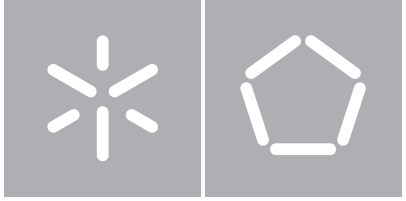
Universidade do Minho
Escola de Engenharia

Rui Emanuel Barros Oliveira

Mobile application to support pediatric medical practice

Outubro de 2013



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Rui Emanuel Barros Oliveira

Mobile application to support pediatric medical practice

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de
Professor Doutor António Carlos da Silva Abelha

Outubro de 2013

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, 30 / 10 / 2013

Assinatura: Rui Emanuel Barros Oliveira

Acknowledgments

I would like to express my gratitude to my supervisor, Dr. António Carlos Abelha, who created the right conditions for the success of this project, being always available to provide the needed help and establishing a contact with a pediatrician since the beginning of the project.

A special thanks to Dr. Simão Pedro Frutuoso, a pediatrician who closely followed the project. Since the beginning of the project, Dr. Simão has been tireless explaining pediatric related concepts and providing all the necessary data to develop the mobile application. The app has been tested by him in order to assure every functionality is working as expected and to make it as intuitive as possible.

I have to acknowledge Dr. Barros Oliveira, a family physician who kindly agreed to test the whole application to assure the app functionalities are correct and provided his suggestions in order to make the application as useful as possible.

I'd also like to thank my family and closest friends, for all the provided support and comprehension for all the times I couldn't be present.

Mobile application to support pediatric medical practice

Abstract

The objective of this project is to develop a mobile application in order to aid pediatricians performing their work. The necessity of this application was initially identified by a pediatrician working in Santo António Hospital of Oporto, after also verifying the interest of some of his coworkers. The bibliography also states some situations where mobile applications may be helpful, such as: errors in the administration of drugs or the difficulty pediatricians face in performing needed mathematical operations.

It is made a review of pediatric applications, mobile mostly, in order to know what kinds of applications are already available for pediatricians. It is presented the analysis of 5 distinct applications, from medical calculators for emergency situations to decision support systems that given a set of clinical characteristics it is provided a list of diagnosis to consider.

Following it is done a study of requirements elicitation and prioritization. Its objective is to know the techniques and tools already studied in the bibliography, as well as to identify the most appropriate ones for this project. Several elicitation and prioritization techniques were used in this project. It is also used a tool to register the requirements.

In order to develop a mobile application that may run on the majority of smartphones in the market, it is made an analysis of the smartphone operating systems market share, as well as of market share projections for the next few years. After identifying the target operating systems for the app it is made a study of the mobile cross-platform development frameworks. The framework choice considered the elicited requirements and the operating systems with the greatest market share.

After a learning period of the involved technologies, the pediatric app is developed using the gathered requirements and following the results of the requirements prioritization. The development of the application was always followed by a pediatrician, and as a result the application was tested and refined during that time. Finally, the application is released as well as a questionnaire to evaluate it.

Aplicação móvel como suporte à prática médica de Pediatria

Resumo

Este projecto tem o objectivo de criar uma aplicação para os dispositivos móveis que auxilie os médicos pediatras no exercício das suas funções. A necessidade desta aplicação foi inicialmente identificada por um médico pediatra que trabalha no Hospital de Santo António do Porto, após também verificar o interesse de alguns colegas de trabalho. A bibliografia também evidencia algumas situações em que as aplicações móveis podem dar o seu contributo, assim como: erros na administração de medicamentos ou dificuldades na realização de operações matemáticas necessárias.

É feita uma análise a aplicações móveis na sua maioria, vocacionadas para pediatria por forma a conhecer o trabalho já realizado nesta área. É apresentada a análise de 5 aplicações diferentes, que vão desde calculadoras médicas para situações de emergência até sistemas de suporte à decisão em que é apresentada uma lista de possíveis diagnósticos dado um conjunto de características do paciente.

De seguida é feito um estudo sobre a elicitação e priorização de requisitos. Este estudo teve por objectivo conhecer as técnicas e ferramentas já estudadas até ao momento, assim como identificar e aplicar as que melhor se adequam a este projecto. São aplicadas várias técnicas tanto de elicitação como de priorização de requisitos. É também utilizada uma ferramenta para o registo dos requisitos.

