FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Context-Based Cultural Visits

Mariana Figueiredo Moutinho Resende Assis

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Maria Teresa Magalhães da Silva Pinto de Andrade

Second Supervisor: Paula Maria Marques de Moura Gomes Viana

October 18, 2020

Resumo

Nas últimas décadas, houve enormes avanços na tecnologia móvel que permitiram um aumento do interesse em estudar e desenvolver sistemas móveis de realidade aumentada, principalmente na área de Património Cultural. Atualmente, a população depende cada vez mais dos telemóveis, por exemplo, quando visitam uma nova cidade, é com o auxílio dos dispositivos móveis que vão procurar informações sobre os monumentos e marcos históricos e esperam informações precisas e personalizadas.

Deste modo, surgiu a necessidade de estudar novas abordagens para a apresentação e sugestão de conteúdos digitais relacionados com cada local histórico e cultural. Estas abordagens terão de considerar informações contextuais sobre o visitante e as suas necessidades.

Este documento apresenta uma aplicação móvel de realidade aumentada, *NearHeritage*, que foi desenvolvida no âmbito da dissertação de mestrado em Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto (FEUP), em colaboração com o INESC TEC.

A investigação realizada foi centrada na utilização de tecnologias modernas para ajudar os visitantes a identificar e a explorar locais de Património Cultural. Desta forma, o utilizador irá receber não só os pontos de interesse da cidade próximos da sua localização, mas também informações detalhadas sobre cada local.

A solução apresentada utiliza sensores e hardware que fazem parte dos dispositivos Android, e tira partido de várias APIs (como a Foursquare API, a Google Maps SDK e a IntelContextSensing SDK) para recuperar informações sobre os locais e o contexto do visitante. Adicionalmente, esses componentes de hardware são cruciais para usar todo o potencial das ferramentas de realidade aumentada para criar conteúdos inovadores, que melhoraram a experiência do utilizador.

As experiências foram realizadas no Porto, Portugal, e os resultados finais demonstram que o conceito de uma aplicação MAR pode melhorar a experiência geral dos utilizadores, ajudando-os a descobrir e aprender mais sobre o Património Cultural que existe em cada cidade, através de uma aventura interativa, divertida e inesquecível.

Palavras-chave: Dispositivos Android, Realidade aumentada, Patrimônio cultural, Experiências interativas, Visitas culturais.

Abstract

Over the last two decades, there have been tremendous advances in mobile technologies, which have increased the interest in studying and developing mobile augmented reality systems, especially in the field of Cultural Heritage. Nowadays, people rely even more on smartphones, for example, when visiting a new city to search for information about monuments and landmarks, and the visitor expects precise and tailored information to his needs.

Therefore, researchers started to investigate innovative approaches for presenting and suggesting digital content related to cultural and historical places around the city, incorporating contextual information about the visitor and his needs.

This document presents a novel mobile augmented reality application, *NearHeritage*, that was developed within the scope of the master's thesis on Electrical and Computers Engineering from the Faculty of Engineering of Porto University (FEUP), in collaboration with INESC TEC.

The research carried out was focused on the importance of utilising modern technologies to assist the visitors in finding and exploring Cultural Heritage. In this way, it is provided not only the nearby points-of-interest of a city but also detailed information about each POI.

The solution presented uses built-in sensors and hardware of Android devices and takes advantage of various APIs (Foursquare API, Google Maps API and IntelContextSensing) to retrieve information about the landmarks and the visitor context. Also, these are crucial hardware components for implementing the full potential of augmented reality tools to create innovative contents that increase the overall user experience.

All the experiments were conducted in Porto, Portugal, and the final results showcase that the concept of a MAR application can improve the user experience in discovering and learning more about Cultural Heritage around the world, creating an interactive, enjoyable and unforgettable adventure.

Keywords: Android devices, Augmented reality, Cultural heritage, Interactive experiences, Cultural visits.

Acknowledgments

First of all, I would like to thank my dissertation advisors, Maria Teresa Andrade from FEUP – Faculdade de Engenheraia da Universidade do Porto and Paula Viana from INESC TEC – Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência, for all the support, help, incentive and valuable suggestions during the entire process of planning and development of this dissertation.

Second, I am very grateful to all my friends and course colleagues who accompanied me during these last five years: Tiago Soares, Ricardo Costa, Ricardo Sousa, Hélder Pereira, Inês Xavier, Diogo Mariano, Pedro Albergaria, Daniel Silva, Ricardo Freitas and Luís Ribeiro. All the friendship and companionship created unforgettable experiences and memories that will always be remembered with great fondness for the rest of my life. A special appreciation to João Granja, for all his encouragement and support that kept me sane and motivated in the development of the thesis during this pandemic time.

Lastly and most importantly, I would like to express my gratitude to my family, especially to my mother, my brother and my godparents, for the unconditional love, kindness and care throughout my life and for always being here for me, no matter what. Also, to my father for being an inspiration and a motivational force that led me in this path of engineering.

Mariana Assis

“Ninguém sabe o suficiente para fazer tudo sozinho.”

José Mariano Gago

Contents

List of Figures

List of Tables

Abbreviations and Symbols

APP	Application
API	Application Programming Interface
AR	Augmented Reality
AV	Augmented Virtuality
CARS	Context-Aware Recommender System
CH	Cultural Heritage
FEUP	Faculdade de Engenharia da Universidade do Porto
GPS	Global Positioning System
INESC TEC	Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência
LARS	Location-Aware Recommender System
MR	Mixed Reality
MAR	Mobile Augmented Reality
POI	Point of Interest
RS	Recommender System
REST	Representational State Transfer
SDK	Software Development Kit
UI	User Interface
VPN	Virtual Private Network
VR	Virtual Reality

Chapter 1

Introduction

The focus of this dissertation is on the development of a mobile application that resorts to the access and manipulation of built-in android sensors and augmented reality contents, in order to provide an interactive and richer outdoor experience to users in a cultural city visit.

With this purpose in mind, this chapter helps to understand the objective and motivation behind the dissertation theme, whilst also briefly discussing the background of the area of research.

1.1 Context

The development of digital technologies has significantly affected the evolution of cultural heritage dissemination, offering new possibilities for growth. Whether it is a mobile application, an online catalogue, or a social media exchange, these new paradigms are reshaping the way visitors discover, learn, and interact with cultural heritage around the world [1].

One of these technologies that offer new ways of interacting is augmented reality, which has enormous potential for the promotion and preservation of cultural heritage [1].

Augmented reality technology enables an enriched perspective of the real world by improving the available information about people, buildings or places in order to evoke and complement historical memories [2, 3].

For example, this technology can be employed to enrich the scene of a historical point of interest with digital information. In this way, it is possible to achieve a cultural guide that comes to life in real-time, so that visitors can obtain and interact with information about the POI [3, 4].

Therefore, emerging technologies can offer new challenges for access and display of multimedia content and information manipulation, such as augmented reality, artificial intelligence techniques, metadata or context information, that can be used to offer richer cultural heritage experiences.

1.2 Motivation

Nowadays, with the transformation of urban areas into smart cities, points of interest can offer new opportunities to take advantage of augmented reality technology, both in the area of promotion materials and in the CH sites that represent points of interest from historical, cultural or other views [5].

In this context, AR content can extend human perception, allowing people to interact with their surroundings in new and unique ways. The use of these technologies in PoIs can give users a different perspective of their context by augmenting what is perceived in historical places, such as museums, monuments and archaeological sites [5].

Therefore, it becomes exciting to envisioning new, open scenarios with mostly intermittent users, in which augmented reality is applied to an outdoor environment. This can create some challenging issues that must be addressed, for example how to detect and track a POI when the user camera is moving [4, 5].

1.3 Goals

The thesis aims to investigate and define innovative approaches for suggesting and presenting content related to different cultural heritage landmarks of a city, based on contextual information about the visitor. This will enable the possibility to select content and make decisions about how to present it using the smartphone augmented reality capabilities.

In this way, the visitor will have all the digital contents at reach in a mobile application when walking through the streets of a city looking at monuments, cathedrals and other sites with cultural interest.

From built-in sensors and hardware of personal smartphones, such as light sensor, microphone, GPS, camera, it is possible to retrieve contextual information about user's behaviour. Besides that, these are crucial hardware components for the successful implementation of interactive augmented reality contents and for determining in which place of cultural interest is the user located.

Successful completion of the project will open the door to the development of innovative applications in the market in which it operates.

1.4 Structure of the Document

In addition to the Introduction Chapter, this dissertation is divided in four more chapters. Therefore, the document is organised as follows:

- Chapter 2 presents the results that were obtained when carrying out a literature review of the thematic under study, which aims to improve the experiences in Cultural Heritage sites.

- Chapter 3 analyses the application interface and all the ways that the user can interact with the App. Consequently, all the implemented functionalities will be discussed.

- Chapter 4 describes relevant information about the implementation of the *NearHeritage* application, such as the logic of the architecture employed, the process of integrating various API and SDK, the method to create the augmented reality content, among other functionalities. Also, it presents results in two different scenarios.
- Chapter 5 analyses the conclusions for this dissertation, the main difficulties felt, and some future implementations to be done.

Chapter 2

Literature Review

2.1 Introduction

This Chapter presents the results that were obtained when carrying out a literature review of the thematic and the different aspects of the problem under study. In this regard, it presents concepts and notions that are the baseline for the identification, selection and integration of approaches and technologies as a way to improve the experiences in Cultural Heritage sites.

Firstly, in Sections 2.2 and 2.3, the basic principles of context and personalisation are discussed, to comprehend that the use of context that surrounds the user is the key to individualise and to enhance the overall experience.

Then, in Section 2.4, the use of information acquired from mobile sensors combined with deep learning technologies is analysed, in order to use the visitor's location to identify places of interest around them, such as monuments and historical buildings. Furthermore, in Section 2.5, it is illustrated how the inclusion of immersive technologies, especially augmented reality during the experience, can improve the acquirement of knowledge and engagement of the visitors.

Lastly, in Section 2.6, through case studies of implemented Mobile Augmented Reality systems, it is demonstrated how adaptive technologies can be effectively complemented with an interactive design, rich narratives, and visitor's choice to create deeply personal experiences.

2.2 Context

User modelling has a vital role in supporting an individual user in order to improve the overall experience, learning methods, and information access. In the creation of context-aware applications, contextualised and personalised information and services are the driving entities.

Throughout literature, context information has been used to transform an application in a context-aware system capable of being modelled by user's behaviours and actions, as well as to manage data and knowledge.

This section explores the definition of context and how to categorise it, in order to provide a better understanding of how it can be used.

2.2.1 Definition and Importance

In the literature, when the definition of context began to be discussed, types of information were identified that fell under that designation. In this way, many definitions and taxonomies were created to classify that contextual information. However, these definitions did not provide a clear way to classify a new type of context information [6].

Considering this, many researchers have proposed several different meanings for context, resulting in fragmented context classification schemes. They have classified from three different viewpoints, namely the definition by example, by categories and in terms of specific context synonyms. For instance, many authors determine context by example and enumerate context elements like location, temperature, noise, as well as the beliefs, desires and commitments of the person. All these interpretations are indirect and suffer from generality and incompleteness [7, 8].

To address the limited notions and early definitions of context, Dey [6] provided a general explanation, which is probably the most widely accepted. In which context is referred to as any information that can be used to characterise the situation of an entity (i.e., a person, place, or object). An entity is considered relevant to the interaction between the user and the application, including the user and the applications themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects [6].

Context can be essential to a system or an application because usually, the user's objective is difficult to determine directly, and context can help infer this information and inform the application on how to best support the user [6].

So, a system is context-aware if it uses context to provide relevant information and services to the user, where relevancy depends on what task the user has [7].

Context-awareness represents a generalized model of input, including both implicit and explicit input, allowing almost any application to be considered more or less context-aware on how it reacts to the input [6].

Mainly, the notion of context includes any information that is important to the interaction between a user and an application, creating a context model that rapidly become large and complex, causing difficulties in complying with demands on the comprehensibility and manageability. So, the next section introduces a formal structure of context, which clusters the context information into categories [8].

2.2.2 Essential Categories of Context

The structuring of context is vital for any approach since it facilitates the engineering of a context model for context-aware applications. The separation in categories determines the design space of context models [8].

In the beginning, many authors accommodate context classification in three categories: human factors, physical environment, and time. The human factors included users, social environment, and tasks, while the physical environment comprised conditions, infrastructure, and location. Alternatively, the enumeration of context categories can consist of physical, system, infrastructure, location, and application context. Similar classifications appear in literature, distinguishing between physical, system, infrastructure, and domain context, as well as between physical, internal, and social context [9].

However, these descriptions lack an approach of how the model would capture a setting with many entities interacting with one another, each connected to its context [8].

One of the most accepted classifications, as shown in Figure 2.1, defines that any information describing an entity's context belongs to one of five categories: individuality, activity, location, time, and relations [8].

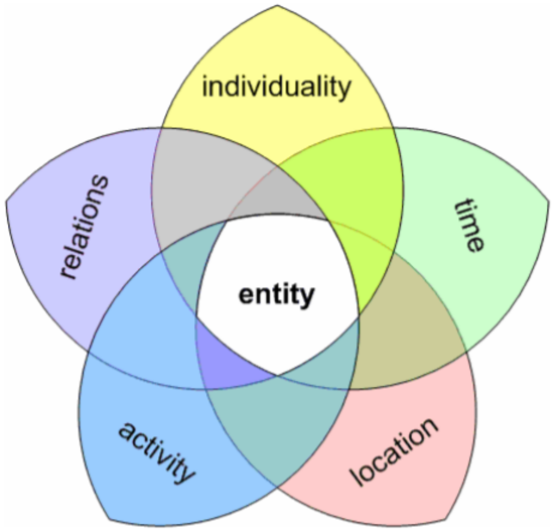


Figure 2.1: Categories of Context [8]

The individuality category contains characteristics and features describing the entity itself. The entity can be an individual or can belong to a group that shares common aspects of the context. Entities can act differently within a context-aware system. They can be active, allowing them to manipulate other entities, or passive. Also, entities can be classified as real, which means that they exist in the real world or virtual, and only exist in information space [8].

The category activity includes all tasks that the entity can be involved in, determining its current wants and needs. This type of context can represent entity purposes, tasks, and actions. The context categories location and time provide the spatio-temporal coordinates of the respective entity [8].

Finally, the relations category captures information about any possible association the entity may establish with another entity. The surrounding entities can be persons, things, devices, services, or information, for example, text, images, movies, sounds. All relations help to create a structure that is an essential part of the entity context. A connection represents a semantic dependency between two entities that appears from situations in which the entities are involved. Additionally, relations are not necessarily static and may develop and disappear dynamically [8].

The resulting understanding of context, places each entity in the centre of a surrounding individual context, supporting the necessity to use context in a context-aware application [8].

2.3 Personalisation of Cultural Heritage Visits

Cultural heritage has been a privileged application for personalisation in the research field, principally for individual support. Since every user differs and their experience is composed of many factors: the physical, the personal, the socio-cultural context, and identity-related aspects. Therefore, they might benefit from individualised support that takes into account contextual and personal attributes [10].

Personalisation is also considered a key component of location-based systems. In order to achieve personalisation, the system builds a model representing the user's needs and preferences, named the user model [11].

In this sense, personalisation allows users to obtain information adapted to their needs, goals, knowledge, interests or other characteristics. User models deliver the main parameters for managing and adapting information to provide to the individual user, since their behaviour may not be constant. Therefore, contextualisation can complement personalisation so that environmental states and contextual information, such as the user location, can be taken into account [10, 12].

2.3.1 Types of Personalisation Systems

The necessity for personalisation in cultural heritage spaces was mainly applied to adapt content to the liking of a single user based on the type of information that they want to see and how they behave during the experience [13].

Personalisation is a wide-ranging term that comprises three types of systems - customisation, context-awareness, and adaptivity [13]:

- **Customisation**, provides the users with many options to set up the application to their liking.
- **Context-awareness**, is the ability of the system to identify the current state of the environment and act accordingly.
- **Adaptivity**, implies that the system maintains a dynamic model of the on-going interaction and dynamically modifies its behaviour to adjust to each scenario.

Context-awareness is a mobile paradigm that applies context to provide relevant information and services to the user, where its relevancy depends on the user's current task, such as user location, nearby people and other devices, and user activity. In this way, context-awareness allows developers to learn about what users are doing with the application, where and when, which is essential to improve the applications to be even more aware of the user context [14].

The outdoor experience of the "Narratives in the Trenches of WWI" in the Italian Alps, showcase the types of personalisation. This experience includes a set of Bluetooth-enabled loudspeakers encased in wooden lanterns placed in points of interest in the trenches. These lanterns are paired with an interactive belt that embodies an NFC reader and a set of 4 cards (NFC tagged). Each card tells a story of the war from a different perspective, empowering visitors with content choices [13].

The system personalised the experience according to users' location and proximity to the hotspots, tracking the users' movements and reacting accordingly. When a visitor wearing the belt enters the area of a loudspeaker, a loud sound attracts the visitor closer. The attraction sounds had the double effect of raising surprise and awareness. When hearing the sound coming from the point of interest nearby, visitors were observed changing direction and move towards the sound source. The automatic start of the narrative then induced people to stop and listen, while visually exploring the environment [13].

Therefore, location awareness can significantly improve the user experience when it is paired with a careful interaction design sensible to the context and the surrounded environment [13].

2.3.2 User Profiles

User profiles have an essential role in recommendation methods since their models represent the user's characteristics, needs and preferences. Most personalisation systems need to build a user model that is usually represented in the form of a user profile in order to identify the needs of individual users and to understand their behaviour. Therefore, acquiring information about users is a fundamental task for making personalised recommendations [15].

The first stage to generate a user profile is to learn about user interests and preferences, that can be learned from their past/present interactions with the systems. These user interactions consist of either explicit or implicit information about the user's interest in items [15].

The explicit information can be obtained through approaches that rely on manual input from users. This information is obtained via registration forms or questionnaires or by rating modules to capture user's interests and preferences. Explicit information includes [15]:

- Demographic information: gender, educational background, age, nationality, location, and occupation.
- Data about interests and preferences: topics of interest, tastes.
- Opinion-based information: reviews, comments, and feedback.

For example, in [16], the system suggests points of interest after capturing the user's interests through a small questionnaire. These interests are applied to predict a personal interest score for each point of interest [17].

On the other side, the implicit information is based on user behaviour and by analysing their interaction with the system [15]. For instance, the application implemented in [18] dynamically develops the user's profiles according to their past behaviour, and it recommends places visited by users with similar profiles [17].

2.3.3 Recommender Systems

Recommender systems provide custom suggestions based on filtering techniques on many different domains, topics and items. These systems recommend items that are predicted to better match user preferences, thereby reducing the user's cognitive and information overload [19, 20].

Generally, when this type of system knows the contextual information that describes the situation the user is in, they can be more effective and compelling. Therefore, in recent years, traditional recommender systems started to take into account many dimensions as relevant aspects of the users' information needs. For this reason, context-aware recommender systems started to gain even more attention within the research community [19].

Nowadays, there exists an increasing amount of georeferenced data and users are typically interested only in local or nearby contexts and resources. The addition of the spatial dimension and, specifically, local geographic information in recommendation systems provides more effective and useful suggestions, which has led to a new application field within CARS called Location-Aware Recommender Systems [19, 21].

2.3.3.1 Traditional Recommender Systems Approaches

Traditional recommender systems are classified into three basic types, depending on the strategy followed to suggest recommendations: collaborative filtering, content-based and hybrid [17].

Collaborative filtering recommender systems evaluate the similarity between users profiles in order to suggest unseen items to a user, so users are correlated with each other. These type of system operate better when many users have similar preferences, actions or feedbacks. Although it needs several ratings from users before the system may begin to make recommendations, which is known as the cold-start problem, another problem is when the number of items to rate far exceeds what a user can rate, which refers to the sparsity problem [17, 22].

Content-based recommender systems analyse item features (contents) and descriptions to recognise items that are similar to those the user prefers. These type of systems perform well when there are enough features for items. So, to perform this type of recommendation, it is necessary to build a user profile that stores the degree of interest in each of the different criteria that describe an activity. However, this type of system suffers from the cold-start problem, and it also has the problem of over-specialisation, which means the user is limited to get recommendations that are very similar to the items he already knows [17, 22].

More recently, both recommender systems types were combined into hybrid recommender systems that exploit the advantages of one system to compensate for the shortcomings of the other, thereby improving the overall performance [20].

Although advances have been succeeded by using, fine-tuning, and extending traditional recommendation techniques, they can disappoint when estimating the relevance of an item in specific situations, for example, when the users are interested only in nearby items. In particular, these systems have problems when tackling scenarios with dynamic variables, such as the location of the user, time or weather [21].

Therefore, these approaches neglect the fact that the user's preferences for items in a certain context may be different from those in another context [23].

2.3.3.2 Paradigms for Incorporating Context in Recommender Systems

Traditional recommender systems are implemented based on partial information of the user preferences, and the input data is typically based on the records of the format <user, item, rating>. In contrast, context-aware recommender systems are developed based on the information of partial contextual user preferences, and typically the data records have the format of <user, item, context, rating>, where each specific record includes not only the preferences of a given user about the items but also the contextual information in which this user is enjoying the items. Furthermore, the information about the current context can be applied in different stages of the recommendation process, leading to several different models to incorporate contextual information [23]:

- **Contextual prefiltering** approach uses contextual information as an input to the recommendation process. So, the current context is used to select only the relevant data set before implementing a traditional recommendation method.

- **Contextual postfiltering** method applies contextual information to the recommendation output. Therefore, recommendations are created using the entire data set. Then, using contextual information, the resulting set is adapted.

- **Contextual modelling** directly applies contextual information to the recommendation function as part of the rating estimation. This paradigm does not use traditional approaches and employs multidimensional algorithms.

2.3.3.3 Location-Aware Recommender Systems

In order to mitigate the problem of traditional RS mentioned in the previous sections, considerable efforts have been invested in the last years to create the research field of Context-Aware Recommender Systems. These novel methods take into consideration the necessity for including the context of the user and items in the process of estimating accurate recommendations. The representation of context in a recommendation process can consider different aspects. For instance, the location of users and items has been proved to be particularly valuable to suggest relevant recommendations [21].

