gile' , idDistrict : '133' , idProvince : '13' },
{ district : 'Gurue' , idDistrict : '134' , idProvince : '13' },
{ district : 'Ile' , idDistrict : '135' , idProvince : '13' },
{ district : 'Inhassunge' , idDistrict : '136' , idProvince : '13' },
{ district : 'Lugela' , idDistrict : '137' , idProvince : '13' },
{ district : 'Maganja Da Costa' , idDistrict : '138' , idProvince : '13' },

```

```
    { district : 'Milange' , idDistrict : '139' , idProvince : '13' },
    { district : 'Mocuba' , idDistrict : '140' , idProvince : '13' },
    { district : 'Mopeia' , idDistrict : '141' , idProvince : '13' },
    { district : 'Morrumbala' , idDistrict : '142' , idProvince : '13' },
    { district : 'Namacurra' , idDistrict : '143' , idProvince : '13' },
    { district : 'Namarroi' , idDistrict : '144' , idProvince : '13' },
    { district : 'Nicoadala' , idDistrict : '145' , idProvince : '13' },
    { district : 'Pebane' , idDistrict : '146' , idProvince : '13' }
  ]
});
```