Para desenvolver uma aplicação móvel que atinja a grande maioria dos dispositivos móveis, é realizada uma análise ao market share dos sistemas operativos móveis, assim como a previsões para os próximos anos. Depois de identificados os sistemas operativos preferenciais para o desenvolvimento da aplicação é feito um estudo das frameworks de desenvolvimento de aplicações móveis multiplataforma. A escolha da framework teve em conta os requisitos adquiridos e os sistemas operativos com mais peso no mercado.

Passado um período de aprendizagem das tecnologias envolvidas neste projecto, é desenvolvida a aplicação em causa, utilizando os dados recolhidos na elicitação de requisitos e seguindo a ordem resultante da priorização dos requisitos. O desenvolvimento da aplicação foi sempre acompanhado por um médico pediatra, sendo a aplicação testada e refinada ao longo desse período. Por fim, a aplicação é lançada assim como um questionário que pretende avaliar a mesma.

Contents

List of acronyms and abbreviations

AIR	- Adobe Integrated Runtime
API	- Application Programming Interface
APP	- Application
AS3	- Action Script 3
BMI	- Body Mass Index
BMJ	- British Medical Journal
BSA	- Body Surface Area
CDSS	- Clinical Decision Support System
CRIB	- Clinical Risk Index for Babies
CRIB	- Clinical Risk for Babies
CSS	- Cascading Style Sheets
EHR	- Electronic Health Record
EMR	- Electronic Medical Record
ERB	- Embedded Ruby
FDA	- Food and Drug Administration
HIS	- Hospital Information System
HTML	- Hypertext Markup Language
HTTP	- Hypertext Transfer Protocol
IDC	- International Data Corporation
IDE	- Integrated Development Environment
IT	- Information Technology
LIS	- Laboratory Information System

MVC	- Model View Controller
NICU	- Neonatal Intensive Care Unit
NTISS	- Neonatal Therapeutic Intervention Scoring System
OS	- Operating System
OTC	- Over-The-Counter
PACS	- Picture Archiving and Communication Systems
PDF	- Portable Document Format
PELOD	- Pediatric Logistic Organ Dysfunction
PICU	- Pediatric Intensive Care Unit
PIM	- Pediatric Index of Mortality
PRISM	- Pediatric Risk of Mortality
SDK	- Software Development Kit
SNAP	- Score for Neonatal Acute Physiology
SNAPPE	- Score for Neonatal Acute Physiology Perinatal Extension
SNAPPE-II	- Score for Neonatal Acute Physiology Perinatal Extension II
SOAP	- Simple Object Access Protocol
SWF	- Shockwave Flash
TPN	- Total Parenteral Nutrition
TPN	- Total Parenteral Nutrition
VM	- Virtual Machine
WHO	- World Health Organization
XML	- Extensible Markup Language

List of Figures

1 Motivation

The pediatrics specialty is the field of medicine that deals with the health of newborns, infants, children and adolescents.

The development of the human body in early ages is done through very different phases. That´s why the World Health Organization created the growth curves of size, weight, body mass index, head circumference and blood pressure. It's difficult for pediatricians to know every "normal" value for every age. "We all know that one of the hardest things about dealing with very sick children is the wide variety of sizes they come in!" (Apps 2011a).

Every health care professional wants to provide the best health care possible to its patients. As the literature demonstrates, the medication errors in the pediatric specialty are frequent and adverse drug effects could be avoided (Kaushal et al. 2001). The collected data about the medication errors says that about 8% of all the medication errors correspond to the pediatric specialty. It is also estimated that 3% of all the hospitalized patients develop a severe reaction as a consequence of the medication administration during its treatment. Several studies also point out that pediatricians have difficulties in performing all the mathematical operations needed, which causes errors in medication doses. In some cases, the prescription or the administration error leaded to a 10 times higher or lower dosage than the accurate one. Errors as big as these are potentially dangerous for the patients, and if not intercepted in time may cause a temporary or permanent injury (Federal et al. 2011).

There are several factors that explain why professionals with high level of education are so prone to medication errors. In the pediatrics specialty, the doctors have patients in different development stages and very different weights, which have physical and physiological changes during time, making the permanent individual medication adjustment a necessity. In every pediatrics patients, the medication dosages are based on the patient's age, body weight, and sometimes the body surface. One problem that pediatric caregivers face is that a big percentage of some drugs, like the parenteral ones available, do not come with

a pediatric presentation. Calculations may be needed on several stages of the medication process, such as: the prescription and the preparation.