For this reason, Location-Aware Recommender Systems were created and take into account the spatial properties of users and items to provide recommendations. The emergence of these systems was enabled by the continuous evolution of mobile computing technologies. Since the mobile devices typically obtain the locations of moving objects through exploiting geographical positioning systems (GPS). Besides that, the evolution of LARS arises from the fact that users typically prefer nearby items, as the effort to reach items close to their physical positions will be smaller [21].

One of the most common application domains of LARS is suggesting points of interest (e.g., monuments) around the user. For this purpose to be achieved, it is necessary to analyse the mobile sensors and the POIs, with particular attention in the moving direction of the user, position, and speed, obtained through the GPS receiver. In this way, it is possible to recommend to the mobile user items of interest that are in a region near the user's current position and in the same direction [21]. With this objective in mind, the next section analyses how to identify points of interest.

2.4 Identifying Historical Places

Mobile technologies combine different tools and enable users to access multiple media simultaneously and from any location, combining several actions and functions in a single device [24].

With accelerometers, compasses, GPS and other embedded sensors in smartphones, it is possible to determine users' locations and the direction they are facing, therefore allowing an application to guide them to significant points based on their position [24].

In this way, the users can access information online and offline as well as consulting a map to follow a route or identify points of interest. Everyone can potentially access historical/cultural information and view and share cultural heritage data in real-time, furthering the possibility to make a social contribution by creating and sharing knowledge and experience with other users [24].

In fact, over the last fifty years, mobile technologies have enjoyed a significant evolution in the cultural heritage sector. It started with elemental content on standard devices and have grown to deliver customised content on several different platforms with personalised functionalities [24].

Some museums use pins to identify points of interest on a map and guide visitors to explore them as well as providing information about the location where artwork was created or places that influenced a given artist's work. Cultural routes can also be organised throughout a city, and media content can enhance the visiting experience, enriching a tour with multimedia and providing further information about a given subject [24].

For example, in Looking for Achille Castiglioni, a location-based mobile application was designed to lead users through eighteen PoIs in downtown Milan. Documents, drawings, photos, and models are accessible through the mobile device and tell the story of selected projects and temporary exhibitions that are linked to different locations that have been geo-tagged throughout downtown Milan. The application does not provide a specific route that users must follow, so they can decide where to go, consulting the map on their mobile screen. Each location activates a short video with images, photos, sketches and drawings as well as an audio description of the building/project, involving users in a tour that lasts about one hour [24].

In more recent years, the notion of Computer vision is achieving newer heights with the advancement in the field of Machine Learning, and specifically deep neural networks have made substantial progress in image recognition [25].

One of the applications for this technology is in Cultural heritage, for example, to ensure that the user is actually in a specific GPS location and facing the POI because one of the easiest way to obtain information about something is to use a picture of the object of interest as a query. Consider, for instance, a cultural tourist who is in front of a monument and wants to have information about it. A straightforward and intuitive action can be of pointing to the monument with a smartphone and obtaining pertinent and contextual information [26].

This section presents concepts and notions associated with the earlier content-based image classification techniques and with the advances made in deep learning techniques in that field, specifically for cultural heritage images.

2.4.1 Content-based Image Classification

The earlier landmark recognition systems used features representation and classification techniques, and later on context information to assist the recognitions [27].

Traditionally, visual features are heuristically designed and can be broadly categorised into global features and local features. The global features characterise the overall attributes of the images. The Local features intend to represent the contents of images by using local information obtained from the salient regions or patches within the images [27].

2.4.1.1 Global Features

The global features are the earliest and easiest feature utilised for landmark recognition because of its low computation cost and simplicity. Generally, the most used global features are colours, edges, and textures. These types of features are only able to describe the global statistical properties of the images and completely disregards the regions of interest. Therefore, it was normally applied in combination with local features for being able to provide a good landmark recognitio [27].

2.4.1.2 Local Features

The local features are applied in landmark recognition due to its ability to describe the properties of regions of interest. The process of finding and extraction the local features generally has two computational phases. In the first one, it is achieved the detection of interesting salient points. In the second phase, the descriptors are estimated in order to distinguish the appearance of the local regions surrounding the identified salient points [27].

The local features can be separated into three categories depending on how the local regions are extracted [27, 28, 29]:

- **SIFT** (Scale-invariant feature transform): this was initially developed by David Lowe and is a representation of the low-level image content, based on transforming the image data into scale-invariant coordinates relative to the features. The local features are usually low-level descriptions of keypoints in an image. The keypoints represents interest points that are invariant to scale and orientation, and each keypoint is associated with one or more orientations, based on local image gradients. The image matching is performed by comparing the description of the keypoints in the different images. Often, too many salient keypoints are extracted from images which cause the number of SIFT feature vectors to become too large.

- **Generative model-based**: this utilises probabilistic models to extract regions from the landmark images. The probabilistic descriptors that exist in this field are mainly focused on the probability distribution of salient regions in the images. These can describe more properties of the image than SIFT based features. Therefore, their information content is more reliable and robust to variations and occlusions. In this sense, generative models-based are more fitting for recognition, although SIFT-based features are more suitable for image matching.

- **Patches-based**: this have also achieved good performance, and there are two patches of extraction methods. The first one is to extract patches based on blocks, so it samples the image either uniformly or randomly to produce local patches. Then, visual features are extracted from these. The second method produces patches based on keypoints.

There also exists the SURF (Speeded Up Robust Features) method that is very similar to the SIFT. The SURF detects some keypoints in an image and describes these keypoints using orientation information. However, the SURF methods are faster than SIFT but still guarantees a good performance [28].

2.4.2 Deep Learning

Deep Learning is a state-of-the-art technology for classification and a branch of machine learning based on a set of algorithms that attempt to learn data representation with multiple levels of representation. Supervised learning is the most common form of deep learning when the model provides desired categories or labels for the result of classification [30].

In this sense, the field of computer vision has been pushed to a fast, scalable and end-to-end learning framework, which can provide outstanding performance results on object recognition, object detection, scene recognition, semantic segmentation, action recognition, object tracking and many other tasks that can be essential in the research field of Cultural Heritage. With the availability of large datasets such as ImageNet, Yahoo Flickr Creative Commons 100 Million datasets, among many others, it is possible to train the networks with a vast amount of correctly labelled images [31, 32].

In the case of image classification for tangible and immovable cultural heritage (such as monuments, historical buildings and archaeological sites), convolutional neural network and its variations can provide excellent performance, due to the attempt of CNN to imitate the image recognition system in the human visual cortex [30, 33].

2.4.2.1 Applications in Cultural Heritage

The interest in identifying monuments and historic buildings appears before the latest improvements in [34] and [26], which presents reviews of applications employing image classification and recognition techniques. Since 2012, convolutional neural networks began outperforming previously used algorithms in detecting and classifying objects from images. The introduction of CNN in heritage studies is proposed by [35], where the target objects are ancient inscriptions and a set of twelve monuments in Pisa (Italy) [36].

More recently, the authors in [32] used deep learning techniques to classify cultural heritage images, more particular to identify ten types of architectural structures such as columns, domes, gargoyles, vaults, among others. Two types of convolutional neural networks were applied, AlexNet and Inception V3, as well as two residual networks, ResNet and Inception-ResNet-v2. The dataset included more than 10,000 images, and the obtained results revealed that deep learning methods achieved better accuracy when dealing with complex problems such as image classification, compared to other cutting-edge techniques [37].

Another application is presented in [30] that uses convolutional and recurrent neural networks to classify Indonesian cultural heritage. In particular, CNN was used for image, audio, and video classification, while RNN was used to classify text. The algorithms were applied to the dataset containing 100 images, 100 audio files, 100 video files, and 100 text files divided into five categories each. The results revealed that, in terms of accuracy, RNN achieved the highest performance, classifying 92% of the text data accurately. In terms of CNN, the best accuracy was achieved for image and video classification (76% each), while audio classification obtained only 57% accuracy [37].

In [25], Inception convolutional neural network is applied to create an application able to identify twelve famous Indian monuments in images. The model was tested on twenty images, that were not a part of the training dataset, and obtained an accuracy of approximately 96% to 99%. Besides that, the model was also able to classify grainy images with high accuracy. So, this was a significant advance from other Machine Learning algorithms in previous models [36].

Due to the timeline of the dissertation and the complexity of employing neural networks, it will not be viable to research further the possibility of using neural networks in this work to identify cultural and historic points-of-interest.

2.5 Augmented Reality

Augmented reality as a concept has existed since the 1960s, although it was only in the last two decades that technological advances made it possible to formulate a distinct research field, which enabled applications in Cultural Heritages spaces [38, 39].

Augmented reality is a visualisation technique that overlay computer-generated data, such as text, video, graphics and other multimedia formats, on top of reality, as captured from the camera of a mobile phone or other devices. By combining the real world and the virtual world, AR aims at enhancing the user experience in a more appealing and innovative way [39, 40].

The current research trends regard AR improvements in the field of Cultural Heritage, include visits to historical landmarks and cultural heritage sites. These developments were capable of providing additional information about the visualised items while also being able to maximise the visitors' enjoyment and enhance their overall experience [40].

This section presents and analyses the concepts of immersive technologies, with a particular focus on augmented reality and its applications in Cultural Heritage.

2.5.1 Categorisation of Immersive Concepts

Immersive technology is a term that incorporates augmented reality, virtual reality, and mixed-reality, which enabled an entirely new way to experience multimedia content. These technologies are part of the *Reality-Virtuality Continuum*, represented in Figure 2.2, that defines the span between real and virtual environments, with augmented reality and augmented virtuality in between [38].

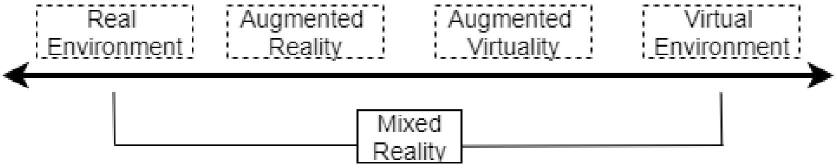


Figure 2.2: Reality-Virtuality Continuum [38]

In this context, AR can be defined by the superimposing of the virtual content over the real content, allowing the addition of virtual objects in the real world. One the other hand, virtual reality is the opposite because there is an overlay of the virtual over the real and the possibility to bring real objects into the virtual environment. Therefore, it presents synthetic content to the senses in a way that visual perception, hearing, and touch approach the experience of a real situation [38, 41].

The third approach, augmented virtuality, provides an augmentation of the virtual world with live scenes from the real world, so it combines real-world objects/scenarios into the virtual world [38].

Finally, mixed reality covers the continuum from AR to AV and aims at blending the real and virtual environments in different ways. It is a broad category covering various forms of AR and AV in a single technology. Therefore, in mixed reality technologies, real and virtual content coexists and interact in real-time [38].

In Table 2.1, it is presented a summary characterisation of all the concepts that belong to immersive technologies [38]:

Table 2.1: Immersive Concepts

Immersive Concept	**Description**
Augmented reality	Intends to improve users perception and understanding of the real world by overlaying virtual information on the view of the real world [38].
Augmented virtuality	Tries to augment the virtual world with scenes from the real world [38].
Virtual reality	Aims at intensifying users presence and interaction with a computer-generated environment without providing a way to interact with or see the real world [38].
Mixed reality	Implements a combination of the real and virtual environments [38].

2.5.2 Definition and Types of AR

As previous mentioned, augmented reality is a system that combines real and virtual content, implementing a real-time interactive environment, and operates in the 3D dimension. In this sense, AR is complementary to reality without entirely replacing it. However, AR studies performed in the past decades have modified the definition of AR to a system that improves users view of the real world by combining virtual and computer-generated information, helping the users to accomplish their tasks [2, 38].

In an augmented reality application, it is possible to find these characteristics [38]:

- It combines real-world and virtual objects.
- Allows the interaction between users and virtual objects.
- It functions in real-time.

The concept of augmented reality can be extended to applications that have the possibility to eliminate objects from reality using graphic overlays, and some authors classified it as Mediated Reality. Both types aim to improve users understanding and interaction with the real environment by combining virtual information and implementing intuitive interaction metaphors. Despite that, AR complements the real-world view with virtual and presents an augmented view, whereas mediated reality overlays synthetic content to remove the real-world view or some part of it virtually [38].

Within the concept of AR, there is also Holographic Augmented Reality that incorporates a more in-depth immersive AR experience, since it implicates using a headset to have a 3D viewpoint. In this way, by using a 3D stereoscopic viewpoint, holographic AR creates an impression of real size depth. Moreover, it provides users with a more embodied experience, especially in museums, by allowing gestural interaction. The users can interact with the virtualisation of the artworks in reality through hand-based gestural interaction [42].

2.5.3 Mobile AR System

In the early developments of AR, the systems were implemented in displays based on heavy hardware and complicated software modules, which were head-mounted. Nowadays, the advances made in mobile technology, such as improvements in built-in cameras, sensors and computational resources, has led to an integrated platform, ideal for the development of AR experiences, often referred to as mobile AR (MAR). This concept was first conceived in the late 1990s, creating early AR systems for cultural and archaeological sites, mainly tested in indoor environments [43, 44].

However, the advantage of portability provided by mobile devices facilitated the expansion of the AR system to outdoor environments. The advances allowed the visualisation of superimposed 3D models of heritage sites over the video stream of the outside environment, visualising virtual points of interest overlaid on top of the phone's camera view, or textual notes of outdoor locations based on GPS measurements [1, 43].

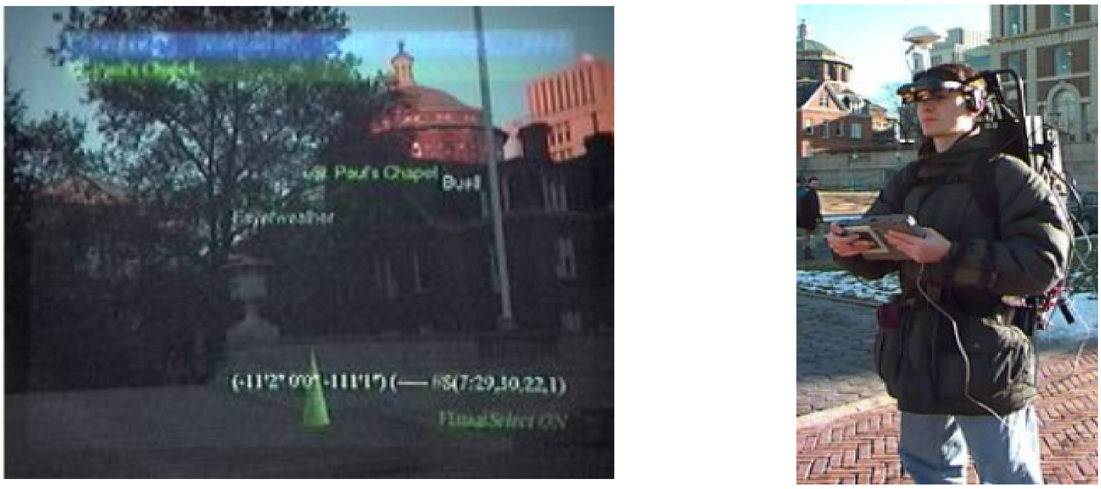
Mobile augmented reality system usually has the following characteristics [1, 44]:

- It combines real and virtual objects in a real environment.
- It is interactive in real-time.
- It registers and aligns real and virtual objects with each other.
- Runs and displays the augmented view on a mobile device.

For those characteristics to be accomplished, the system has to consider the various sensors of the device (camera, gyroscope, microphone, GPS), has to determine the type of information that is going to render in the screen of the mobile device, which may require access to local storage in the device or a remote database. Finally, the system has to project the output to the mobile device screen together with the current view of the user [44].

One of the first mobiles augmented reality system was the Touring Machine, represented in Figure 2.3. The users carrying the AR mobile unit while navigating freely on the Columbia University campus were presented with information about various points of interest. This mobile prototype, along with its numerous cables and hardware, had a total weight of 18 kg [41, 45].

The Section 2.6 presents and analyses more case studies of MAR systems inserted in the field of cultural heritage, since the first applications to the cutting-edge applications.



(a) (b)

Figure 2.3: (a) and (b) Touring Machine [45]

2.5.3.1 Key Concepts of MAR

The main features and key concepts in using AR through mobile devices are going to be presented, in order to provide a better understanding of what AR is, and how the user can interact with the real world and the digital information layer [2, 5]:

- **Reality View:** refers to the video image captured by the user smartphone camera in real-time, over which the augmented reality system will overlap digital virtual objects to create an enriched and augmented view of reality.
- **Registration and tracking:** represents a method for aligning virtual objects within a 3D coordinate in the reality view, using smartphones sensors, such as GPS, compass and accelerometer.
- **Point of Interest:** is a representation of an information point on a digital geographical AR map, which refers to a location in the real world.
- **Virtual object:** represents all digital content with augmented reality and superimposed it in the view captured by the camera. The content can be text, 2D and 3D images, 2D or 3D animations, movies, music, sound effects, interactive media and hypertext.
- **Markerless and Marker-based AR:** detects and identifies either artificial markers or natural features of objects in the real world.
- **Location-Based Tracking:** it is a means to obtain georeferenced information through the use of location sensors (compass, accelerometer and GPS- longitude, latitude, altitude).
- **Six Degrees of Freedom:** this concept is associated with the ability to maintain aligned a real-world object in a three-dimensional space. It refers to the six tracking actions that

any smartphone AR application must support: Forward/Back, Left/Right, Up/Down (GPS), Yaw (compass), Pitch and Roll (accelerometer).

2.5.3.2 Mobile AR Development Frameworks

The frameworks for the implementation of AR features in Android applications, present to developers a coding environment that has the tools to support the creation of innovative augmented reality contents [46]. A research was carried out in [47] about the available frameworks to implement augmented reality contents in mobile applications, and is presented in Table 2.2.

In the following, it presents the most popular AR platforms available on the market for Android device, in a concise manner and alphabetical order as they appear in the table [47]:

- **ARCore** [48] is Google's version for an AR platform that uses different APIs, allowing the device to be aware of its environment, comprehend the world and interact with information.

- **Vuforia** [49] supports the development of AR applications with advanced vision and recognition of a range of everyday images, objects and environments.

- **EasyAR Sense** [50] is a free augmented reality engine, which provides several cutting-edge features, such as SLAM, 3D object tracking, and screen recording, and simultaneous detection and tracking for multiple types of targets.

- **Wikitude** [51] is an AR SDK that combines instant tracking technology (SLAM), object recognition and tracking, state-of-the-art image recognition and tracking, location-based AR for mobile, tablets and smart glasses, 3D model rendering and video overlay.

Table 2.2: A comparison study between AR Frameworks [47]

Framework	**Last version found**	**Markers**	**Sensors**	**SLAM**	**Dynamic occlusion**	**License**
AR Core	29/08/2020	2D Tracking	GPS, IMU	Yes	No	Open source, free, commercial
EasyAR Sense	21/07/2020	2D Tracking, 3D object tracking	No	Yes	No	Free, commercial
Vuforia	09/09/2020	2D Tracking, 3D object tracking	IMU	Yes	No	Free, commercial
Wikitude	25/08/2020	2D Tracking, 3D object tracking	GPS, IMU	Yes	No	Free, commercial

The ARCore SDK was chosen to be used in the development of this dissertation due to its features and characteristics that are analysed in more depth in the Section 2.7.4.

2.5.3.3 Benefits of Implementing AR Systems in Cultural Heritage

In the early days, Virtual Reality was a prominent technology offering users an interactive, simulated environment. Although, its main disadvantage is that the user does not develop a connection with the real world and the surroundings, since it demands the full immersion of the user within the virtual environment. In contrast, Augmented Reality provides this connection since it is a superimposition of computer-generated data onto the real world. Due to this factor, AR increased popularity among individual users [39].

Particularly in the field of Cultural Heritage, the creation of mobile AR systems allows supporting the visitors in terms of obtaining relevant information and improving their knowledge regarding a monument, museum, archaeological site or other historical places. At the same, it enhances the overall user experience and offers increased levels of enjoyment throughout the process. These type of systems are possible due to the GPS functionalities of mobile devices that allow users to acquire benefits and navigate themselves to the selected cultural locations of interest with the help of direct annotations [39].

Furthermore, such systems are able to deliver personalise multimedia content according to the user's characteristics and their context, resulting in a new interactive and highly dynamic experience [39].

2.6 Case Studies

Throughout literature, the advances in mobile computing provided the ability to create new experiences that have enhanced the way to receive, interact with and present information about the world around the users. The emergence of components in the mobile devices such as integrated cameras, different types of sensors, and state-of-the-art processing power, has enabled outdoor AR experiences, in multiple research fields [41].

This section presents a brief overview of case studies that are related to mobile augmented reality in the field of Cultural Heritage.

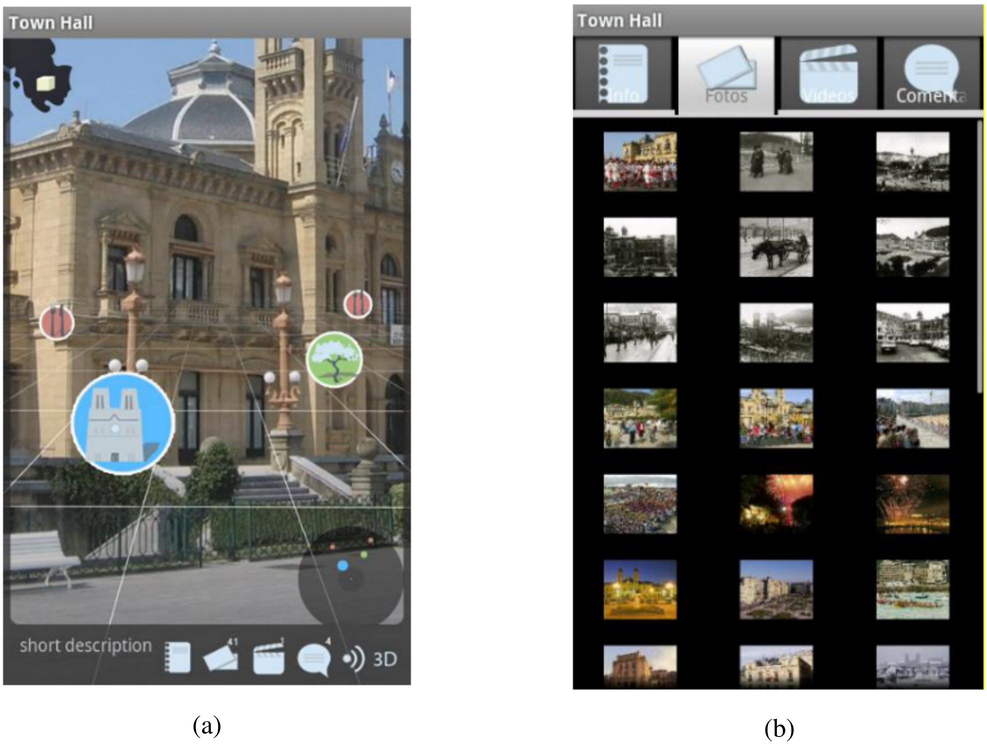
The *Archeoguide* project in [52], was one of the first personalises MAR systems for cultural heritage sites built around the Olympia in Greece, that remains an exemplar proof of concept for all later applications. The system includes a navigation interface, 3D models of ancient temples and statues, and avatars which are competing in a historic run held in the ancient stadium. The mobile device tracks the user's position on-site using GPS and is able to determine the viewing direction. In this way, the Archeoguide system can estimate the current view of the reconstructed objects. The rendered images are finally revealed in the display unit, generating an augmented view of the real world modified to the user's movements in real-time [45, 52].

The authors in [3] describe the *MobiAR* system, presented in Figure 2.4, that allows users to discover the surrounding areas just by pointing in any direction with the camera of the mobile device. In the display, the user is able to observe the video captured by the camera and the points of interest overlaid according to the distance and direction from the standing location of the user.

Each POI is associated with different types of multimedia content. When users select a POI by touching the screen, it triggers the immediate appearance of the available multimedia elements (description, comments, images and video) in the bottom bar.

The *Historical Tour Guide* [53] is another example of a mobile AR system demonstrated in Figure 2.5. The system experience occurred in a city of Norway, and it is a location-aware information system that uses mobile augmented reality to display local historical photographs about the evolution of points of interest over the years. There are three ways to access information in the application: clicking on a point of interest tapping a photo from the list of available photos and looking on the map. The main functionalities of the system are the augmented reality view where points of interest are displayed as floating icons overlaying the camera feed, the photo overlay that allows users to see historical images over the present-day scene and the map that shows the user's current position and the position of photos from the decade selected on the timeline.

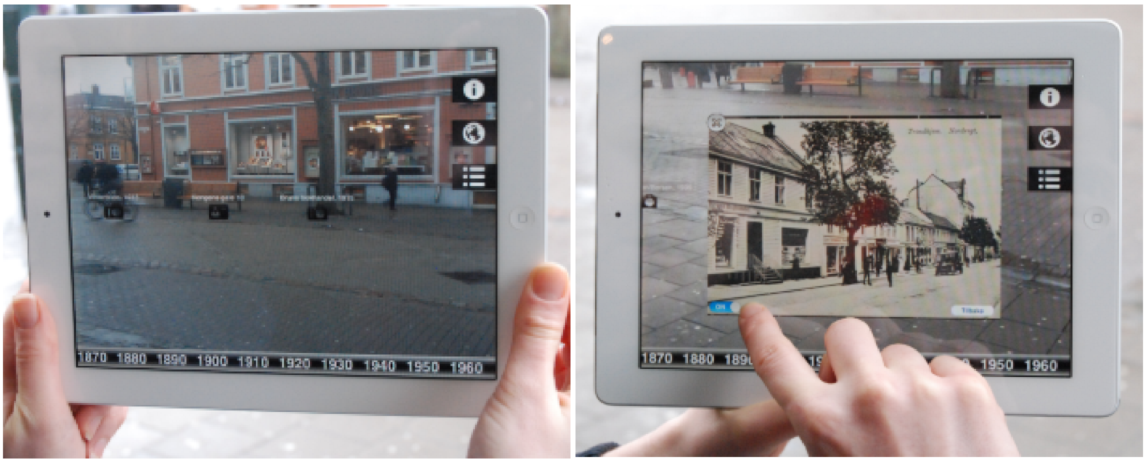
TowerAR [40] is also a MAR-based time-travelling system, presented in Figure 2.6, that displays historical information (3D models and images) about monuments in Italy, such as the leaning tower, the Cathedral, the Baptistry and the Sacred Cemetery. In order to augment the video stream from the camera, the user needs to point the mobile device towards the monument and then can select the information to be displayed on the target. The information can be display just in 2D on top of the video or the full 3D reconstruction of the building. The users were not only satisfied by the application but also felt excited when using it.



(a)

(b)

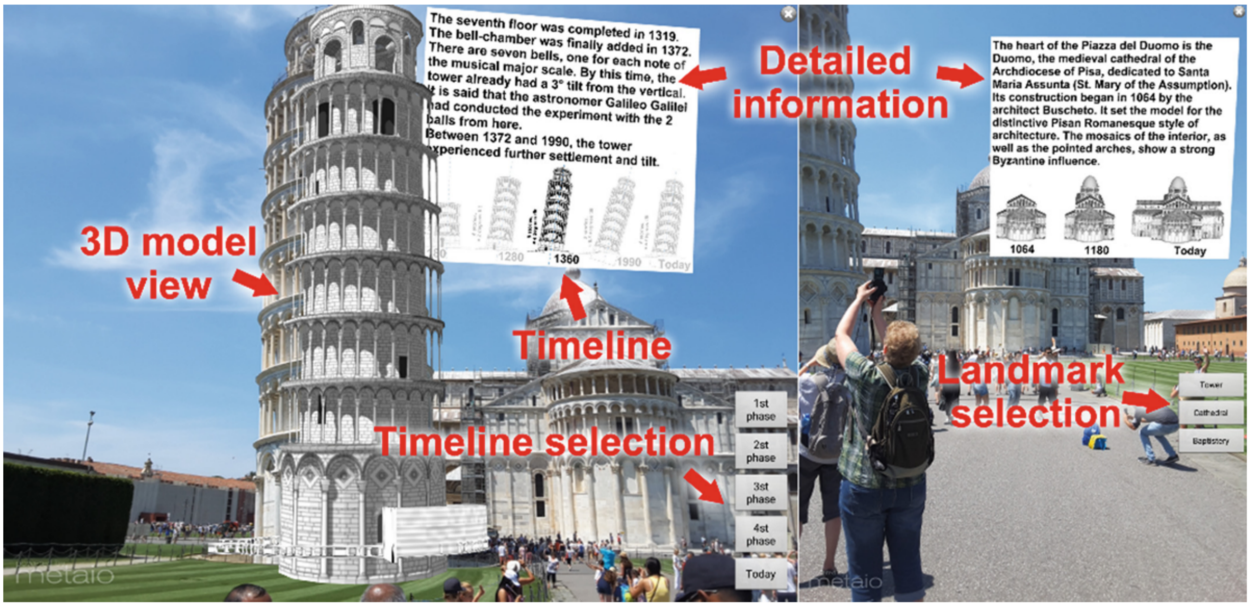
Figure 2.4: MobiAR - (a) AR view of the environment, and (b) Access to multimedia content related to that POI [3]



(a)

(b)

Figure 2.5: Historical Tour Guide - (a) AR view of the environment, and (b) Photo overlay [53]



The seventh floor was completed in 1319. The bell-chamber was finally added in 1372. There are seven bells, one for each note of the musical major scale. By this time, the tower already had a 3° tilt from the vertical. It is said that the astronomer Galileo Galilei had conducted the experiment with the 2 balls from here. Between 1372 and 1990, the tower experienced further settlement and tilt.

The heart of the Piazza del Duomo is the Duomo, the medieval cathedral of the Archdiocese of Pisa, dedicated to Santa Maria Assunta (St. Mary of the Assumption). Its construction began in 1064 by the architect Buscheto. It set the model for the distinctive Pisan Romanesque style of architecture. The mosaics of the interior, as well as the pointed arches, show a strong Byzantine influence.

Figure 2.6: TowerAR - Main Components [40]

Another example is the system described in [4], where a mobile augmented reality application was created to enhance the user's cultural experience during the sightseeing of a city. This application detects the POIs that the mobile device's camera's video feed is facing and produces an augmented scene with a virtual image slider. In this way, it provides users with additional digital content about the identified POI through a cloud computing service [40].

With the emergence of computer vision techniques, cutting-edge AR experiences were possible to create, such as the one presented in [54]. The proposed system is a novel wearable outdoor audio guide that adapts to the actions and interests of the user. The application can automatically collect points of interest by exploiting the user GPS position and the POI descriptions are obtained from the Internet. In order to determined accurately if the visitor is facing a point of interest, it was implemented a real-time computer vision system that consistently matches the image viewed from the user camera with an automatically obtained visual database of the surrounding. On persistent matches, the guide starts the audio description generated through text to speech technology.

The latest example of the use of augmented reality in a mobile application is the new feature provided by Google Maps named Live View [55]. This feature enables the use of augmented reality when the user is exploring a city on foot, taking in the sights and sounds of a new place. In this way, Live View allows the visualisation of AR objects, such as directions, arrows and distance markers, that are placed right on top of the real world, so the user can spend less time figuring out which way it should be headed. This is possible due to the combination of Street View's real-world imagery, machine learning and smartphone sensors, that improved the global localisation. For example, the Live View feature is able to take into consideration the elevation of a place, so it is more accurately displayed the location of the destination pin.

2.7 Development Platforms

This section analyses and studies several APIs and SDKs with functionalities and features that are required for the implementation and development of this dissertation.

2.7.1 Places API

All Places APIs such as Google Places API and Foursquare Places API can be incorporated into an application to take a geographic location or a string as input and return a list of nearby places. In this section, these two API are analysed and studied to assess the possibility of integrating this dissertation application.

2.7.1.1 Google Places API

The Places API [56] from Google is a service that returns information through a JSON or an XML response about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points-of-interest. The prominence of each place is affected by the ranking in Google index, global popularity, among other factors. Generally, the API returns the most prominent places first, unless specified that is based first on location, which returns more unique places, but of less quality.

The primary features of this API are place search, that is based on a location or a string, place details, place photos, place autocomplete, that automatically completes what the user is typing, query autocomplete, that predicts the query, returning suggested queries as the user is typing.

Places API is not free although it is possible to set up a billing account in order to get a fee credit (usable for Google Cloud Platform products, that includes Places API) and a monthly recurring fee credit (exclusive for Google Maps Platform products).

2.7.1.2 Foursquare Places API

The Places API [57] from Foursquare provides developers with real-time access to Foursquare's global database of venue data and user content to implement location-based experiences in Android applications. The Foursquare's database has over 105MM places available across 190 countries and 50 territories and 70+ venue attributes and 900+ categories, provided by the Foursquare consumer community.

The API support custom endpoints such as geo-tagging, venue search, venue recommendations, venue details (such as URL, ratings, tips, menus, among others), venues photos and many more. All the endpoints that return a list of venues, for example, venue recommendations or venue search, return venues that are near to a user current location. Therefore, the input to the API is the current location and the radius of the search for the venues. The results can be filtered by name or by a particular category, for example, in the monuments/landmarks and museums categories.

The API is a RESTful API based on HTTP requests and JSON responses, so all the requests need to be encrypted and sent via HTTP.

In brief, the Foursquare Places API provides location-based experiences with diverse information about venues, photos, and check-ins.

These functionalities of Foursquare Places API are all available for free, up to 1000 calls to the API. It is possible to set up a billing account, just for identity verification, that upgrades the calls to the API at 10 000 free calls.

2.7.2 Google Maps SDK

Google Maps SDK [58] for Android enables developers to apply maps based on Google map data in their applications. The API manages access to Google Maps servers, data downloading, map display, and response to map gestures. Besides that, it is possible to make API calls to add markers, polygons and overlays to a basic map, and to modify the user view of a certain map area. In this way, this API provides additional pieces of information for map location and user interaction with the map.

These functionalities of Maps API can be all used for free.

2.7.3 Fused Location Provider API

Fused location provider API [59] is a location API in Google play services that merges different signals to provide the mobile device location information in a simple and battery-efficient way. Therefore, the API handles location technologies, such as GPS and Wi-Fi, so that the developer can specify the required quality of service. Thus, it is possible to request the most accurate data or the best accuracy available with no additional power consumption. The following features are available:

- Last Known Location: returns last known location of the user's device.

- Location Update: can give location updates to a callback in the application at a particular interval and provide additional information such as directions and velocity.
- Location Settings: specify the priority for accuracy and power consumption, and the interval for updating the known location.

These features of Fused Location API can all be used for free.

2.7.4 ARCore SDK

ARCore [48] is a platform developed by Google for implementing augmented reality experiences in Android Applications. It provides higher-level APIs for the applications to interact with the real world. ARCore combines three key technologies for rich AR experiences since it tracks the position of the user into the three-dimensional space, estimates the light and detect flat surfaces or planes. Thus, it allows developers to focus on the application logic and creating beautiful AR experiences. The API uses the smartphone's camera and various sensors, such as the accelerometer and gyroscope, to track the user's position in the world and give this information to the application. As previously mentioned, ARCore operates on three fundamental principles:

- The **Motion Tracking** principle allows smartphones to understand its current position relative to the real world. In order to accomplish that, ARCore utilises the smartphone camera to identify features points and to track any change in their position as the camera moves. Based on the collected data, the API calculates the relative distance of the feature point from the position of the camera. Besides that, it also relies on the built-in smartphone sensors to detect the orientation of the smartphone.
- The principle of **Understanding the Environment** enables the smartphone to detect the size and location of all type of surfaces: vertical, horizontal and angled. This is possible due to ARCore continually searches for clusters of feature points that appear to be on a common surface. Also, the API detects the edges in the camera feed, identifying the boundaries of the surface.
- The **Light Estimation** providing the smartphone with the ability to sense the environment's lighting conditions, giving information about the average light intensity and colour correction. Developers can utilise this information to provide shadow and shine to their object in order to increase the realism of the scene.

So, as the user moves the smartphone in the real world, ARCore is able to understand its surroundings and emulate the real world digitally, in which it is possible to place the virtual objects in the detected surface/plane.

When Google first presented ARCore, it was a requirement that developers had knowledge in either *Unity* or 3D graphics with *OpenGL* to create AR applications. However, with the growth of ARCore, google created Sceneform, a new library, which enables developers to create AR applications using only knowledge of the Android SDK and the Kotlin/Java language. Sceneform has many handy features such as:

- An automatic compatibility check for ARCore enabled phones.
- Checking for camera permissions.
- A scene graph API to abstract all the complexities.
- A plugin for manipulating 3D assets.

These features and functionalities of ARCore can all be used for free.

2.7.5 Intel Context Sensing SDK

Intel Context Sensing SDK [60] is a library for Android that can enable context-aware abilities and services in applications. The SDK offers numerous methods to utilise the services, either independently or in combination. The SDK includes Context APIs that takes advantage of several built-in context type providers. The context states that can be sensed are Activity Recognition, Audio classification, Pedometer, Installed applications, Running applications, Battery, Calendar, Call, Contacts, Device information, Network, Device Position, Ear Touch Event Detection, Weather, among many others. Audio Classification was the context API that was the most studied in depth because it is going to be used in this work.

The Intel's "Audio Classification API" provides a semantic description for the surrounding audio. This API returns a weighted list based on the type of audio identified, from the most probable to the least probable. The audio can be classified in SPEECH, MUSIC, MECHANICAL, MOTION, and `CROWD_CHATTER` . The developer is able to specify the settings that configures the sensing behaviour. There are two settings: mode that is the number of samples collected before being communicated to the application, and continuousFlag that activates the application when the device is in "sleep" mode and a new context type state is received.

2.8 Concluding Remarks

The results of the literature review presented throughout this chapter display the importance of the acquisition and comprehension of context data, placing each visitor in the centre of a surrounding individual context which enables the creation of a meaningful and more in-depth experience. Especially since it is a step further in connecting the user-developer gap because it provides a fundamental understanding of the concept for users and helps the engineering of the concept for software developers [8].

Following, it was considered how the context allied with new opportunities for personalised cultural visits, improves, for example, the presentation of information and the exploration of engaging content for the specific visitor or group. Nonetheless, such evolution brings new challenges. For instance, the extensive available information and material, the increasing interest in supporting socialisation and the interest in user-generated content, associated with a need to ensure high-quality information at low costs, pose somehow conflicting requirements [10].

Mobile Augmented Reality systems are valuable for the Cultural Heritage field since it can help access information and improving visitors knowledge regarding points of cultural interest, for example, a monument or historical building, while enhancing the overall experience and offering increased levels of entertainment throughout the process [39]. These systems can be combined with neural networks in order to identify a point of interest through the camera feed rapidly, so it is an interesting field to be researched and implemented into cultural guide systems. However, in this dissertation, this area will not be covered.

Along with the opportunities that MAR systems offers, several case studies were analysed to understand what was already done and to evaluate the performance of the experiences developed with this type of system.

Chapter 3

User Interface and Application Interaction Design

The application interface of *NearHeritage* is presented throughout this chapter. In this way, all the screens are displayed, and the way the user can interact with them is discussed. Consequently, all the available functionalities that allow users to discover Cultural Heritage in a new city in a quick, easy and enjoyable way, is explained. All these functionalities were carefully thought out and planned by conducting a prior study of the possibilities in a hypothetical but realistic case.

Therefore this chapter is divided into three main sections. Section 3.1 demonstrates a hypothetical case for the use of the android application, followed by Section 3.2, where it described the use cases. In Section 3.3, all the screens and features are uncovered and analysed.

3.1 Hypothetical Case

This section has the purpose of illustrating a hypothetical case for the use of the android application *Near Heritage*, developed throughout this dissertation. Anna will be the fictional character of the storyboard, visiting a city for the first time, and is going to take advantage of all the main features of the app. Although, first it will be presented the visit to the city if Anna had not used the app, to understand the importance of having an application that assists visitors in the adventure of discovering the Cultural Heritage in a new city.

Anna is an enthusiastic tourist, who is going to visit Porto for the very first time and is eager to get to know the city and the culture. As she does not know anything about the city, Anna needs to do previous research on the internet, in order to know which monuments and historical places she should visit. In this way, she can organise and plan the visit and find out the paths to reach each cultural place.

Upon arriving in the city of Porto, Anna keeps getting lost in the narrow and peculiar streets of downtown, taking too long to get from one place to another. Besides that, Anna does not know which place to visit next, because she is not aware of which are the closest or farthest places from her current location. At the end of the visit, Anna is upset for the little amount of information that she has about each cultural landmark to show her friends and family. Also, Anna realises that in the monument *Torre dos Clérigos*, she did not even visit everything available because she was not informed that it is possible to not only go up to the tower but also visit the beautiful church.

If Anna had used an android application, specifically the *NearHeritage* application, her visit to Porto could have been much more exciting and enjoyable. This App has incredible tools and features that can provide information about nearby historical landmarks, museums or monuments, and help a tourist find them. So next, it will be described how the visit would have been like if Anna had the *NearHeritage* application to help her in this new adventure.

After downloading and opening the App and giving the required permissions (GPS and camera), Anna creates an account and can now visualise the home screen (NearMe screen) which has a list of the cultural heritage landmarks in the surrounding areas of her GPS location. Besides that, Anna can see information about what distance she is from each historical place, the addresses, among other things. As it is possible to see in Figure 3.1.

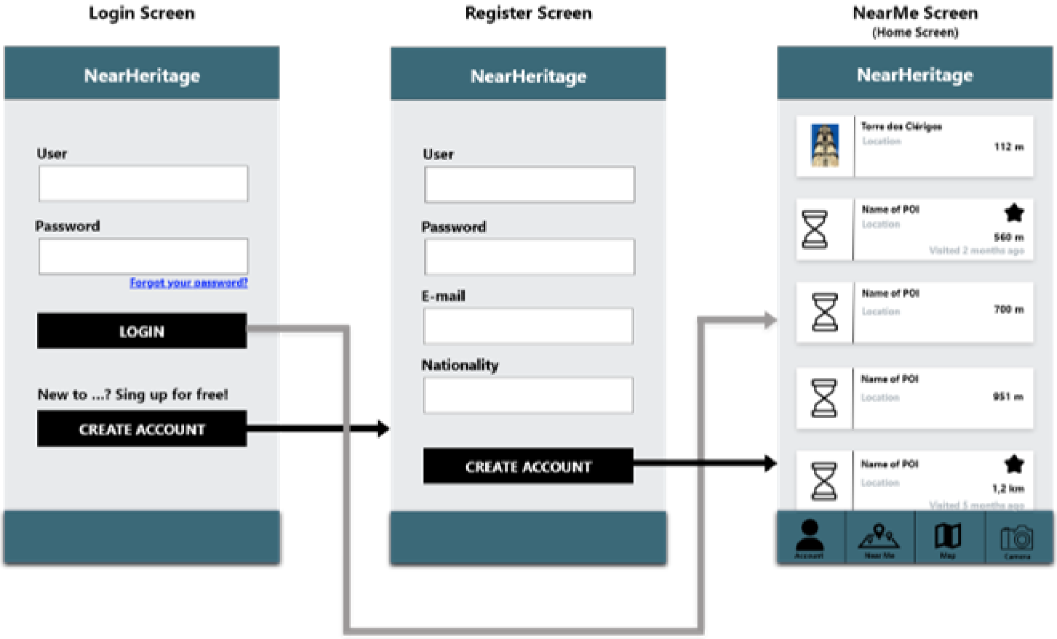


Figure 3.1: Initial interaction with the application

Then scrolling through the list, she decides to click on the landmark named *Torre dos Clérigos* and receives more specific details and additional pieces of information about scheduling, contacts, rating. As it is presented on the screens in Figure 3.2.