In some cases it is necessary to use drugs that are administrated in very small portions, for example: doses lower than 1 milligram for each kilogram of weight. Performing calculations with values as low as these and the necessity of performing units conversion (for example: milligrams to micrograms) make errors easier to happen (Federal et al. 2011).

A pediatrician has to watch several factors to ensure the normality of all the biological and physiological parameters during the caregiving practice, and needs to check a lot of tables and graphics. For example the percentile tables including the percentiles of weight, height, body mass index, percentage of fat mass and body surface values; as well as the values of normal blood pressure, hemoglobin and blood biochemistry.

The constant calculations and the numerous times pediatricians have to check tables, may delay the medical consultation but also makes human error more likely to happen in the overly execution of those tasks. The automation of these tasks available through a smartphone application aims to make the pediatricians life easier, saving their time and diminishing the human error risks during calculations.

Many medical applications for smartphones have been developed and widely used by health professionals and patients. The use of smartphones is getting more attention in healthcare day by day. Medical applications make smartphones useful tools in the practice of evidence-based medicine at the point of care (Mosa et al. 2012).

Manhattan Research, a health care consulting firm, estimated that the percentage of U.S. physicians using smartphones would be 81 per cent in 2012. Smartphone use in hospitals “is almost ubiquitous,” says Dr. Dante Morra of Toronto’s University Health Network. Most doctors are confident that having more access to information is good for patient care. “Medical practice has changed,” Dr. Phillip Yoon proclaims. It’s no longer about the physician and the patient; it’s “the physician, the patient, and an information technology device.” (Engelhart 2010).

2 Objectives

Together with a pediatrician working in Santo António Hospital of Oporto, it was identified the necessity of a mobile tool to support clinical practice and decision. This tool is going to aid pediatricians in several tasks they commonly perform, such as: the evaluation of the best practice in neonatal jaundice, anemia; aid with the technical issues of venous catheter or arterial umbilical insertions and converting the most commonly used units by pediatric caregivers.

The aim of this master thesis is to develop a mobile application that has the ability to aid pediatric caregivers in the decision or diagnosis to offer patients the best practice in the most common situations. It is also necessary to identify the different functionalities that would be helpful in the pediatricians smartphones, as well as to prioritize them. With the prioritization of the functionalities done, it is possible to identify how important each of these functionalities really is, and consequently select the functionalities that are going to be part of the mobile application.

The mobile application proposed in this thesis, presents itself as an alternative to the solutions today pediatricians are using, regarding a well-defined set of calculus, units conversions and decision support in some diagnosis which are common in pediatrics and are error prone. As a result, this application is an effort to simplify, quicken and reduce the error of the complex set of tasks pediatricians have to work with to make decisions.

One of the big advantages of this mobile application is that it is going to be designed and supervised in cooperation with pediatric physician. Also, this single application has calculation aid to perform techniques and unit conversion.

3 State of the Art

The use of computing tools to aid pediatric caregivers is not a recent theme, and was already introduced in the year of 1959 in an article advocating that computers might help in the diagnosis process (P Ramnarayan & J Britto 2002).

During recent years, healthcare professionals have required access to many technologies at the point of care, such as: Hospital Information Systems (HISs) including Electronic Health Record (EHR) or Electronic Medical Record (EMR) systems, Clinical Decision Support Systems (CDSSs), Picture Archiving and Communication Systems (PACSs), Laboratory Information Systems (LISs), evidence-based resources, clinical applications, drug databases, disease diagnosis applications and clinical communication (Mosa et al. 2012).

In this section, several applications that are related with the pediatric specialty will be discussed. The main set of applications is for mobile devices, with the exception of Sabichão. It is a computer application that was developed by Dr. Simão, graduate pediatric assistant of Centro Hospitalar do Porto, that is cooperating with this work, and has a lot of features that aid pediatricians in a various set of tasks.

The applications that are going to be discussed are: Isabel, Paeds ED, DrugDoses, Epocrates and Sabichão. These applications have different purposes, such as: Disease diagnosis, clinical decision support, drug reference and medical calculators. Below there is a brief explanation of the application categories as well as where each of the discussed applications fit.