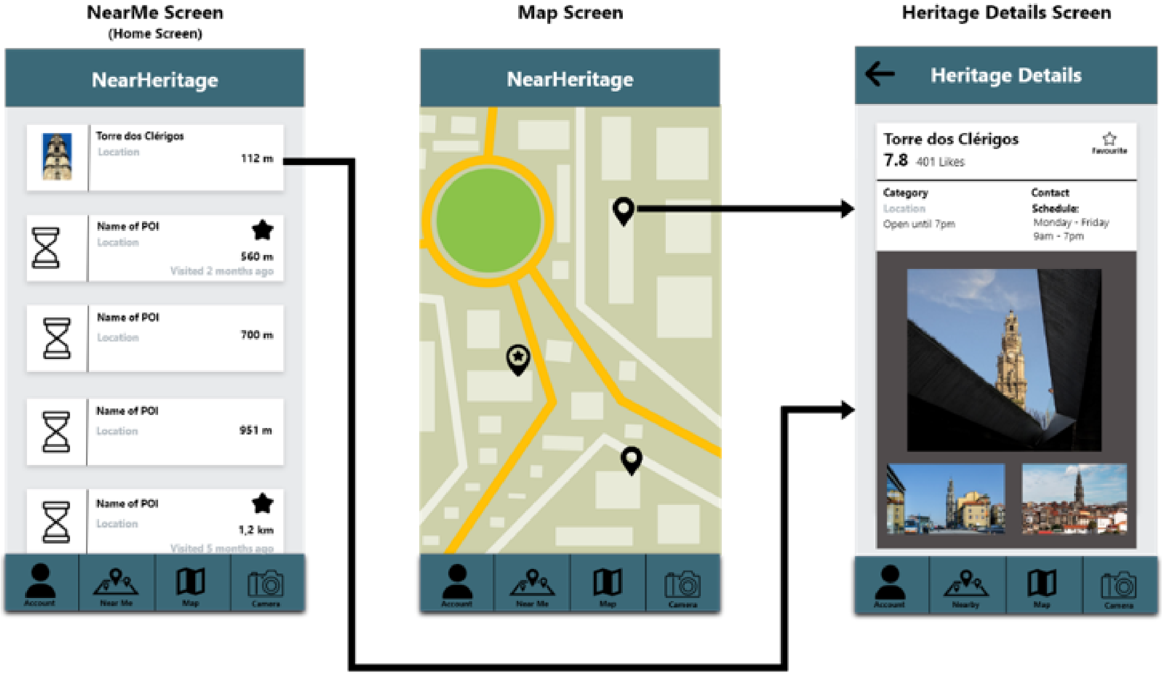


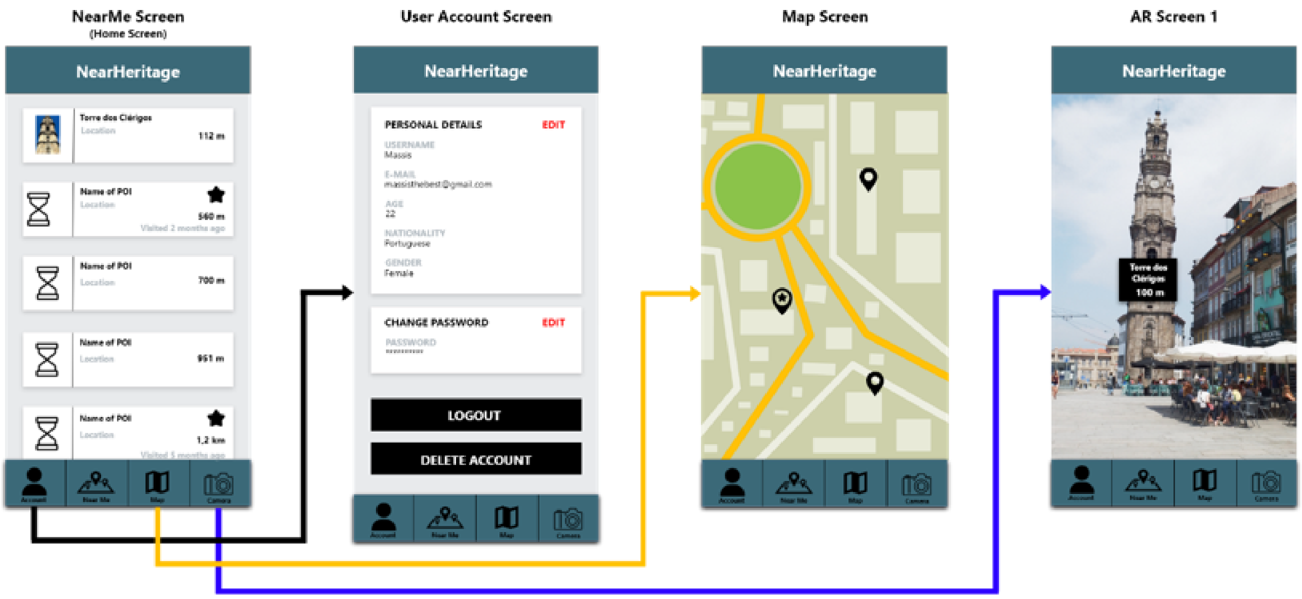
Figure 3.2: Actions to get to Heritage Details screen

Anna gets really interested in actually visiting that landmark but does not know how to get there. So, she decided to explore the App in order to see if one of the other features could help her.

In this way, she finds the account screen, the map screen and the AR screen, presented in Figure 3.3. On the map screen, Anna realises that she can follow her location in real-time and, at the same time, view the location of the monuments. While the AR screen made Anna very curious to understand what this feature is able to do.

When Anna arrives at *Torre dos Clérigos*, she receives a notification to go to the AR screen, demonstrated in Figure 3.4, where she is presented with augmented reality multimedia content. So, she is able to see a short video with cultural information about that specific monument. This is only possible because she is inserted in a perfect context in terms of noise and brightness. If there was a lot of surrounding noise, Anna would only receive visual content, such as images or 3D objects. If there was too much brightness due to very bright sunshine, then the App would only present to Anna, auditory information about the landmark.

The interaction with multimedia content through augmented reality made Anna intrigued by what she would find when she visited other historical places.



Torre dos Clérigos
Location
112 m

Name of POI
Location
560 m
Visited 2 months ago

Name of POI
Location
700 m

Name of POI
Location
951 m

Name of POI
Location
1,2 km
Visited 5 months ago

PERSONAL DETAILS
USERNAME
Massis
E-MAIL
massisthebest@gmail.com
AGE
22
NATIONALITY
Portuguese
GENDER
Female

CHANGE PASSWORD
PASSWORD
****

Figure 3.3: Destinations screens of the bottom bar

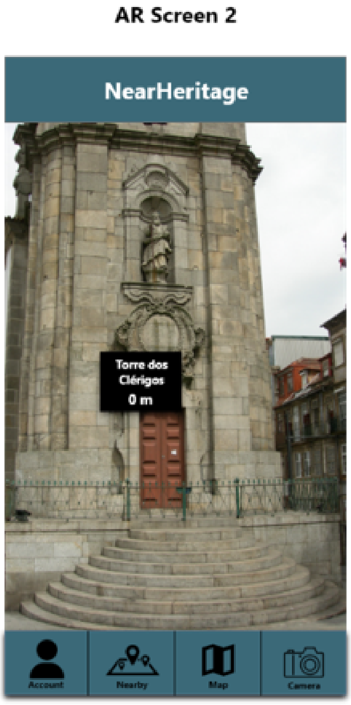


Figure 3.4: Tourist arriving at the landmark

3.2 Use Cases

The hypothetical case presented in the previous section is a starting point to analysing the main functionalities of the *NearHeritage* application, which are:

- User registration in the App.
- Showcasing a list of the nearby Cultural Heritage venues that can be, for example, monuments, landmarks or museums.
- Providing a map with the user location and the Cultural Heritage venues location.
- Presenting personal user information's provided when the user registered, such as username, e-mail, date of birth, age, nationality and gender.
- When the user arrives at the Cultural Heritage venue, the user can receive multimedia content about the history of the place through augmented reality techniques.

In addition to these functionalities, this application has many more features and tools to assist the user in his visit and improve the overall experience. All of them are displayed in Figure 3.5, that represents the use case diagram for the application. So, the additional features are:

- Visualisation of detailed information about each Cultural Heritage venue.
- Possibility of editing personal account information's.
- Capability of editing the account password.
- Logout of the account.
- Deleting the account.
- Power to decide enabling or disabling the camera tutorial.
- Taking pictures of the cultural places and throughout the augmented reality experience.
- Displaying all the photos taken in a photo gallery.
- Ability to share the photographs.
- Deleting photos from the photo gallery.

The next sections analyses the design associated with each one of the presented functionalities to provide an understanding of how they are connected. Additionally, it is explained how the navigation between the different screens is structured.

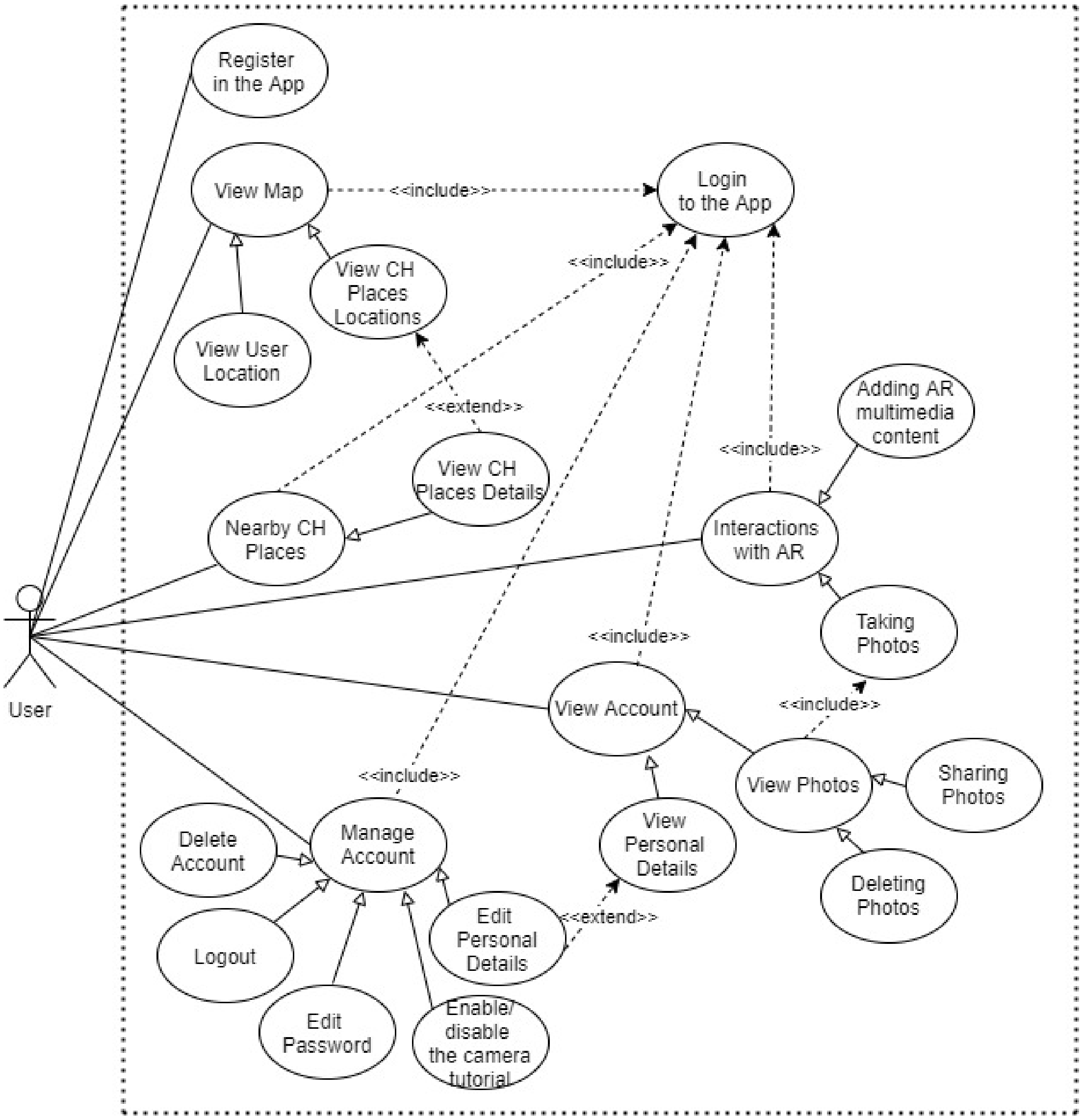


Figure 3.5: Use Case Diagram

3.3 Screens Design and Interaction

The design for the application *NearHeritage* has a structural base that is equal to all the screens, illustrated in Figure 3.6. This structure is divided into three fundamental parts:

- **Screen title** that helps to situate the user in the application.
- **Work Area**, which is the zone that varies the most across all the screens, depending on the purpose and functionalities that the screen has to offer to the users.
- **Bottom bar navigation** that allows access to the four primary screens of the application, which are the *Near Me* screen, *Map* screen, *Account* screen and *Camera* screen.

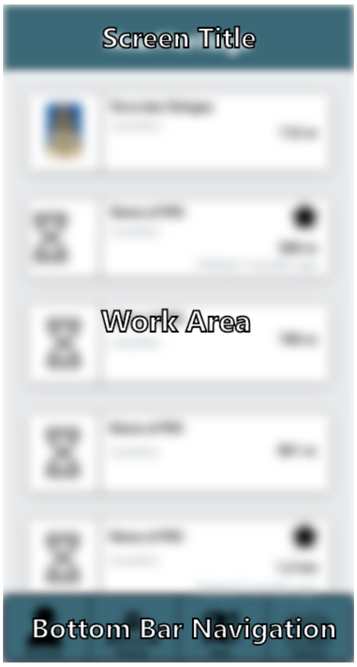


Figure 3.6: Overall application design structure

The four screens associated with the bottom bar navigation are top-level destinations in the navigation hierarchy. The login screen can also be considered a top-level destination, although the user typically only sees that screen one time. The rest of the screens are considered lower-level destinations. The Figure 3.7 illustrates the structured of the application's navigation hierarchy and thus, how the screens relate to each other.

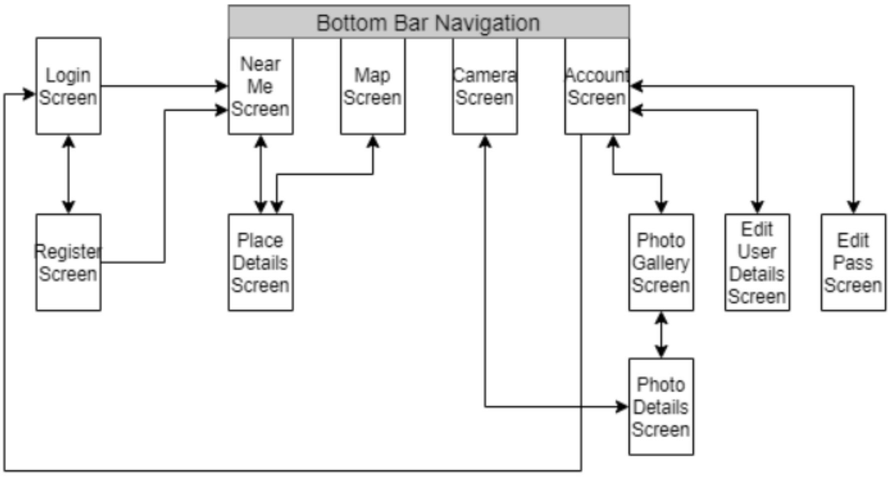


Figure 3.7: Application navigation diagram

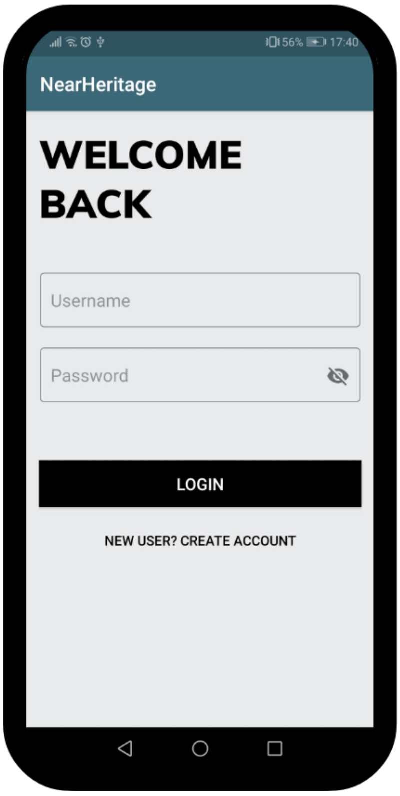
3.3.1 Login and Registration Screen

When the application is initialised, it will do a series of required verifications for the App to launch.

First, it will check if the user has already given permissions for the App to access the GPS, to write in external storage, to use the camera, to record audio and make phone calls. In case of these permissions have not yet been provided, the system will ask the user for them.

Second, the application must confirm the existence of a connection to the GPS and any source of internet, such as Wi-Fi, mobile data or VPN. This is necessary in order for all the application features to work correctly. Therefore, if the user does not have these connections activated, the App will request the user to turn them on.

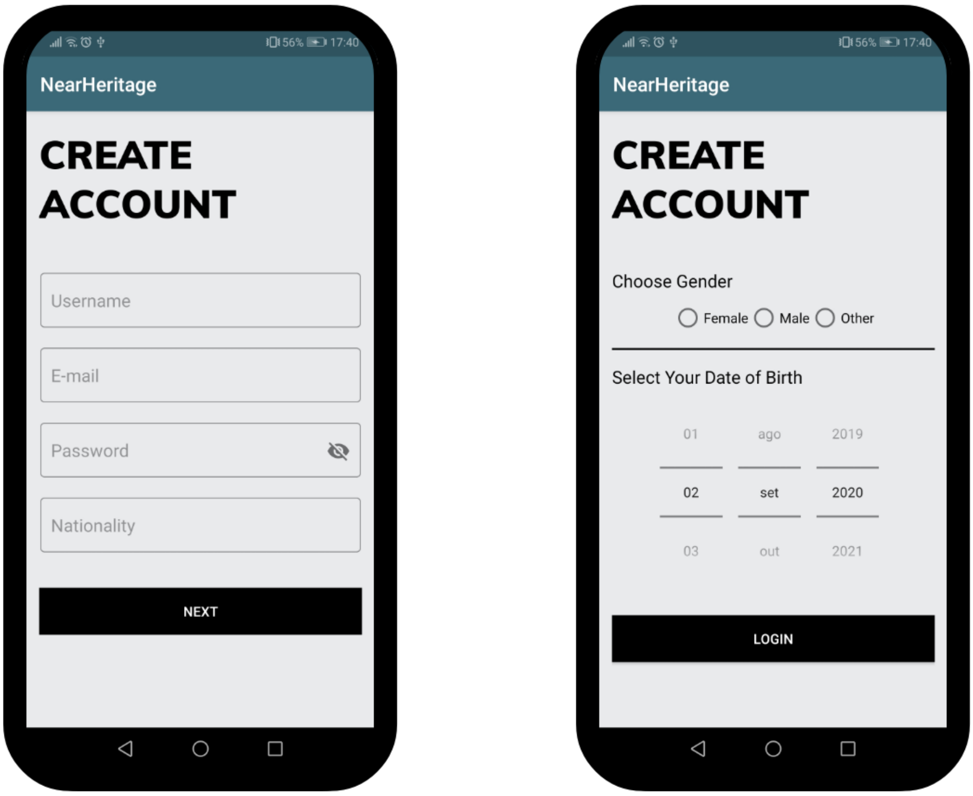
Lastly, the App will verify if there is a user logged in. In case it has, the user will see the Near Me screen opening, that will be analysed in Section 3.3.2. In case it has not, the App will redirect the user to the Login screen, presented in Figure 3.8.



WELCOME BACK

Figure 3.8: Design of the login screen

The login allows the App to identify the user and fetch their personal data to display in the Account screen, that will be presented in Section 3.3.5. The personal user data will be acquired during the registration process (see Figure 3.9), triggered by clicking the button "NEW USER? CREATE ACCOUNT" that is part of the login screen.



(a) (b)

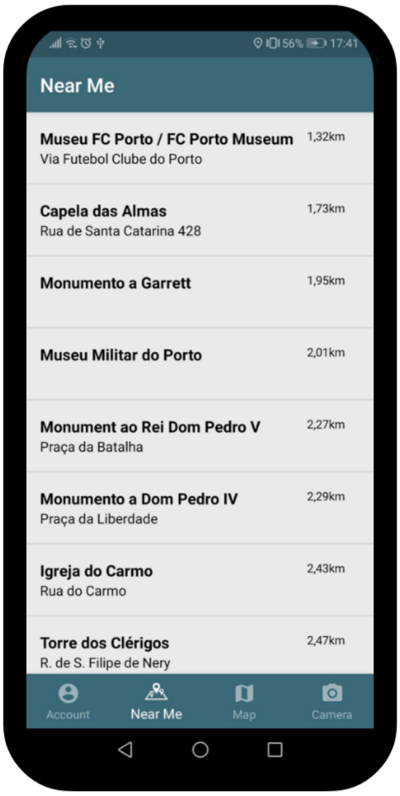
Figure 3.9: (a) and (b) Design of the registration screens

3.3.2 Near Me Screen

The Near Me screen is the home screen for this application, where the user can receive information about what are the nearby Cultural Heritage venues. These venues and the information displayed about them are fetched from the Foursquare API, in which the input is the current user location. When the user location changes, the nearby Cultural Heritage places are updated.

In Figure 3.10, it is possible to visualise a list of the fetched Cultural Heritage venues, along with information about the address and the distance to them from the user location. The venues are displayed in the list from the closest to the user's location to the farthest.

The user can trigger the application to display a new screen by clicking on one of the venues listed. This screen is named Venue Details and will be presented in the next section.