Clinical decision support systems are computing tools that use specific knowledge to generate patient specific advices or interpretations. The existence of CDSS's is a result of the information overload available nowadays (P Ramnarayan & J Britto 2002). Isabel, Epocrates ID and some Sabichão tools also fit in this category.

Drug reference applications generally include the names of drugs, their indications, dosages, pharmacology, drug-drug interactions, contraindications, cost, and identifying characteristics (Mosa et al. 2012). The drug reference discussed here is the Epocrates application.

A medical calculator or clinical calculator is a software program for calculating various clinical scores and indices such as body mass index (BMI), body surface area (BSA), coronary heart disease risk, individual drug dosing, etc. Usually calculation of clinical scores or indices involves complex formulas using several input parameters (Mosa et al. 2012). Sabichão has several medical calculators that are going to be discussed further.

3.1 Isabel

The Isabel system, product of the Isabel Medical Charity, had the initial aim to provide free access to a decision support system for pediatric caregiving, available on the web (P Ramnarayan & J Britto 2002). Nowadays, this system is no longer free and is not exclusive to the medical specialty of pediatrics, but also for the adult population and it is available at (http://www.isabelhealthcare.com).

Figure 2.1 – Image showing the Isabel interface for searching and viewing results.

Given a set of clinical characteristics, the system provides a list of diagnostics to consider. Each one of the presented diagnosis provide related texts, images, related guidelines, as well as a dedicated section to past experiences with the obtained diagnosis and results (P Ramnarayan & J Britto 2002). This CDSS also provides access to journals, external web resources and protocols for each diagnosis. When a search for the patient symptoms is made, it is shown a list of drugs that may be the cause for the problem.

The system has a dedicated part for data entry, where it is possible to fill with patient related data, such as: age, gender or travel history. The user also has to choose the kind of information that is pretended, which might be diagnosis, causative drugs or bioterrorist agents. Finally, the user has a field to insert clinical features, like symptoms or signs. After the search is made it is provided a list of 10 diagnoses to consider, presented by relevance. It is also possible to change the order results shown, and also filter only the results of a particular specialty. The system signs with red flags every diagnosis that might be harmful for the patient if not considered by the pediatrician (Isabelhealthcare 2012a).

A study with Isabel that involved 594 patients in emergency situation tested the system with a set of diagnoses chosen by an expert panel. The system returned diagnosis results in 95% of inpatients and 90% of "must not miss" diagnoses. In 78% of the cases the correct diagnosis appeared in the first 10 suggestions (Padmanabhan Ramnarayan et al. 2007).

Several studies demonstrate the usefulness of the Isabel system in the clinical practice, however most of them are only focused in the adult population. The usefulness of these systems in pediatrics is still to be studied by the medicine literature (Manicone et al. 2011).

If wanted, Isabel may be integrated with EMR systems (Electronic Medical Record), which saves the time and effort of inserting the clinical data related to the patient. When this integration is used, the data is extracted and inserted in the system automatically (Isabelhealthcare 2012b).

There is also an Isabel application for mobile devices, such as: iPhones, iPads, and Androids. The mobile application is free to download, however it uses the online system to deliver information. To access the system it is necessary to pay a fee in one of the plans available in the website.

Figure 2.2 – Isabel mobile application for iPhone.

3.2 Sabichão

Sabichão is an application developed by a pediatrician (Dr. Simão Frutuoso), whose interest for both medicine and informatics led him to develop it using Visual Basic for Applications and Microsoft Excel 2007. The program intends to respond to several professional needs of the pediatric caregivers, which were identified by him after several years of experience in the field.

The final result is a set of tools available for all physicians that treat pediatric patients, mostly in a hospital setting.

These tools are organized in several modules that work independently.

One of the most complete and sophisticated module is the Total Parenteral Nutrition (TPN) module. It is possible to run three versions of this module, inside the Sabichão, each one fully adapted to the needs of

the three main departments of the Centro Hospitalar do Porto that uses the application: the Pediatric Intensive Care Unit (PICU), the Neonatal Intensive Care Unit (NICU) and the Pediatric Ward.

TPN is used to give total intravenous nutrition, vitamins and trace elements to critically ill neonates and children that cannot be fed orally. It's an advance procedure usually performed only in an ICU setting.

This module has the following features:

- Automatic calculations of quantities of each component, minimizing human error.
- Display warnings and alerts related to composition, doses, rhythm, and incompatibilities.
- Automatic insertion of data from the protocol in use (only the NICU version)
- Saving of up to 1000 prescription sheets so they can be reedited later.
- Prescription of sheets for several days in advance.
- Automatic prescription of daily sheets for any number of days with just one mouse click (only Pediatric Ward version)
- Print the prescription sheet
- Automatic creation of labels for TPN (Total Parenteral Nutrition) bags, with:
    - Bag composition
    - Volume and rhythm of infusion
    - Period of validity, according to the composition.
    - Two labels, one for each type of bag (protein and carbohydrate bag and the lipid bag)
- Sabichão automatically sends the prescription sheet and both the labels to the Pharmaceutics Department where the bags are elaborated using a PDF exporting feature.

Another module is the Emergency sheet: It intends to give the correct doses of drugs and other information for emergency situations. As in the TPN module, there are also two versions inside Sabichão, one for neonates and the other for older children.

These sheets are to be filled when the patient is admitted, using only the patient name, age, weight, and height, and made readily available. In an emergency situation, the information contained in it can be lifesaving.

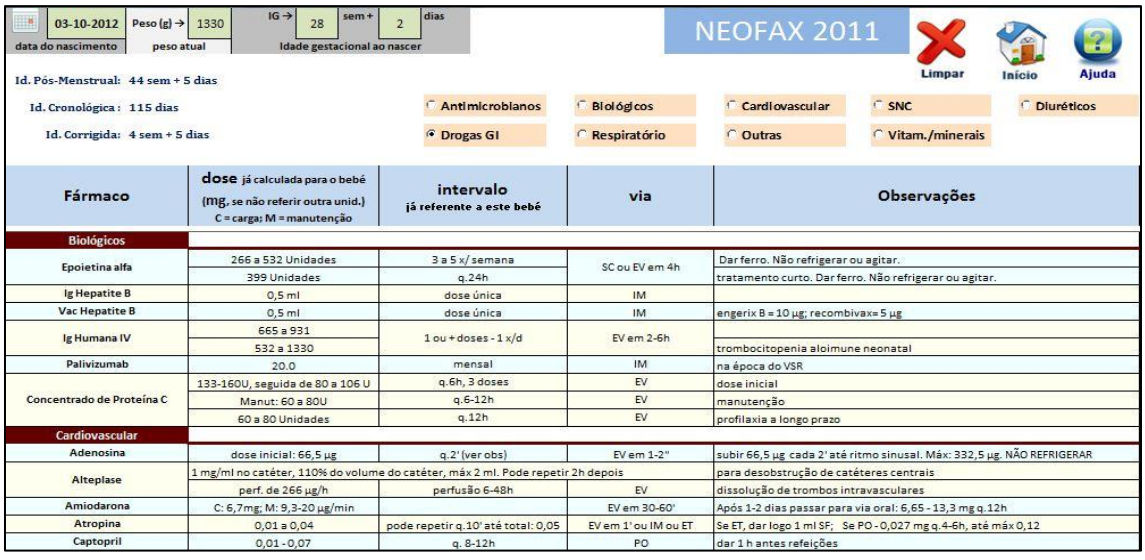
These sheets contain:

- Emergency drugs doses, already calculated for this particular patient
- Dose for defibrillation shock
- Percentiles of blood pressure, including a table with normal blood pressure for the first week of live, with ±1 sd values
- Measures for insertion of tracheal tube (by nose or mouth)
- Measure or body surface (square meters)
- Measure of Body Mass Index

Another group of modules were conceived to help the physician prescribing drug perfusions, according to the concentration of the drug and the dose pretended.

This group of modules includes drugs such as dopamine, dobutamine, epinephrine, norepinephrine, midazolam, fentanyl, morphine and vecuronium.