Near Me

Museu FC Porto / FC Porto Museum 1,32km
Via Futebol Clube do Porto

Capela das Almas 1,73km
Rua de Santa Catarina 428

Monumento a Garrett 1,95km

Museu Militar do Porto 2,01km

Monument ao Rei Dom Pedro V 2,27km
Praça da Batalha

Monumento a Dom Pedro IV 2,29km
Praça da Liberdade

Igreja do Carmo 2,43km
Rua do Carmo

Torre dos Clérigos 2,47km
R. de S. Filipe de Nery

Figure 3.10: Design of the near me screen

3.3.3 Venue Details Screen

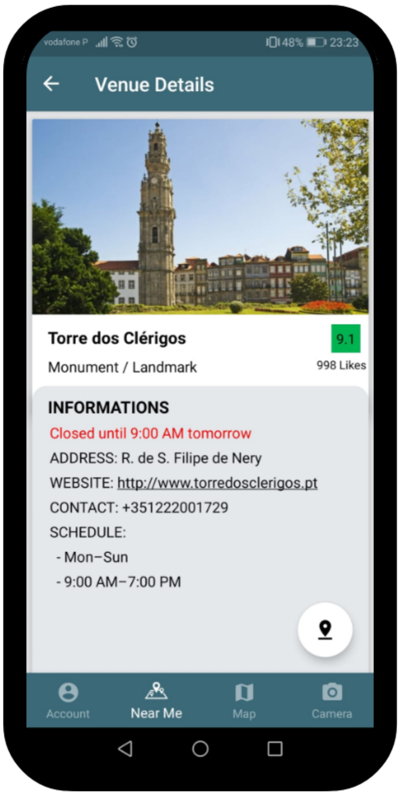
The Venue Details screen has the purpose of helping the user getting more knowledge about the venue and getting familiar with it. In this way, the user has access to important pieces of information that is retrieved from the Foursquare API, when the user selects a venue in the near me screen or the map screen. The information fields returned from the API can be:

- Name.
- Phone contact.
- Address.
- Categories, such as the museum, landmark/monument, church and outdoor sculpture.
- URL to the website of the venue.
- Rating and likes.
- Business Hours.

- Indication if at that time the venue is open or closed and until what hour.

As the information in the Foursquare API is built by the users and developers of the API, some fields may be not available and return null. In Figure 3.11, it is possible to visualise the implemented design of the Venue Details screen.

Besides the informational value of the screen, the user can interact with it in other ways. For example, the user can click on the phone number, and the App will automatically start a call to the venue, or if the user clicks on the URL, the App will open the website. Additionally, by clicking on the 📍 button, the App will redirect the user to the map screen and will change the position of the map's camera to venue location.



Torre dos Clérigos

9.1

Monument / Landmark

998 Likes

INFORMATIONS

Closed until 9:00 AM tomorrow

ADDRESS: R. de S. Filipe de Nery

WEBSITE: http://www.torredosclerigos.pt

CONTACT: +351222001729

SCHEDULE:

- Mon–Sun
- 9:00 AM–7:00 PM

Figure 3.11: Design of the venue details screen

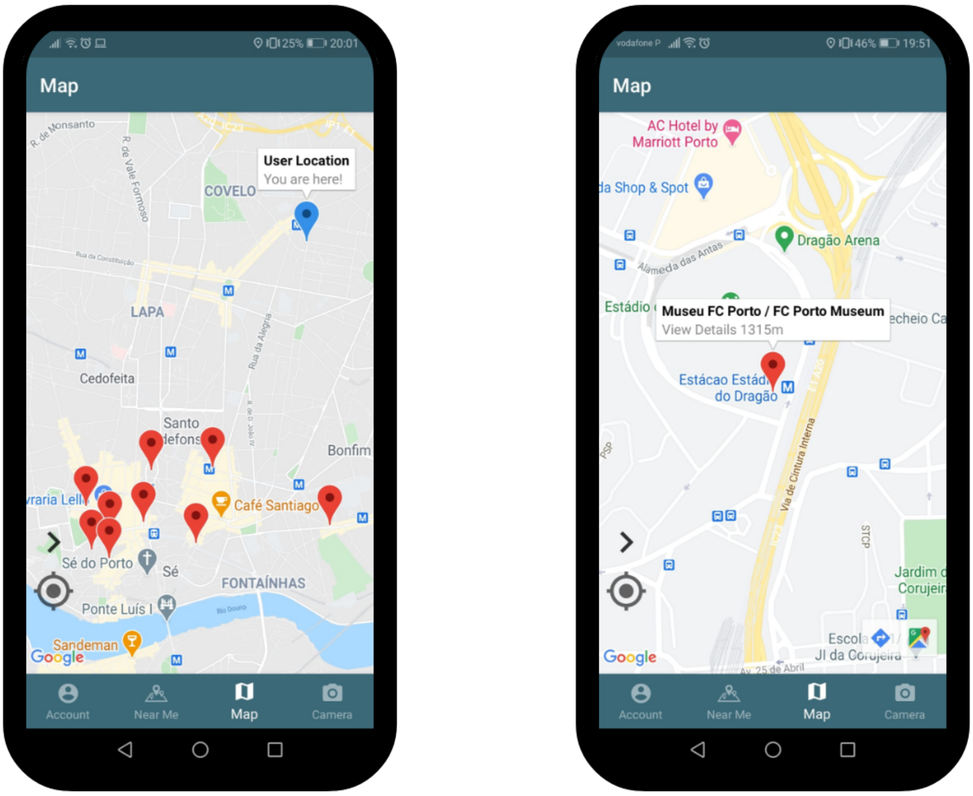
3.3.4 Map Screen

In this screen, users can visualise in a map, provided by Google Maps SDK, a marker for their current location and markers for the location of all the Cultural Heritage venues. In this way, users have a visual perception of how distant each venue is, which is possible to visualise in Figure 3.12a.

Besides that, the user has at his disposal various buttons to help him navigate in the map, such as:

- The ⊙ button will change the position of the map's camera to the user position.
- The › button will change the position of the map's camera to the next nearby venue, going from the closest to the user's location to the furthest.
- The ◈ button will automatically open Google Maps and create a route from the user location to the selected venue.
- The 🗺 button will start Google Maps with the location of the selected venue on the map.

Lastly, a snippet will appear when the user clicks on a marker of a venue, as it is illustrated in Figure 3.12b. The user can then tap on the snippet, and the App will present the venue details screen, described in the previous section.

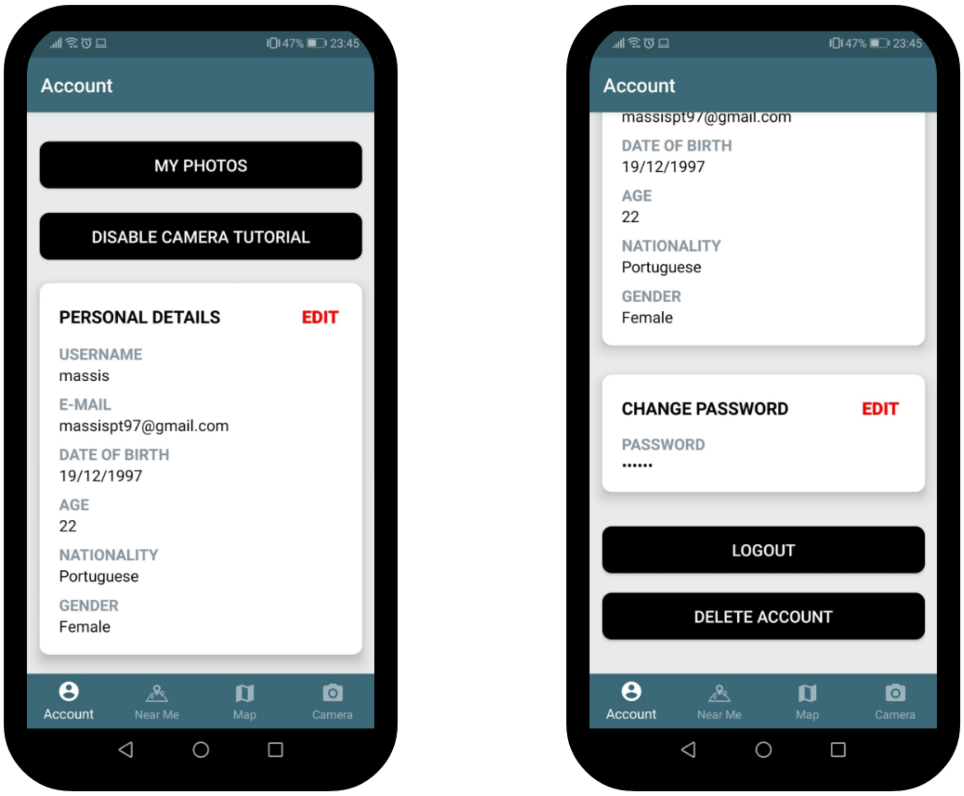


(a) (b)

Figure 3.12: Design of the map screen

3.3.5 Account Screen

The Account screen, as its name indicates, is where users can see their personal information, that the application acquired on the registration process (see Section 3.3.1). This screen provides users with the ability to make alterations to their account, whether it is updating their personal data or deleting their account.



Account

MY PHOTOS

DISABLE CAMERA TUTORIAL

PERSONAL DETAILS EDIT

USERNAME
massis

E-MAIL
massispt97@gmail.com

DATE OF BIRTH
19/12/1997

AGE
22

NATIONALITY
Portuguese

GENDER
Female

(a)

Account

massispt97@gmail.com

DATE OF BIRTH
19/12/1997

AGE
22

NATIONALITY
Portuguese

GENDER
Female

CHANGE PASSWORD EDIT

PASSWORD
••••••

LOGOUT

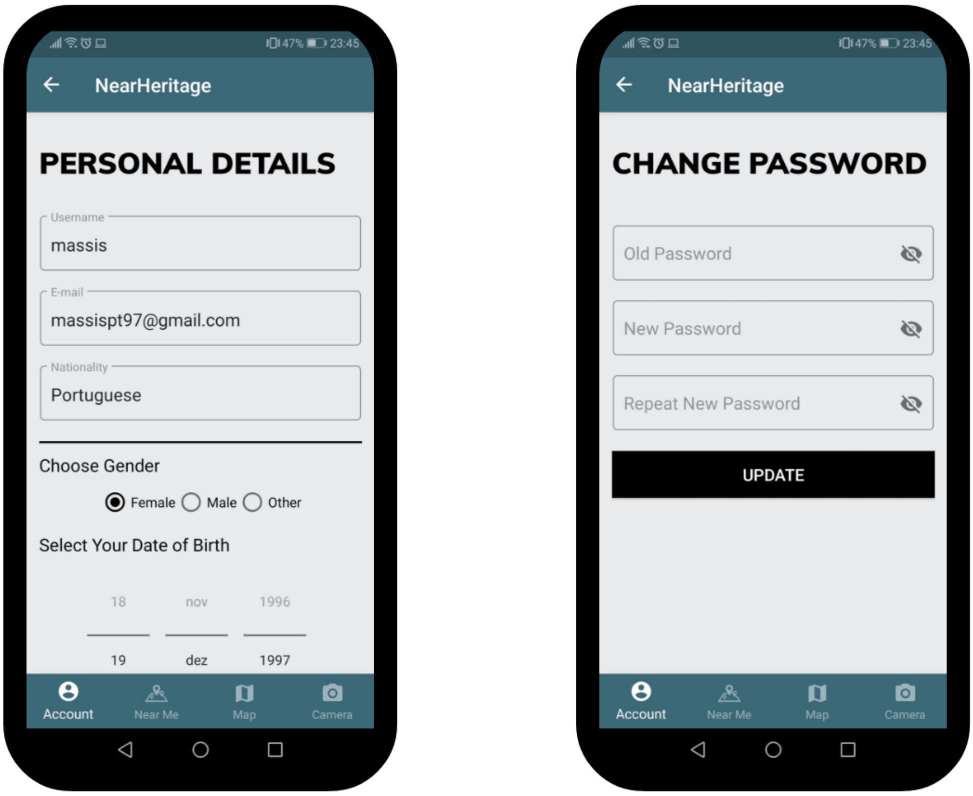
DELETE ACCOUNT

(b)

Figure 3.13: Design of the account screen

All the functionalities from the account screen (see Figure 3.13) that are in reach of the users are:

- The "My Photos" button that makes the App start the Photo Gallery screen, presented in the next section.
- The “Enable/Disable Camera Tutorial” button allows users to activate or deactivate the tutorial for using AR, that is going to be discusses in Section 3.3.7.
- The capability of updating personal information by clicking on the "Edit" present in the "Personal Details" container that will lead to the screen displayed in Figure 3.14a.
- The user can modify the account password by triggering the "Edit" displayed in the "Change Password" container that will lead to the screen presented in Figure 3.14b.
- The button to logout of the account.
- The button to delete the user account.



Username: massis

E-mail: massispt97@gmail.com

Nationality: Portuguese

Choose Gender

● Female ○ Male ○ Other

Select Your Date of Birth

18 nov 1996

19 dez 1997

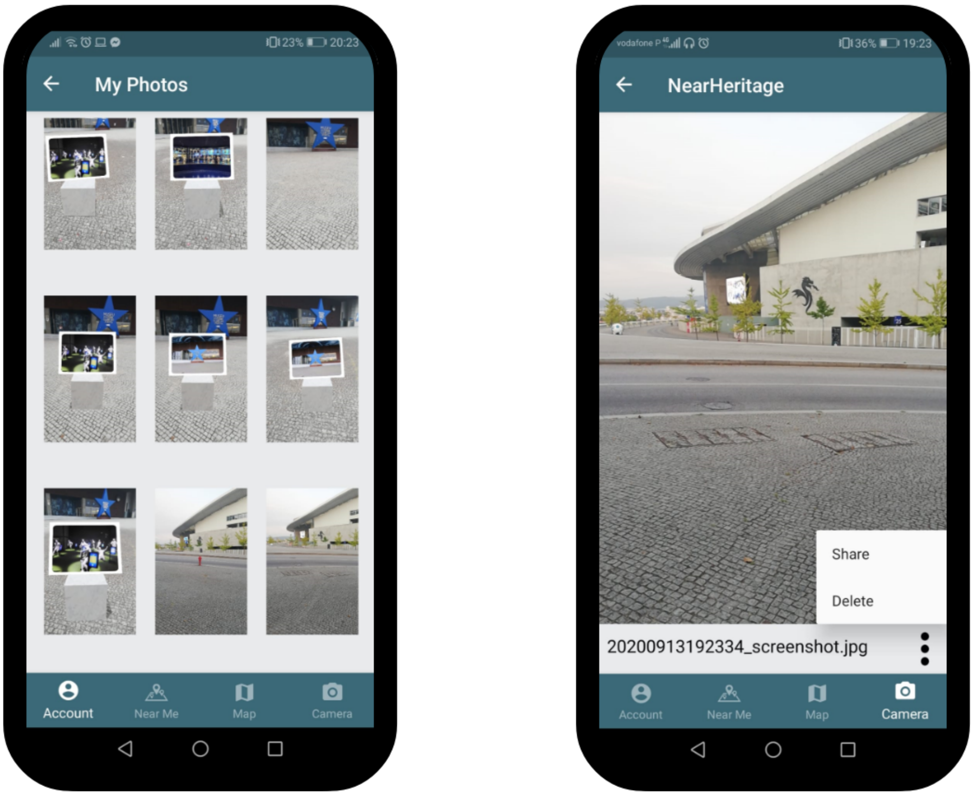
(a) (b)

Figure 3.14: (a) Design of the edit personal details screen and (b) design of the edit password screen

3.3.6 Photo Gallery Screen

This screen has the purpose of showing all the photos that the user took on the camera Screen (see Section 3.3.7), displayed in Figure 3.15a. Besides that, the user can tap on one of the photos, triggering the opening of a new screen, visible in Figure 3.15b. This screen is named Details Photo Screen, where a user can see the selected image in full screen, along with its name. Near the image name, it is possible to find the ⋮ button, in which the user has access to the following functionalities:

- Deleting the chosen photograph.
- Sharing the photo in the user social media or with family and friends.



(a) (b)

Figure 3.15: (a) Design of the photo gallery screen and (b) design of the details photo screen

3.3.7 Camera Screen

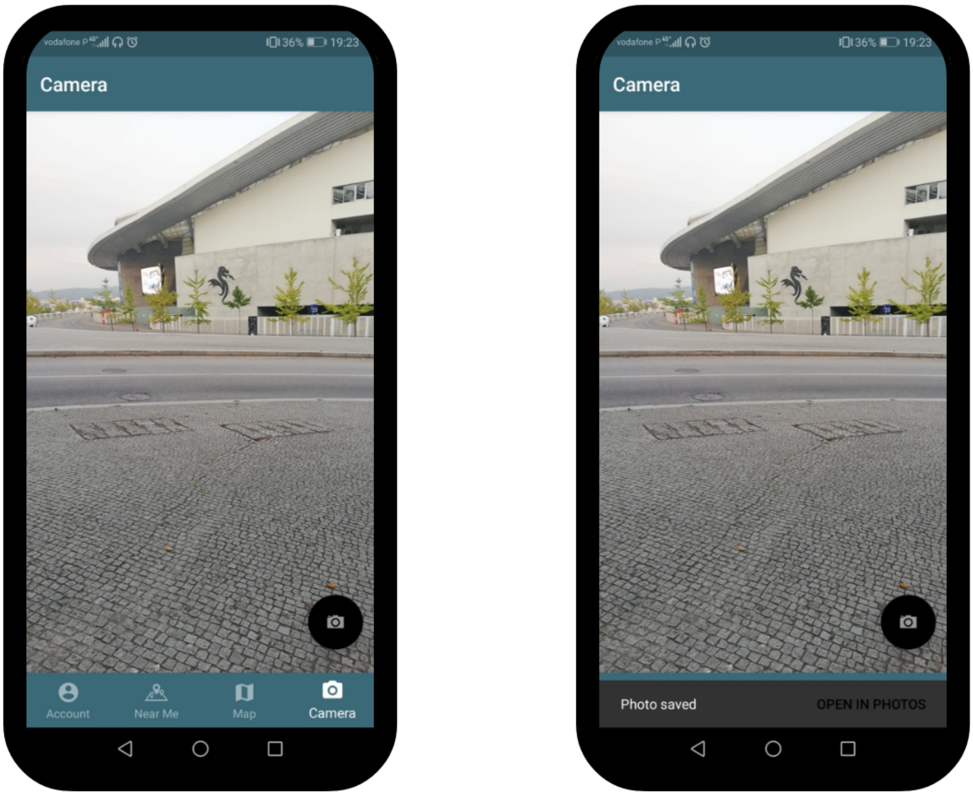
In this screen, the user will be able to experience augmented reality multimedia content. These tools are enabled when the user arrives at a Cultural Heritage venue. Until then, the user can only use this screen to take photographs through the 📷 button, present in Figure 3.16a.

After a photo is taken, the user will see a snackbar appear at the bottom of the screen (see Figure 3.16b). This snackbar can redirect the user to the details photo screen in order to view the photo just taken in full screen.

When the user is close to a Cultural Heritage destination, the AR experience can take place and the following sequence will happen, shown in Figure 3.17 (the number of the sequence match the numbers in the Figure):

1. The App makes a notification to alert the user to go to the camera screen.

2. Then the user has to go to the camera screen and the AR experience starts with the appearance of the search icon and the ⊕ button on the screen. The user has now to move their device to allow the App to establish full tracking and detect surface geometry.



(a) (b)

Figure 3.16: (a) Camera screen before user arriving at a CH venue and (b) appearance of a snackbar after the photo is taken

3. When search icon disappears, it means that a geometrical surface was found and the user must click on the ⊕ button so that the AR object is placed in the surface.

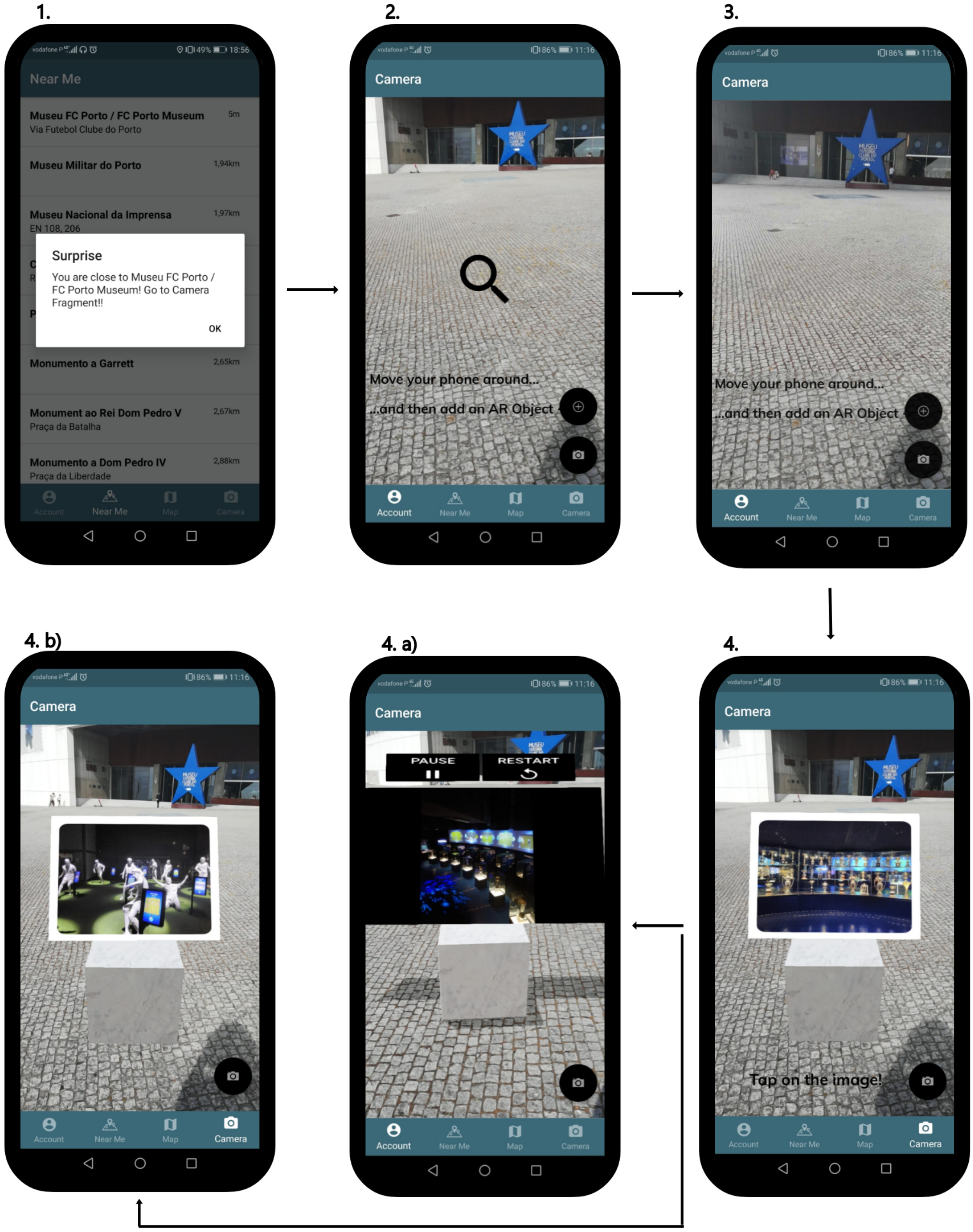
4. Then the ⊕ button will then disappear, but the object is placed. Next, the user has to tap the image and depending on the user context, in terms of light and sound, two things can happen:

   (a) A video will start playing if the user is inserted in a perfect context in terms of noise and brightness. The user will also be able to control the video, through the controls play/pause and replay that appear on top of the video frame.

   (b) If there were a lot of surrounding noise, tapping on the images would only make more images appear.

It should be noted that if it were too bright, but too much surrounding noise, the steps explained above would not have happened. In this case, the user would only have heard an audio about the Cultural Heritage venue.

Throughout the experience, the users can be guided with a tutorial, that can also be visualised in Figure 3.17 in steps 2 to 4. In this way, the users have help in order to know what is the next necessary action to take during the augmented reality process. Initially, this tutorial is automatically enabled, but the user has the possibility of choosing to disable or enable in the Account screen (see Section 3.3.5).



Museu FC Porto / FC Porto Museum 5m
Via Futebol Clube do Porto

Museu Militar do Porto 1,94km

Museu Nacional da Imprensa 1,97km
EN 108, 206

Surprise

You are close to Museu FC Porto / FC Porto Museum! Go to Camera Fragment!!

Monumento a Garrett 2,65km

Monument ao Rei Dom Pedro V 2,67km
Praça da Batalha

Monumento a Dom Pedro IV 2,88km
Praça da Liberdade

Move your phone around...

...and then add an AR Object

Tap on the image!

Figure 3.17: Steps in the AR experience

Chapter 4

Application Development

This Chapter presents relevant information about the implementation of the *NearHeritage* application, specifically the different components of the application, the logic of the architecture employed, the integration of the various API and some external libraries, the process of creating the augmented reality content and other functionalities. Besides that, its also presented results of the use of the application in two different scenarios.

4.1 Architecture

This section presents the system overview of the *NearHeritage* Application developed throughout this dissertation. It is important to note that the application runs only in the client (Android OS) and does not follow a client-server model. As it is possible to see in Figure 4.1, the system is composed of six main components:

- Acquisition of User Location.
- Retrieving Nearby CH Venues.
- Maps Tool.
- Augmented Reality Tool.
- Obtaining User Context Data.
- Managing User Account.

The component Acquisition of User Location utilises the Fused Location Provider API in order to provide the Application with the current user location. This component is always working in the background, and it fetches a new location at a specific time interval.

The component Retrieving Nearby CH Venues fetches data related to the Cultural Heritage venues that are near the user. This is possible to achieve by using the Places API, provided by Foursquare. This component needs the previous component since the input to retrieve the data from the Foursquare API is the current user location.

The Maps Tool component uses the Google Maps SDK to display a map based on Google Maps. This component requires the previous two components because it needs the user location and the location of the nearby CH venues to show these locations to the user in the map.

The Augmented Reality Tool provides the Application with the ability to demonstrate to the users augmented reality content about each Cultural Heritage venues. This component needs to obtain data from the component Retrieving Nearby CH Venues to know if the user is less than twenty meters away from a venue of cultural interest and which venue is, in order to trigger the process of displaying AR to the user. Besides that, this component also depends on another component, the Obtaining User Context Data which will retrieve context data from the light sensor and the Intel Context Sensing SDK. This contextual data is required because the type of multimedia content displayed with the Augmented Reality Tool component changes accordingly to the user context in terms of brightness and the surrounding audio.

The last component is the Managing User Account that gathers and manages user data from the registration process to validate the user login and to display data related to the user and his experience in the application.

The architectural model for the interaction between the User Interface and each one of the components will be explained in the following section.

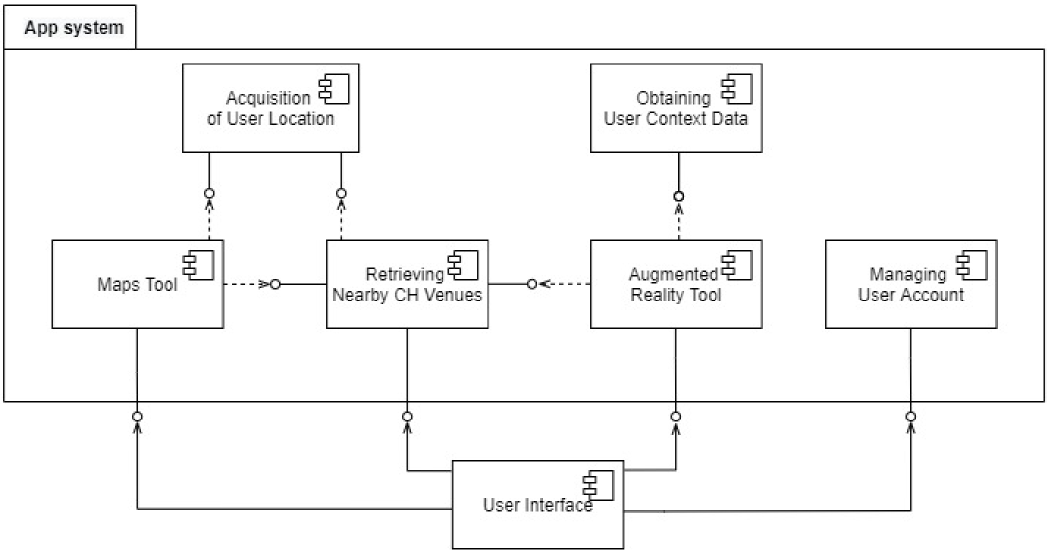

App system
Acquisition of User Location
Obtaining User Context Data
Maps Tool
Retrieving Nearby CH Venues
Augmented Reality Tool
Managing User Account
User Interface

Figure 4.1: Application System Overview

4.1.1 Implemented Model

The model presented in Figure 4.2 handles the interaction between the user interface and each component that was presented in the previous section. This architectural model is formed by a UI controller, a viewModel that serves liveData and a repository that manages, for example, the local and remote data source.

The UI controller consists of a fragment or activity, and its corresponding layout file. So, it is responsible for the layout structure displayed on the screen and can also execute user interface logic.

The ViewModel provides data to a specific UI controller and contains the data handling logic to communicate with the repository. In this sense, it performs UI logic, exposes appropriate data to the UI and manipulates the repository based on actions from the UI. Therefore, it delegates the data-fetching process to the repository. The communication between the viewModel and the UI controller is through LiveData, that is considered an observable data holder.

Finally, the Repository, as previously mentioned, handles data operations so that the rest of the App can retrieve easily needed data from, for example, APIs or databases.

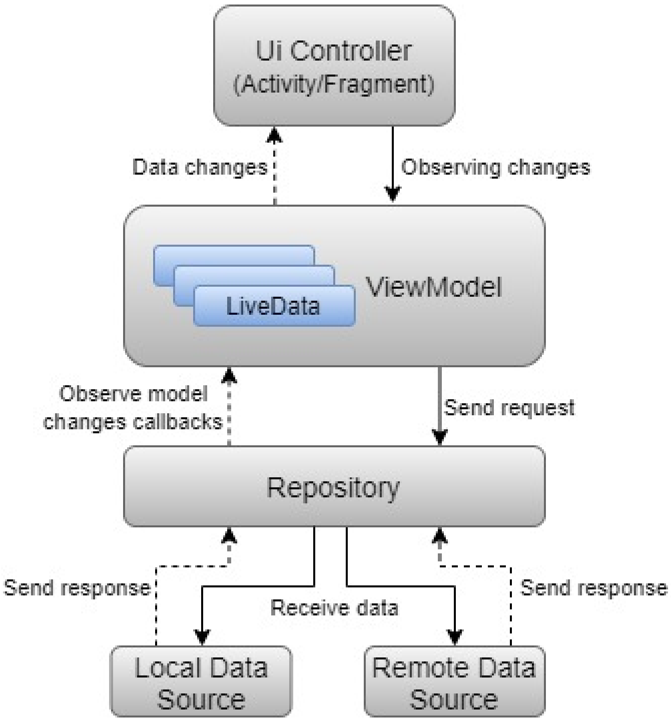


Figure 4.2: Architecture Model

In simple words, the Ui controller, the viewModel and the repository work together in the following manner. The ViewModel will hold a LiveData object, which will be updated with new values that are fetched from the APIs by the repository. The UI controller will be an observer of the changes that occur in that LiveData object. So, when the LiveData object is updated with a new value, the Ui controller will receive the new value and can now make the necessary changes to the UI.

For example, in the *NearHeritage* application, when the location of the user changes, the list of nearby CH venues also changes. Therefore, the UI controller will receive the updated values of the CH venues through a LiveData object and can now make alterations to the UI so that the user can see the new data.

Another vital component in this application is the Navigation Component that handles the navigability between screens.The class *MainActivity.java* plays this important role and holds the navigation host that swaps the destinations when a user navigates in the App. The navigation host implements the controller that manages the navigation by interacting with the navigation graph where it exists all the destinations and the different paths to them.

As the user navigates through the application, the navigation controller is notified of a specific destination or a specific path in the navigation graph that the user wants to go. Then the navigation controller displays the appropriate destination in the navigation host.

4.2 System Components

The system components are briefly presented in Section 4.1 and this section analyses the components in terms of the logic implemented and the required tools (such as APIs, SDKs). In this way, its possible to discuss the role of the components in the functioning of the *NearHeritage* Application.

4.2.1 Acquisition of the User Location

This section analyses the implementation of the component Acquisition of the User Location. As its names indicate, this component is going to provide the Application with the user location by using the Fused Location Provider API (see Section 2.7.3). In this way, the component has a role in the following use cases:

1. Showcasing a list of the nearby Cultural Heritage venues that can be, for example, monuments, landmarks or museums.

2. Providing a map with the user location and the Cultural Heritage venues location.

In use case 1, the user location is the input to retrieve nearby Cultural Heritage venues, and in use case 2, the retrieved user location is going to be used to display a marker of the user in a map.

4.2.1.1 Implementation Approach

It is possible to visualise the architecture of this component in Figure 4.3, that follows the same logic of the one presented in Section 4.1.1. In this way, the repository is named Location Repository, and it will make the API calls to the Fused Location Provider API, in order to obtain the current user location. The NearMe fragment and the Map fragment are the UI controllers that are going to receive the data changes that occur in the Location Repository to display it to the user. For example, in the Map fragment, the user location is shown in a marker on the displayed map.

Besides the two UI controllers, there is also a repository that will need to receive the changes that happen to the location data, the Foursquare Repository. This repository needs to receive data from the location repository so that the new fetched location data can trigger the process of retrieving new nearby venues from the Foursquare API. The development and integration of the Foursquare API will be explained in the next section.

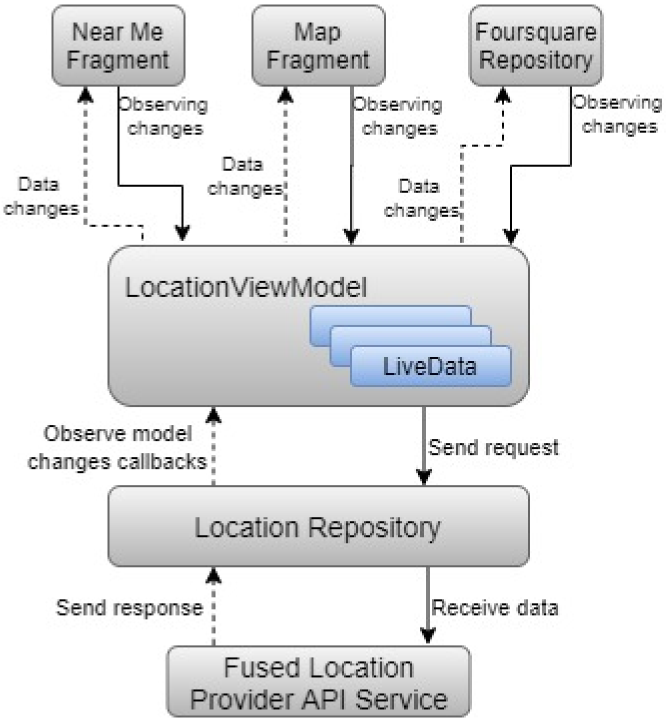


Figure 4.3: Location architecture

The algorithm developed in the location repository to retrieve the user location is based on using the feature of fetching location updates from the Fused Location Provider API. This feature will give location updates to a callback implemented in the location repository at a particular interval. After receiving the callback, the algorithm presented in Figure 4.4 will begin, and the following steps will take place:

1. If it is the first time receiving a location update, the value received will be set in the LiveData object, so that the UI controller can present it to the user.

2. If it is not the first time getting a location update, the value received will be compared with the previously obtained value.

    (a) If the values are equal, the algorithm will do nothing.

    (b) If the values are different, the value received will be set in the LiveData object so that the UI controller can display it.

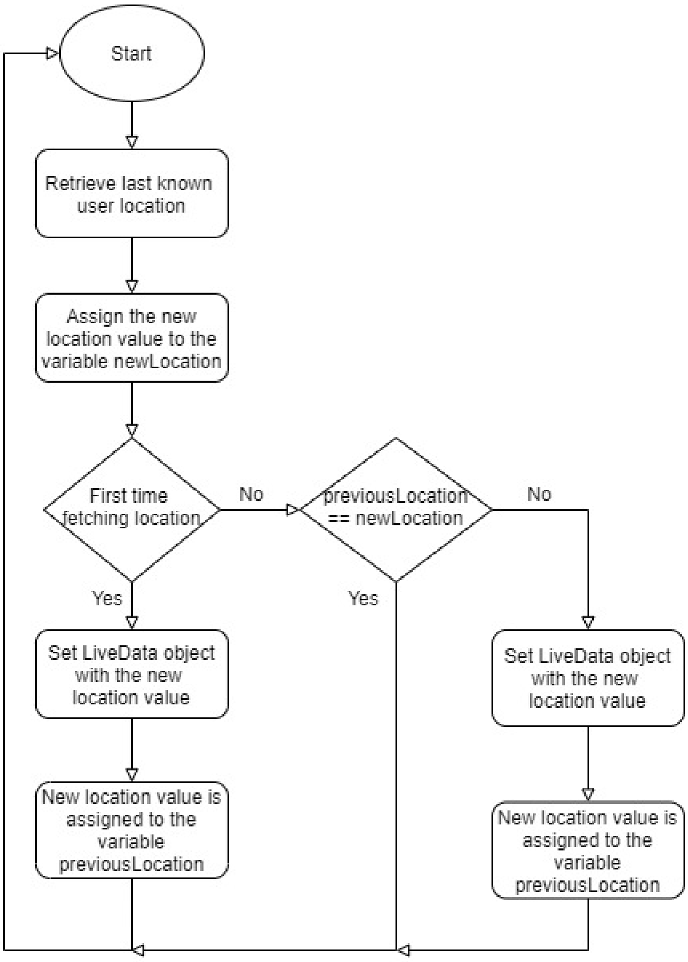

Start
Retrieve last known user location
Assign the new location value to the variable newLocation
First time fetching location
No
previousLocation == newLocation
No
Yes
Yes
Set LiveData object with the new location value
Set LiveData object with the new location value
New location value is assigned to the variable previousLocation
New location value is assigned to the variable previousLocation

Figure 4.4: Location algorithm

4.2.2 Retrieving Nearby CH Venues

This section displays the development of the component Retrieving Nearby CH Venues that is going to fetch data related to the nearby Cultural Heritage venues according to the user location, by using the Places API provided by Foursquare. In this sense, this component plays a part in the accomplishment of the following use cases:

1. Showcasing a list of the nearby Cultural Heritage venues that can be, for example, monuments, landmarks or museums.

2. Providing a map with the user location and the Cultural Heritage venues location.

3. When the user arrives at the Cultural Heritage venue, the user can receive multimedia content about the history of the place through augmented reality techniques.

4. Visualisation of detailed information about each Cultural Heritage venue.

This component performs a crucial role for the functioning of the application, since it retrieves the nearby CH venues for their display in a list, it fetches the locations of the CH venues for their display in a map, it informs the application when a user in close to a CH venue to start the AR experience, and obtains more information about a specific CH venue.

The relationship between the classes of this component and the connection to the Foursquare API also follows the same logic of the architecture discussed at the beginning of this chapter.

In this case, as shown in Figure 4.5, the repository in named the Foursquare Repository and the UI controller that are going to observe the changes that occur in it are the NearMe fragment, the Map fragment, the Venue Details fragment and the Camera fragment.

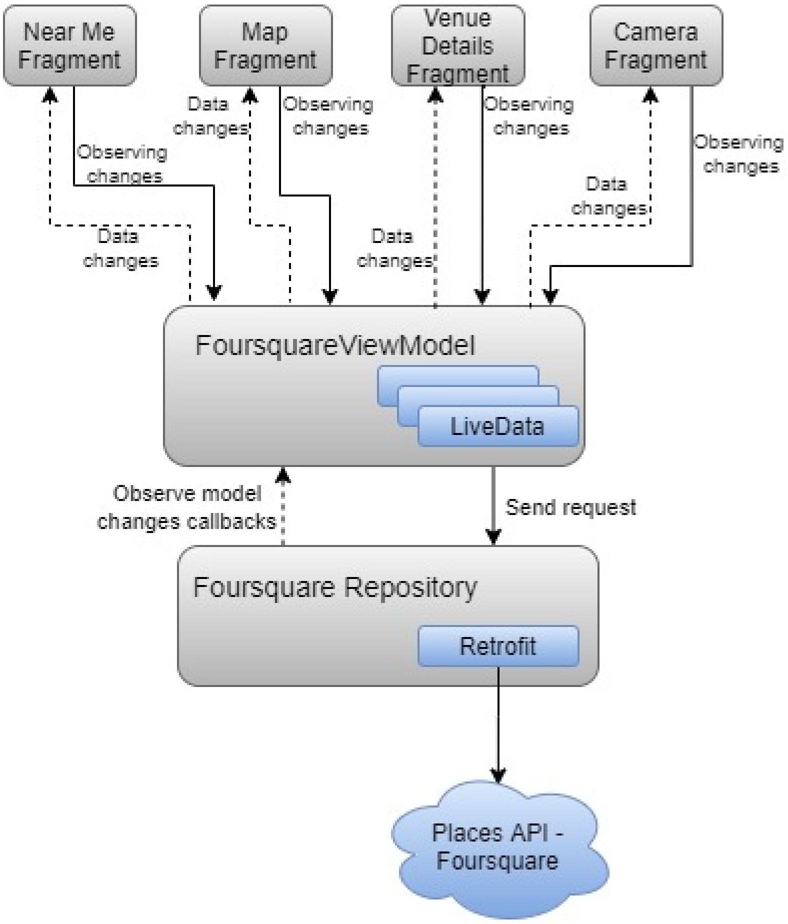


Figure 4.5: Foursquare architecture

The NearMe fragment will require data from Foursquare to present to the user the list of cultural heritage venues nearby to the user location. The map fragment also needs the list of venues but to display it in a map, where each marker represents a venue of the list. The Venue Details fragment requests for data about a specific venue to display to the user more details and information about it. Lastly, the Camera fragment is going to observe data changes from foursquare repository to know when a user is less than twenty meters from a venue of cultural interest and which venue is. The other fragments will also observe that data to present to the user a warning in the screen to go to the Camera screen and enjoy augmented reality content.

A crucial part of the foursquare repository is knowing what the current user location is to begin the process of retrieving new data from the API to provide the UI controllers the changes to the data. For this reason, the foursquare repository is an observer of the location repository and receives all the changes that happen to the user location, as it was already explained in the previous section.

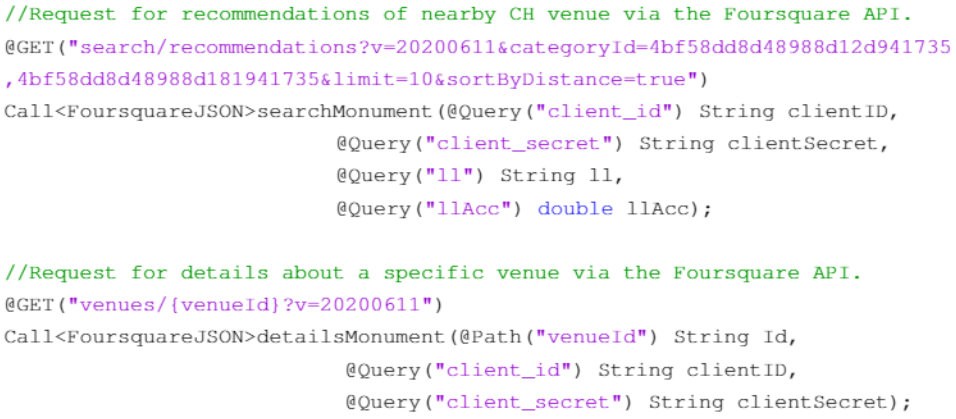
The Foursquare API is a RESTful API based on HTTP requests and JSON responses, so all the requests need to be encrypted and sent via HTTP. In order to do so, it will be necessary to implement libraries such as Retrofit and Gson to manage the HTTP connections and the JSON parsing. In the next section, it will be explained and demonstrated how to use these two external libraries in the Foursquare Repository in order to make requests to the API.