Related to this module there is the Neofax module, this being more complete than the formers but



Fármaco	dose já calculada para o bebé (mg, se não referir outra unid.) C = carga; M = manutenção	intervalo já referente a este bebé	via	Observações
Biológicos				
Epoietina alfa	266 a 532 Unidades	3 a 5 x/ semana	SC ou EV em 4h	Dar ferro. Não refrigerar ou agitar.
	399 Unidades	q.24h		tratamento curto. Dar ferro. Não refrigerar ou agitar.
Ig Hepatite B	0,5 ml	dose única	IM	
Vac Hepatite B	0,5 ml	dose única	IM	engerix B = 10 µg; recombivax= 5 µg
Ig Humana IV	665 a 931	1 ou + doses - 1 x/d	EV em 2-6h	
	532 a 1330			trombocitopenia aloimune neonatal
Palivizumab	20.0	mensal	IM	na época do VSR
Concentrado de Proteína C	133-160U, seguida de 80 a 106 U	q.6h, 3 doses	EV	dose inicial
	Manut: 60 a 80U	q.6-12h	EV	manutenção
	60 a 80 Unidades	q.12h	EV	profilaxia a longo prazo
Cardiovascular				
Adenosina	dose inicial: 66,5 µg	q.2' (ver obs)	EV em 1-2"	subir 66,5 µg cada 2' até ritmo sinusal. Máx: 332,5 µg. NÃO REFRIGERAR
Alteplase	1 mg/ml no catéter, 110% do volume do catéter, máx 2 ml. Pode repetir 2h depois			para desobstrução de catéteres centrais
	perf. de 266 µg/h	perfusão 6-48h	EV	dissolução de trombos intravasculares
Amiodarona	C: 6,7mg; M: 9,3-20 µg/min		EV em 30-60'	Após 1-2 dias passar para via oral: 6,65 - 13,3 mg q.12h
Atropina	0,01 a 0,04	pode repetir q.10' até total: 0,05	EV em 1' ou IM ou ET	Se ET, dar logo 1 ml SF; Se PO - 0,027 mg q.4-6h, até máx 0,12
Captopril	0,01 - 0,07	q. 8-12h	PO	dar 1 h antes refeições

Figure 2.3 – Image showing several biologic and cardiovascular drugs info, by the NEOFAX module.

intended only for neonates. This module is based on the well-known book with the same name. Sabichão uses the 2011 version of Neofax.

Once introduced the date of birth, weight and gestational age of the baby, Sabichão gives back with only a mouse click, all the drugs used in a NICU setting, already calculated to this particular baby, doses, scheduling, warnings, route of administration, and so on.

In the next group of modules, Sabichão offers several scales for evaluating many things, like the Glasgow Coma Scale, scores for evaluating the risk of mortality and morbidity (Pediatric Risk of Mortality (PRISM), Pediatric Index of Mortality (PIM), Pediatric Logistic Organ Dysfunction (PELOD), Clinical Risk Index for Babies (CRIB), Score for Neonatal Acute Physiology – perinatal extension (SNAPPE), Neonatal Therapeutic Intervention Scoring System (NTISS). All this "scale" modules are easy to use.

The Glasgow Coma Scale is used to measure the level of awareness of a patient. A set of option buttons is used to insert the conditions that apply to the patient.

These conditions are related with the eyes opening, verbal and motor response of the patient. A patient with a score 15 is fully alert, while one with a score of 3 is in deep coma.

All the other scores are used to evaluate the severity of the patient's illness and the odds of dying. Some of them use signs and symptoms, others use laboratory values, and one (NTISS) measures the severity of illness of the neonate by the amount of medical intervention the baby needs.

All these scores require the insertion of data by means of option buttons, quite handy to use, and only a few seconds are required to fill in a form. Then Sabichão returns the answer, sometimes using advance logarithmic calculations, other times just by sampling adding the punctuation.

Pediatric Logistic Organ Dysfunction (PELOD) is used to measure the performance of organic systems in a multiorgan failure situation. The pediatrician is asked to fill parameters about the respiratory, cardiovascular, neurological, hepatic, renal and hematologic systems. After finishing the questionnaire, it is generated a final score and a death probability.

The Clinical Risk for Babies (CRIB) score is a tool for assessing initial neonatal risks and comparing performance of neonatal intensive care units (Network 1993). The tools for the first and second version of CRIB have six and five fields to fill respectively, in which the values of the scoring system are displayed while the user is filling the form. When the form is completed the CRIB score is displayed.

There's also available a tool for the Score for Neonatal Acute Physiology (SNAP) and the Score for Neonatal Acute Physiology Perinatal Extension II (SNAPPE-II).