4.2.2.1 Retrofit

In this section, it will be explored how to make requests to the Foursquare API using the Retrofit library. Retrofit [61] provides an implementation of RESTful APIs, and whose primary functionalities are to communicate data between the Android application and RESTful services.

In this App, the endpoints utilised from the Foursquare API are venue recommendations and venue details. The first endpoint returns recommended venues around a specific location to be used specifically in the NearMe, Map and Camera fragment. The second endpoint fetches data about a specific venue according to the venue id that is going to be utilised in the Venue Details fragment.

The HTTP requests for the endpoints are represented as Retrofit Calls definitions within the Android environment. This is possible to visualise in the interface *FoursquareService.java*, present in the Figure 4.6.



```
//Request for recommendations of nearby CH venue via the Foursquare API.
@GET("search/recommendations?v=20200611&categoryId=4bf58dd8d48988d12d941735
,4bf58dd8d48988d181941735&limit=10&sortByDistance=true")
Call<FoursquareJSON>searchMonument(@Query("client_id") String clientID,
                                   @Query("client_secret") String clientSecret,
                                   @Query("ll") String ll,
                                   @Query("llAcc") double llAcc);

//Request for details about a specific venue via the Foursquare API.
@GET("venues/{venueId}?v=20200611")
Call<FoursquareJSON>detailsMonument(@Path("venueId") String Id,
                                    @Query("client_id") String clientID,
                                    @Query("client_secret") String clientSecret);
```

Figure 4.6: Interface for the requests to the Foursquare API

The *searchMonument()*, for instance, corresponds to the endpoint venue recommendations and utilises the annotation @GET, which defines the HTTP protocol and its endpoint. It is necessary to define some parameters to use recommendations through the Foursquare API, that are presented in Table 4.1.

Table 4.1: Functionalities of the parameters to endpoint venue recommendations

Parameters	Functionalities
v	It is a date that represents the version of the API for which the developer expects from Foursquare.
Category_id	The returned results will only be within the categories specified.
Limit	The call to the API will only return a limit number of venues.
SortbyDistace	The venues returned are the ones closest to the location provided.
Client_id and Client_secret	The credentials necessary to make the request to the API.
ll	Represents the latitude and longitude of the user's location.
llAcc	Used to improve location matching.

After all the functions are defined in the interface, it is possible to build a request to the Foursquare API through a retrofit instance. After the request is complete, it is still necessary to parse the obtained JSON response for the UI controller be capable of displaying the data to the users. This will be explained in the next section.

4.2.2.2 Gson

In this section, it will be explored how to parse responses from the Foursquare API to the android application using the Gson library. Gson [62] is an open-source Java library that can be used to convert Java Objects into their JSON representation or convert a JSON string to an equivalent Java object. In this case, the Gson library will be used to deserialise the JSON string into Java objects.

The JSON response of the Foursquare API has many levels, and each level has a corresponding Java Object. In the case of the searchMonument(), the levels are: JSON strings-> response -> group -> results. In this way, it is necessary to use the data classes FoursquareJSON, FoursquareResponse, FoursquareGroup, and FoursquareResults to get to the level where each UI controller can use the venues data. The Gson library will take care of deserialising the JSON response through the many levels of data.

4.2.3 Map Tool

This component uses the Google Maps SDK to allow the Application the possibility of displaying a map based on Google Maps. In this way, users will have a visual perception of their location and of the location of the nearby Cultural Heritage venues. Therefore, this component provides the accomplishment of the use case: Providing a map with the user location and the Cultural Heritage venues location.

4.2.3.1 Implementation approach

The Map Fragment is the UI controller that implements the life cycle management and the services for visualising the GoogleMap object provided by Google Maps SDK. When the GoogleMap object is ready for display, the function *onMapReady* will be called and two types of markers will be set:

- User Marker that represents the location of the user.
- List of Cultural Markers that represents all the locations of the Cultural Heritage venues.

All the markers locations are requested to the Location Repository and Foursquare Repository, already explained in Sections 4.2.1 and 4.2.2. When the Map Fragment receives new data from these repositories, two functions are called to make the necessary changes to the locations of the markers.

In addition to these functionalities, this fragment can also receive data from other fragments. For example, in the Venue Details screen, there is a button that redirects the user to the Map screen and will change the position of the map's camera to the location of the selected venue, as it was already mentioned in Section 3.3.3.

In order to do so, it is necessary to pass the name of the venue from the Venue Details fragment to the Map fragment. The navigation component allows attaching data to a navigation operation by defining arguments for a destination.

Therefore, it is attached to the path from the Venue Details screen to the Map screen, the name of the venue. The UI controller in the Map fragment will verified if it received any arguments. In case it did, it will find the marker of that venue by the received name, and move the map camera to the venue location

4.2.4 Augmented Reality Tool

The component Augmented Reality Tool was built for the Application to have the ability to display augmented reality content about each Cultural Heritage venues to the users. In this way, this component has a part in providing the following use cases:

1. When the user arrives at the Cultural Heritage venue, the user can receive multimedia content about the history of the place through augmented reality techniques.
2. Taking pictures of the cultural places and throughout the augmented reality experience.

The UI controller for the Camera screen (see Section 3.3.7) is the core part of all the development that led to the creation of an enjoyable augmented reality experience. This UI controller observes when the user is near a cultural heritage venue, that was already explained in Section 4.2.2, triggering the retrieve of context data from the light sensor and from the Intel Context Sensing SDK, which in its turn triggers the start of the AR content and all its functionalities. This section, analyses all the elements behind the implementation of the augmented reality experience.

4.2.4.1 Obtaining User Context Data

The type of multimedia content displayed with the help of augmented reality tools will change accordingly to the user context in terms of brightness and the surrounding audio.

The values of brightness can be divided into three levels, which are: 10 to 50 lx is a dim environment, 10 to 1000 lx is normal, and above 10000 lx is considerable too bright. The last level is a poor environment to display any visual multimedia content such as image or video. In this case, it would only trigger audio content.

In order to retrieve the level of brightness of the surrounding user environment, it was used the built-in Android environment sensor that measures illuminance. This sensor is hardware-based and usually exists in all smartphone because most device manufacturers use it to control screen brightness. Besides that, the sensor returns a single sensor value for each data event, not requiring any calibration or filtering.

The sensor manager from Android allows access to the device light sensor by registering the sensor.TYPE_LIGHT, and then the App will receive any changes that happen to the illuminance data.

The audio surrounding the user can be obtained through the utilisation of the Intel Context Sensing SDK, described in Section 2.7.5. This library returns a weighted list based on the type of audio identified, from the most probable to the least probable. The audio can be classified in SPEECH, MUSIC, MECHANICAL, MOTION, and CROW_CHATTER. In case the higher percentage is classified as SPEECH or CROWD_CHATTER, the multimedia content to be displayed is images; otherwise, it would be video.

As in the previous sections, the architecture followed was the one presented in Section 4.1.1. In this case, as displayed in Figure 4.7, it will exist a Camera Repository, that will request data to the Intel Context Sensing SDK and to the Android Sensor Manager, and the only observer is the UI Controller for the Camera screen that will receive changes to the context variables. This UI Controller will only receive one time the alterations to the data because the App only needs to know the surrounding context when the user is near to a venue and wants to start the AR experience.

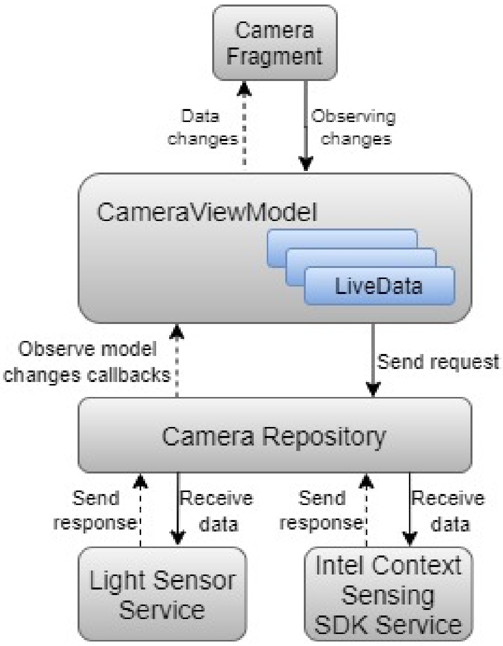


Figure 4.7: Camera architecture

4.2.4.2 Creation of Augmented Reality Content

After all the initial setup, the UI controller can now add ArFragment provided by Sceneform SDK to the camera screen. The ArFragment automatically handles and maintains the ARSceneView, the ARCore session, performs the runtime checks necessary for the App to work, and is where the AR objects will be placed.

The ARCore session and the ARSceneView have essentials roles in the development of the AR experience. The ARCore session keeps track of the planes that have been identified and the objects that have been anchored. While, the ARSceneView renders the camera images from the session onto its surface, highlighting the detected planes and holding all the nodes, in which the virtual objects will be rendered.

For this application, it was built a custom ArFragment to create a more realistic and immersive AR experience. In this way, the environmental HDR configuration was set in the ArFragment to provide a more natural shadowing and reflection of the AR objects.

In order to start the detection of planes, the user must move the phone around to enable the ArFragment to detect "visually distinct features" (feature points) in each captured image. In this way, it is possible to combine these points with the device sensors to determine the position and orientation of the camera (pose). When a plane is detected, it will be highlighted, and the user can now click on the ⊕ button to place the AR object (renderable).

The action of clicking on that button will trigger the process of identifying an anchor to place the renderable. Therefore, it will estimate the screen centre and perform a hit test on that particular point. If the hit test discovers that the plane was detected in that particular point, the Camera Fragment will call the function *addToScene()* and pass the anchor to it.

In the function *addToScene()*, it will be created a node that will contain the AR object created in the class *UiRenderable*. This node will be set on the anchor node that is associated with the point in the discovered plane.

The class *UiRenderable* is responsible for creating a node with a 3D cube, that has a marble texture, and a child node that will be directly placed above the cube. The child node will initially have the renderable of a 2D image with a frame created dynamically around it, as it is possible to observe in Figure 4.8. The idea behind placing the image above a cube is to provide more height to the AR object so that the user does not have to be looking at the ground to see the AR content. Besides that, it was also to try mimicking the podiums where above usually are pieces of artwork in the museums.

Then, the user can tap the image above the cube, and two things can happen depending on the results from the context data processed in component Obtaining User Context Data:

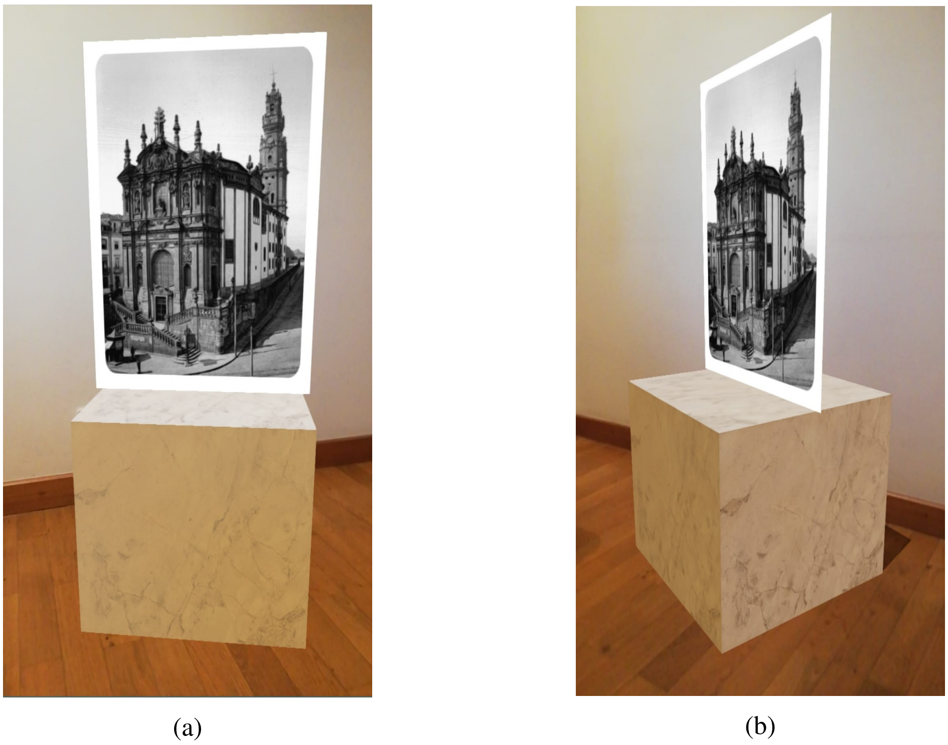
1. A 2D video can be played about the cultural heritage venue, with controls to play/pause and restart the video.

2. A serious of other 2D images related to the venue will be displayed.

These two possibilities for the AR experience will be displayed in the Section 4.3.

Some challenges arise due to working in a 3D space with 2D elements. For instance, the objects need to be visible to the users no matter the camera position. As demonstrated in Figure 4.8, the 2D elements might not always face the user, and the elements must always be facing the user. In order to solve this problem, it was used a billboard technique that is often used in computer games for rendering objects. Basically, it is necessary to obtain the camera position of the user and the position of the 2D elements. Then, the result of subtracting the two values is a direction vector that will be used in the method *Quaternion.lookRotation()* to achieve the 3D rotation of the element.

In the different stages of the AR process, it was developed a tutorial to help the user understand the next necessary actions to take during the evolution of the AR experience, as it was already explained in Section 3.3.7. This tutorial is a simple text with a blinking animation that will appear in two different phases of the AR experience. The first one is to indicate to the user that it is necessary to move the phone around and to click on the button. The second one is to let the user know that it is required to tap on the image above the cube to appear new AR content.



(a) (b)

Figure 4.8: (a) and (b) Behaviour of a 2D image in a three-dimensional space

4.2.4.3 Taking Photographs

The UI controller provides one more functionality to the camera screen that is allowing the user to take photos of the experience by clicking on the 📷 button (see Section 3.3.7).

Each photo will require a unique file that is going to be created by using the standard pictures directory, and an album named "Sceneform". The photos name will be based on the time, so any overwrite between them will not happen. The method that takes the photo uses the PixelCopy API to capture a screenshot of the ArSceneView. This is done asynchronous since it happens between frames. Then the bitmap of the captured screenshot is saved to the smartphone.

Afterwards, a snackbar will appear to redirect the user to the Photo Gallery screen (see Section 3.3.6), that will display all the photographs taken by the user. The photographs are fetched from the album Sceneform from the pictures directory.

4.2.5 Managing User Account

The component Managing User Account focuses on gathering user data from the registration process to validate the login of the user and to display data related to the user and his experience in the App. In this way, this component implements the following application uses cases:

1. User registration in the App.

2. Presenting personal user information's provided when the user registered, such as username, e-mail, date of birth, age, nationality and gender.
3. Possibility of editing personal account information's.
4. Capability of editing the account password.
5. Logout of the account.
6. Deleting the account.
7. Power to decide enabling or disabling the camera tutorial.
8. Displaying all the photos taken in a photo gallery.
9. Ability to share the photographs.
10. Deleting photos from the photo gallery.

This component allows the application to have the ability to register and login a user and to modify the user data, as it was already mentioned. Additionality, it provides the display, share or deletion of photographs taken by the user with the app, and it offers the chance to enable or disable the tutorial associated with the augmented reality experience.

In order for this component to work, it is necessary to persist data while the Application in being used to store, for example, first-time indications of the utilisation of the App and to save the user data that is considerably small and is not worth creating a database. In this way, it was employed a form of data stored named *SharedPreferences*, that is explained in Section 4.2.5.1. Additionally, in Sections 4.2.5.2 and 4.2.5.3 its discussed the employment of this form of data store to perform the login and registration process and to alter user data. Finally, Section 4.2.5.4 analyses the functionalities that this component is able of performing with photos taken by the user.

4.2.5.1 Interaction with Shared Preferences

SharedPreferences is an interface that provides not only access to user preference data but also the ability to modify it. The stored values are presented in a key-value format, in which each preference has a key associated with the saved value. All the preferences kept in *SharedPreferences* are available to all the components in the App, but it is not accessible by other Apps. In addition, the data present in *SharedPreferences* can only be cleared by performing one of the three following options:

- Uninstalling the application.
- Clearing the application data through smartphone settings.
- Clearing programmatically through the *SharedPreferences* method clear().

As *NearHeritage* has the purpose of establishing a new concept of tour guide applications that take advantage of augmented reality tools to provide more content to the user, it was not a priority to build a database to store the user data. In this way, the *SharedPreferences* was implemented to perform the logic of user logging in and out, deleting the account, displaying the user data in the Account fragment and store some necessary flags.

Therefore, the *SharedPreferences* is used to manage all the data that the user fills in the registration process and to maintain two boolean values. These boolean values are two flags: the *is_user_login* and the *is_tutorial*. The first one is to know if the user is logged in or not. The second one is to display or omit a tutorial present on the Camera fragment.

The is_user_login flag is employed in the *MainActivity.java* class. In case the flag is true, the App will open the NearMe screen. If the flag returns false, the App will redirect to the Login screen.

4.2.5.2 Login and Registration

In the login process, the inputs provided by the user - username and password - are compared to the ones saved in the *SharedPreferences*, using the class *UserSessionManager.java*. In case the inputs do not match, the user will be notified.

In the registration process, all the inputs provided by the user - username, e-mail, password, nationality, gender and date of birth - are in some way verified using the *UserUtils.java*:

1. Verification if all the parameters are filled.
2. Verifies if the username, password and nationality have no whitespaces and more than four letters.
3. Verifies if the e-mail has the usual format (xxx@xxx.xx).
4. Verifies if the user is of age for using the App.

4.2.5.3 Updating User Data

The Account fragment was developed around allowing the users to visualise their data and to modify it. In this way, it is necessary to implement the retrieve and alteration of the user data stored in *SharedPreferences*. The user data was provided by the user during the registration process already discussed in the previous section.

So, the Account fragment implements various functions developed in the *UserSessionManager.java* class, that manages the data present in *SharedPreferences*. The screens for all the following functionalities can be visualised in Section 3.3.5.

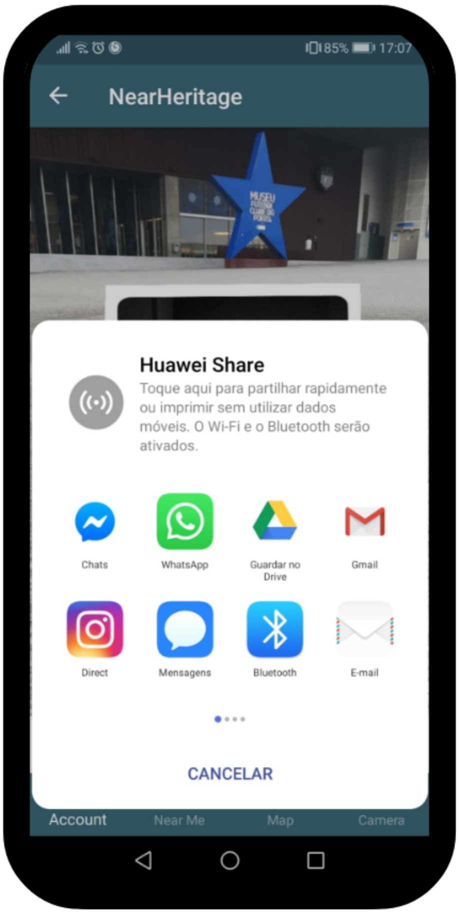
- updateProfileDetails(String name, String mail, String nat, String gender, String birth, int age) – updating the values present in *SharedPreferences* with the new values provided by the user in the Edit Personal Details screen.

- updatePass(String pass) - updating just the password value present in *SharedPreferences* thought the Change Password screen.
- updateTutorial(Boolean tutorial) - for changing the boolean value of the flag for displaying or not the tutorial in the Camera Screen, through the user clicking on the button "Enable/Disable Camera Tutorial" present in the Account screen.
- logout()- for changing the state of the user login to false, and the App will open the Login screen. This is possible by the user clicking on the button “Logout” present in the Account screen.
- deleteUser()- for deleting all the data saved in *SharedPreferences* and the Login screen will be open. This is possible by the user clicking on the button “Delete Account” present in the Account screen.

4.2.5.4 Managing User Photographs

The photographs taken by the user in the Camera screen (see Section 3.3.7) can be fetched from the album Sceneform in the directory pictures stored on the Android device. In this way, it is possible to display the photographs as it is represented in the Photo Gallery screen (see Section 3.3.6).

The user will be able to delete and share the photos in the Photo Details screen (see Section 3.3.6). The sharing of photos is done through an intent of the type ACTION_SEND that allows to send data across Android apps. In order to send the data, it is necessary to specify the data and its type, and the system will identify compatible receiving activities and display them to the user, as it is possible to visualise in Figure 4.9. In order to delete a photograph, it will be necessary to create a file variable with the path to the image, and then call the function *file.delete()*.



Huawei Share

Toque aqui para partilhar rapidamente ou imprimir sem utilizar dados móveis. O Wi-Fi e o Bluetooth serão ativados.

Figure 4.9: Sharing Photographs

4.3 Results

This section analyses the results of the use of the *NearHeritage* Android application, especially the benefits of using augmented reality contents. The results showcase two different scenarios where the user context is different so that the functionalities of the prototype can be evaluated. The Cultural Heritage venues chosen are in two different areas of the city of Porto, Portugal.

4.3.1 First Scenario - *Museu Fc Porto*

The first scenario takes place in the *Museu Fc Porto*. In the day that the results were captured, there was not many people walking around near the venue, nor nothing that could be prejudicial to the user context. So, in this case, the user was inserted in a perfect context in terms of brightness and surrounding noise, which means that the type of multimedia content displayed through the AR Tool is video.

In this scenario, the user first needs to search for the AR plane, then place the AR object, which is a cub with an image on top, and afterwards, the user can tap the image, and the video will start playing. Additionally, the user has at his disposal the controls for the video. This is possible to visualise in Figure 4.10.

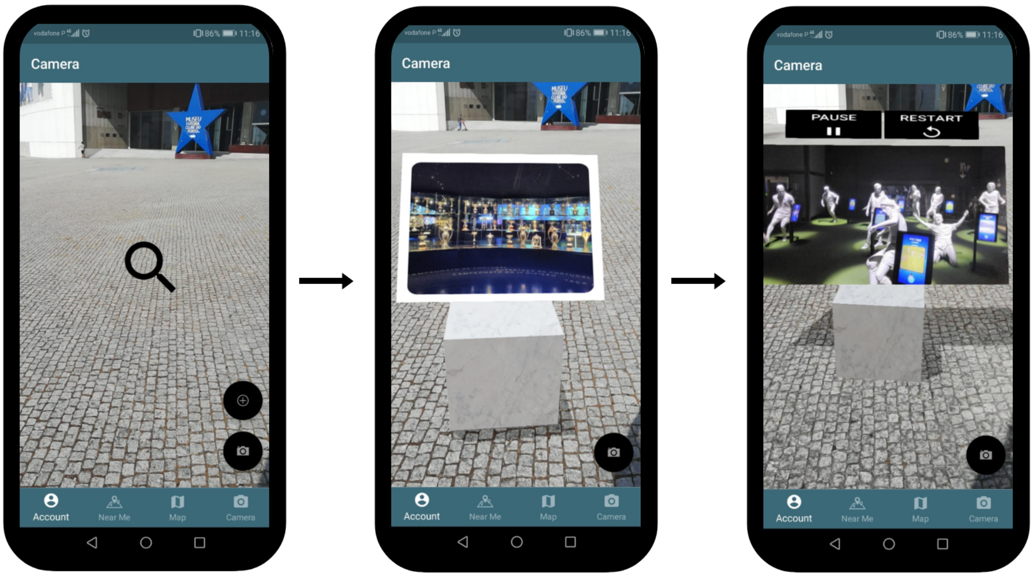


Figure 4.10: First Scenario of the AR experience

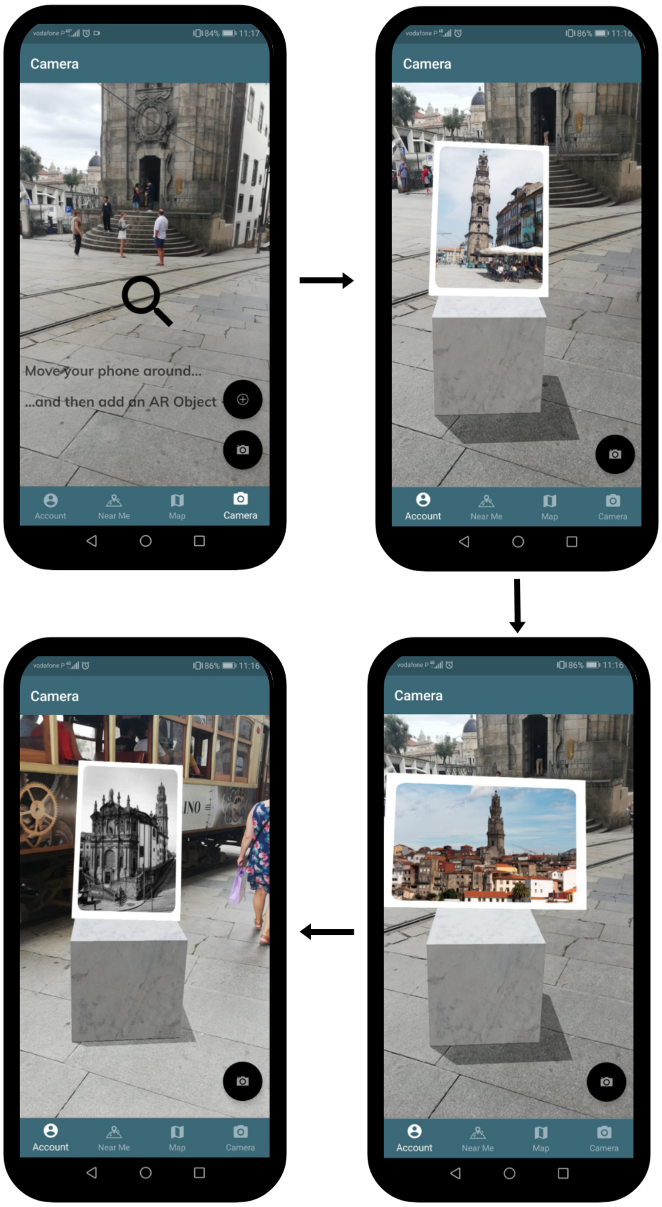
The overall experience with the augmented reality content in this scenario went has it was expected, it was easy to find an AR plane to place the object. Also, the location of the *Museu Fc Porto* helped because it had a large area surrounding the venue to experiment with the AR content without having people walking through the AR object.

The one downside was that when the AR Tool finds an AR plane, it will highlight that plane but since the floor is a light colour and also had the sun streaming in it, it was not possible to see the plane highlighted. As most of the experiments of this dissertation were carried out at home, this problem was not detected and will have to be corrected in future work.

4.3.2 Second Scenario - *Torre dos Clérigos*

The second scenario takes place in the *Torre dos Clérigos*. In the day that the results were taken, there was a lot a people walking around near the venue and, consequently, there was too much surrounding noise. So, in this case, the user was inserted in a good context in terms of brightness, but not in terms of surrounding noise, which means that the type of multimedia content displayed through the AR tool is images.

In this scenario, the user first needs to search for the AR plane, then place the AR object, which is a cub with an image on top, and afterwards, the user can tap the image, and more images will be displayed. This is possible to visualise in Figure 4.11.



Move your phone around...

...and then add an AR Object

Figure 4.11: Second Scenario of the AR experience

In this case, the augmented reality experience did not go has expected due to many factors. First, since most of the experiments of this dissertation had to be carried out at home, it was not considered inclined planes which difficult the process of finding and placement of the AR object. Second, the number of people near the venue and the small area that surrounded the venue, complicated the placement of the AR object because there were not many free spaces near the venue to conduct the experiments.

In the future work, it will be necessary to correct these small problems with the augmented reality content, which were not considered because, as it was already mentioned, most of the tests were carried out at home in a perfect environment in terms of light, sound, type of floor, among other factors.

Besides the augmented reality experience, the rest of the *NearHeritage* app performed well in both scenarios. In the following link, `https://paginas.fe.up.pt/~up201505756/master_thesis/video_thesis.mp4`, it is possible to visualise a video to demonstrate the use of all the functionalities of the app in the *Museu Fc Porto* in a context where the type of multimedia content displayed through the AR Tool is images.

Although there are some improvements to make to the display of augmented reality contents, there results still showcase that the concept of the MAR application can improve the user experience in discovering and learning more about Cultural Heritage around the world, creating an interactive and enjoyable adventure.

Chapter 5

Conclusion

The focus of this dissertation was to study and develop a mobile augmented reality system in order to create an outdoor experience in Cultural Heritage places. In this way, the system needs to be able to guide users that are visiting a new city and want to discover the vast culture and history that it has to offer.

First of all, a research of the dissertation thematic was conducted to understand the different aspects involving the problem under study. In this regard, it was possible to comprehend concepts and methods that are the baseline to identify, select and integrate different approaches and emerging technologies as a way of improving the overall user experience. One of these emerging technologies is augmented reality, which has the ability to enhance the way users receive and interact with multimedia content and present information about different Cultural Heritage places around a city.

After the careful analysis of the different concepts and applications that already exist in this research field, the implementation of the Android application began. This application plays the role of a tour guide when a user visits a new city by suggesting nearby monuments, landmarks, museum and any other type of cultural and/or historical place, for users to visit.

When a user arrives at a cultural point-of-interest, it will trigger the display of information through multimedia content that will be presented by using augmented reality techniques. This will transform the user experience in a way that discovering and learning more about Cultural Heritage will be much more interactive and amusing.

The implementation of this dissertation was only possible through the utilisation of built-in sensors and hardware of Android devices, and various API, presented in Section 2.7, to retrieve information about the user context, about the Cultural Heritage places and for the creation and development of augmented reality content.

The results for the concept of this mobile augmented reality system demonstrates that it should be developed more applications that take advantage of AR tools, in order to provide more appealing and enthusiastic ways of presenting the vast Cultural Heritage that each country has to offer.

5.1 Main Difficulties

In the development of this dissertation, some difficulties were felt. The first one was having to change the dissertation theme due to the pandemic time that we all live in, and having to work from home. This caused a sense of discouragement because part of the planning and progress already made had to be redone.

Besides that, having to work from home had some difficulties. For instance, there is not a clear separation of work time and rest time because everything is done in the same space. In addition, having to work alone and not be surrounded by other colleagues, who were also going through the process of developing the dissertation, can be mentally challenging.

Finally, in terms of development tools and techniques utilised, it was not exactly difficulties, it was just a significant learning curve because the knowledge in developing Android application and using AR tools was little. This was challenging, but at the same time, it was an extra incentive to learn more about these tools and techniques.

5.2 Future Work

In terms of future work, there are still developments that can be implemented in this dissertation, that could not be achieved throughout the semester.

First, it is required to conduct usability tests with real users to test the various ways in which the user interact with the application and to see if the experience aligns with user expectations. Besides that, it would be necessary to refine the reality augmented content and the way to present it. For instance, the AR experience could be triggered by the user taken a photograph of the CH place.

Second, it was not implemented the identification of the Cultural Heritage places through the utilisation of neural networks. In this way, it would be essential to create an extensive data set with real context and images data in order to experiment with neural networks. Through this method, it could be possible to obtained better descriptions of the context and place that user is at and therefore making better decisions regarding AR information.

Third, it would be interesting to provide users with a more quick and instinctive way of giving feedback on the AR content that is being presented. For instance, through the implementation of voice interaction between the users and the augmented reality experience.

Lastly, to improve the overall user experience and the places that are suggested to him, it would be interesting to see the results of updating the user profile and preferences depending on the places that the user decided to visit and the feedback it gave to them.

References

[1] Răzvan Gabriel Boboc, Mihai Duguleană, Gheorghe-Daniel Voinea, Cristian-Cezar Postelnicu, Dorin-Mircea Popovici, and Marcello Carrozzino. Mobile augmented reality for cultural heritage: Following the footsteps of ovid among different locations in europe. *Sustainability*, 11(4), 2019.

[2] Guida Casella and Moises Coelho. Augmented heritage: Situating augmented reality mobile apps in cultural heritage communication. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, ISDOC '13, page 138–140, New York, NY, USA, 2013. Association for Computing Machinery.

[3] David Marimon, Cristina Sarasua, Paula Carrasco, Roberto Álvarez, Javier Montesa, Tomasz Adamek, Idoia Romero, Mario Ortega, and Pablo Gascó. Mobiar: Tourist experiences through mobile augmented reality. *Information and communication technologies in tourism*, 01 2010.

[4] Pietro Vecchio, Francesca Mele, Lucio Tommaso De Paolis, Italo Epicoco, Marco Mancini, and Giovanni Aloisio. Cloud computing and augmented reality for cultural heritage. In Lucio Tommaso De Paolis and Antonio Mongelli, editors, *Augmented and Virtual Reality*, pages 51–60, Cham, 2015. Springer International Publishing.

[5] F. Vera, J.A. Sa&acute;nchez, and O. Cervantes. Enhancing User Experience in Points of Interest with Augmented Reality. *International Journal of Computer Theory and Engineering*, 8(6), dec 2016.

[6] A.K. Dey, G.D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2-4):97–166, 2001.

[7] A.K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

[8] Andreas Zimmermann, Andreas Lorenz, and Reinhard Oppermann. An operational definition of context. In Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors, *Modeling and Using Context*, pages 558–571, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[9] C. Emmanouilidis, R.-A. Koutsiamanis, and A. Tasidou. Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications*, 36(1):103–125, 2013.

[10] Liliana Ardissono, Tsvi Kuflik, and Daniela Petrelli. Personalization in cultural heritage: the road travelled and the one ahead. *User Modeling and User-Adapted Interaction*, 22(1):73–99, Apr 2012.

[11] K. Veningston and R. Shanmugalakshmi. Personalized location aware recommendation system. In *2015 International Conference on Advanced Computing and Communication Systems*, pages 1–6, 2015.

[12] Andreas Zimmermann, Marcus Specht, and Andreas Lorenz. Personalization and context management. *User Modeling and User-Adapted Interaction*, 15(3):275–302, Aug 2005.

[13] Elena Not and Daniela Petrelli. Blending customisation, context-awareness and adaptivity for personalised tangible interaction in cultural heritage. *International Journal of Human-Computer Studies*, 114:3 – 19, 2018. Advanced User Interfaces for Cultural Heritage.

[14] Semih Utku. and Canan Eren Atay. A mobile location-aware recommendation system. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval - Volume 1: KDIR, (IC3K 2014)*, pages 176–183. INSTICC, SciTePress, 2014.

[15] Wanvimol Nadee. *Modelling user profiles for recommender systems*. PhD thesis, Queensland University of Technology, 2016.

[16] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden. The city trip planner: An expert system for tourists. *Expert Systems with Applications*, 38(6):6540 – 6546, 2011.

[17] Montserrat Batet, Antonio Moreno, David Sánchez, David Isern, and Aïda Valls. Turist@: Agent-based personalised recommendation of tourist activities. *Expert Systems with Applications*, 39(8):7319 – 7329, 2012.

[18] Luis Castillo, Eva Armengol, Eva Onaindía, Laura Sebastiá, Jesús González-Boticario, Antonio Rodríguez, Susana Fernández, Juan D. Arias, and Daniel Borrajo. Samap: An user-oriented adaptive system for planning tourist visits. *Expert Systems with Applications*, 34(2):1318 – 1332, 2008.

[19] Lorenzo Massai, Paolo Nesi, and Gianni Pantaleo. Paval: A location-aware virtual personal assistant for retrieving geolocated points of interest and location-based services. *Engineering Applications of Artificial Intelligence*, 77:70 – 85, 2019.

[20] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39:319 – 333, 2014.

[21] María del Carmen Rodríguez Hernández, S. Ilarri, Raquel Trillo, and Ramon Hermoso. Location-aware recommendation systems: Where we are and where we recommend to go. *CEUR Workshop Proceedings*, 1405:1–8, 01 2015.

[22] I. Benouaret and D. Lenne. Personalizing the museum experience through context-aware recommendations. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 743–748, 2015.

[23] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alexander Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32(3):67–80, 1 2011.

[24] Mauro Ceconello and Davide Spallazzo. Exploring the city. valorisation of culture through situated and informal learning. In *Putting Tradition into Practice: Heritage, Place and Design*, pages 579–586, Cham, 2018. Springer International Publishing.

[25] S. Gada, V. Mehta, K. Kanchan, C. Jain, and P. Raut. Monument recognition using deep neural networks. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–6, 2017.

[26] Giuseppe Amato, Fabrizio Falchi, and Claudio Gennaro. Fast image classification for monument recognition. *J. Comput. Cult. Herit.*, 8(4), August 2015.

[27] T. Chen, K. Wu, K.-H. Yap, Z. Li, and F.S. Tsai. A survey on mobile landmark recognition for information retrieval. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 625–630, 2009. cited By 24.

[28] G. Amato, F. Falchi, and F. Rabitti. Landmark recognition in visito tuscany. *Communications in Computer and Information Science*, 247 CCIS:1–13, 2012. cited By 6.

[29] Wengang Zhou, Houqiang Li, and Qi Tian. Recent advance in content-based image retrieval: A literature survey. *CoRR*, abs/1706.06064, 2017.

[30] R. A. Kambau, Z. A. Hasibuan, and M. O. Pratama. Classification for multiformat object of cultural heritage using deep learning. In *2018 Third International Conference on Informatics and Computing (ICIC)*, pages 1–7, 2018.

[31] Jose Llamas, Pedro M. Lerones, Eduardo Zalama, and Jaime Gómez-García-Bermejo. Applying deep learning techniques to cultural heritage images within the inception project. In *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection*, pages 25–32, Cham, 2016. Springer International Publishing.

[32] Jose Llamas, Pedro M. Lerones, Roberto Medina, Eduardo Zalama, and Jaime Gómez-García-Bermejo. Classification of architectural heritage images using deep learning techniques. *Applied Sciences*, 7(10):992, Sep 2017.

[33] M. Ćosović, A. Amelio, and E. Junuz. Classification methods in cultural heritage. In *VIPERC@IRCDL*, volume 2320, pages 13–24, 2019.

[34] T. Chen, K. Wu, K. Yap, Z. Li, and F. S. Tsai. A survey on mobile landmark recognition for information retrieval. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 625–630, 2009.

[35] Giuseppe Amato, Fabrizio Falchi, and Lucia Vadicamo. Visual recognition of ancient inscriptions using convolutional neural network and fisher vector. *J. Comput. Cult. Herit.*, 9(4), December 2016.

[36] V. Palma. Towards deep learning for architecture: A monument recognition mobile app. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W9:551–556, 2019.

[37] Radmila Janković. Machine learning models for cultural heritage image classification: Comparison based on attribute selection. *Information*, 11(1):12, Dec 2019.

[38] Mafkereseb Kassahun Bekele, Roberto Pierdicca, Emanuele Frontoni, Eva Savina Malinverni, and James Gain. A survey of augmented, virtual, and mixed reality for cultural heritage. *J. Comput. Cult. Herit.*, 11(2), March 2018.

[39] Chris D. Kounavis, Anna E. Kasimati, and Efpraxia D. Zamani. Enhancing the tourism experience through mobile augmented reality: Challenges and prospects. *International Journal of Engineering Business Management*, 4:10, 2012.

[40] Mihai Duguleana, Raffaello Brodi, Florin Girbacia, Cristian Postelnicu, Octavian Machidon, and Marcello Carrozzino. Time-travelling with mobile augmented reality: A case study on the piazza dei miracoli. In *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection*, pages 902–912, Cham, 2016. Springer International Publishing.

[41] Luís Pádua, Telmo Adão, David Narciso, António Cunha, Luís Magalhães, and Emanuel Peres. Towards Modern Cost-Effective and Lightweight Augmented Reality Setups. In *Virtual and Augmented Reality*, pages 396–423. IGI Global, 2018.

[42] Isabel Pedersen, Nathan Gale, Pejman Mirza-Babaei, and Samantha Reid. More than meets the eye: The benefits of augmented reality and holographic displays for digital cultural heritage. *J. Comput. Cult. Herit.*, 10(2), March 2017.

[43] Chris Panou, Lemonia Ragia, Despoina Dimelli, and Katerina Mania. An architecture for mobile outdoors augmented reality for cultural heritage. *ISPRS International Journal of Geo-Information*, 7(12):463, Nov 2018.

[44] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui. Mobile augmented reality survey: From where we are to where we go. *IEEE Access*, 5:6917–6950, 2017.

[45] Clemens Arth, Raphael Grasset, Lukas Gruber, Tobias Langlotz, Ro Mulloni, Dieter Schmalstieg, and Daniel Wagner. The history of mobile augmented reality developments in mobile ar over the last almost 50 years, Nov 2015.

[46] Fabricio Herpich, Renan Luigi Martins Guarese, and Liane Margarida Rockenbach Tarouco. A comparative analysis of augmented reality frameworks aimed at the development of educational applications. *Creative Education*, 08:1433–1451, 2017.

[47] Gheorghe-Daniel Voinea, Florin Girbacia, Cristian Cezar Postelnicu, and Anabela Marto. Exploring cultural heritage using augmented reality through google's project tango and arcore. In Mihai Duguleană, Marcello Carrozzino, Matjaž Gams, and Iulian Tanea, editors, *VR Technologies in Cultural Heritage*, pages 93–106, Cham, 2019. Springer International Publishing.

[48] Google. Arcore sdk. Accessed on August 08, 2020. URL: `https://developers.google.com/ar`.

[49] PTC Inc. Vuforia. Accessed on June 02, 2020. URL: `https://developer.vuforia.com/`.

[50] Ltd. VisionStar Information Technology (Shanghai) Co. Easyar sense. Accessed on June 02, 2020. URL: `https://www.easyar.com/`.

[51] Wikitude GmbH. Wikitude. Accessed on June 02, 2020. URL: `https://www.wikitude.com/`.

[52] Tim Gleue and Patrick Dähne. Design and implementation of a mobile device for outdoor augmented reality in the archeoguide project. In *Proceedings of the 2001 Conference on*

Virtual Reality, Archeology, and Cultural Heritage, VAST '01, page 161–168, New York, NY, USA, 2001. Association for Computing Machinery.

[53] A. Haugstvedt and J. Krogstie. Mobile augmented reality for cultural heritage: A technology acceptance study. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 247–255, 2012.

[54] Claudio Baecchi, Tiberio Uricchio, Lorenzo Seidenari, and Alberto Del Bimbo. Outdoor object recognition for smart audio guides. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1247–1248, New York, NY, USA, 2017. Association for Computing Machinery.

[55] Google Maps Mirko Ranieri Product Manager. Live view. Accessed on October 12, 2020. URL: https://blog.google/products/maps/new-sense-direction-live-view/.

[56] Google. Places api google. Accessed on August 08, 2020. URL:https://developers.google.com/places/web-service/overview.

[57] Inc. Foursquare Labs. Places api foursquare. Accessed on August 08, 2020. URL: https://developer.foursquare.com/docs/places-api/.

[58] Google. Google maps sdk. Accessed on August 08, 2020. URL: https://developers.google.com/maps/documentation/android-sdk/overview.

[59] Google. Fused location provider api. Accessed on August 08, 2020. URL: https://developers.google.com/location-context/fused-location-provider.

[60] Intel. Intel context sensing sdk. Accessed on August 08, 2020. URL: https://web.archive.org/web/20170718015907/https://software.intel.com/en-us/documentation/context-sensing-sdk-for-android-states-datasheet.

[61] Square. Retrofit. Accessed on September 07, 2020. URL: https://square.github.io/retrofit/.

[62] Google. Gson. Accessed on September 07, 2020. URL: https://github.com/google/gson.