Both of these scoring systems are a simplified newborn illness severity and mortality risk scores (Douglas K. Richardson, John D. Corcoran, Gabriel J. Escobar 2001). These systems have several parameters in common and as a consequence when these parameters are inserted the tool uses them for the calculation of both scoring systems.

The last tool calculates the Neonatal Therapeutic Intervention Scoring System (NTISS), which is a therapy based severity of illness index (James E. Gray, Douglas K. Richardson, Marie C. McCormick, Kathryn Workman-Daniels 1992). The tool provides a set of yes or no parameters the pediatrician has to answer. These parameters have an associated value that is displayed once the parameter is selected, and the final score is updated while the form is being completed.

The next group of modules is related to anthropometric measures and centiles. This group is suitable for use, not only in a hospital setting, but also in ambulatory practice.

The main anthropometric parameters include weight, height, cephalic perimeter, body mass index. After inserting the values and the age/sex of patient, it displays the respective centiles, both for newborn babies and for older people, up to 17 years old.

Premature babies are also contemplated. They have a form module that gives weight centiles from 22 up to 44 weeks gestational age.

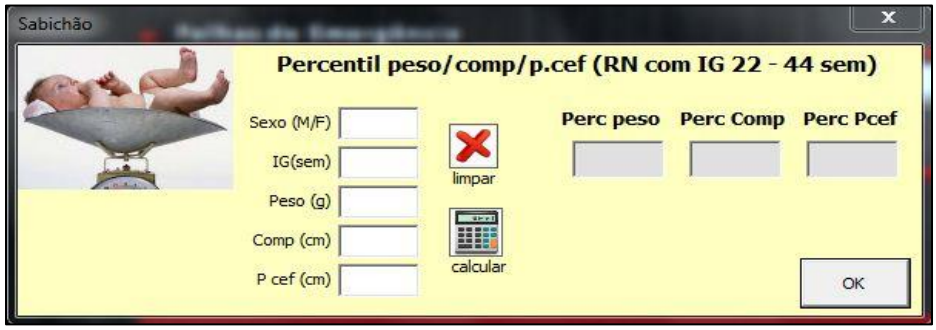


Figure 2.4 - Medical calculator for the weight, length percentiles and cephalic perimeter.

In one module of this group, Sabichão reveals the predicted adult height of a child (with a 95% confidence interval) based on his gender and his parents height.

Another module provides the body fat percentage, requiring only a few input data.

Still in this group there are other modules: for determination of Body Mass Index, Body surface area (according to several known formulas) and Rohrer ponderal index.

The last section available has tools for several purposes, some of them helping in the diagnose, while others are intended to aid pediatrician performing medical procedures or even both scenarios. Hence there are tools related with: blood pressure, unit's conversion, renal function, umbilical arterial and venous catheters, jaundice, acid-base equilibrium, energy needs, date calculations, traumatic lumbar puncture and blood transfusion in newborns.

The blood pressure tool is intended for patients until 17 years old, and its function is to check the tables of arterial hypertension instead of the pediatrician. Given the age, height and gender of a patient, it provides the values of the 50^{th}, 90^{th}, 95^{th}, 99^{th} percentiles for systolic, diastolic and medium arterial tension.

In the units conversion tool there are three convertors available. The first one allows the user to convert any amount of several elements (for instance: sodium or magnesium), from one concentration unit (ex: mg/dl) to another (ex:mmol/L or mEq/L). The second convertor gives the user the possibility to select one molecule and shows the default conversion ratio to the most common used units. In the third converter the user may insert a value and see it converted between several units, such as: inches and centimeters, ounces and grams or degrees Celsius and degrees Fahrenheit.

The renal function tool aims to help the pediatrician to perform renal function calculations. After filling the necessary information, this tool provides information about: fractional excretion of sodium, estimated creatinine clearance, glomerular filtration rate, rate of phosphate reabsorption, renal insufficiency index, anion gap and plasma osmolarity.

Another available tool is related with the umbilical catheterization. It helps the pediatrician in the insertion of the umbilical catheters (both venous and arterial). It uses a figure of a baby with its internal organs depicted and two horizontal bars, one that represents all length of the umbilical artery (the other for the umbilical vein). Sliding the horizontal bar makes a black point move along the respective vessel in the figure, revealing the point where the tip of the catheter would stay, if it was inserted until the deep showed in the bar.

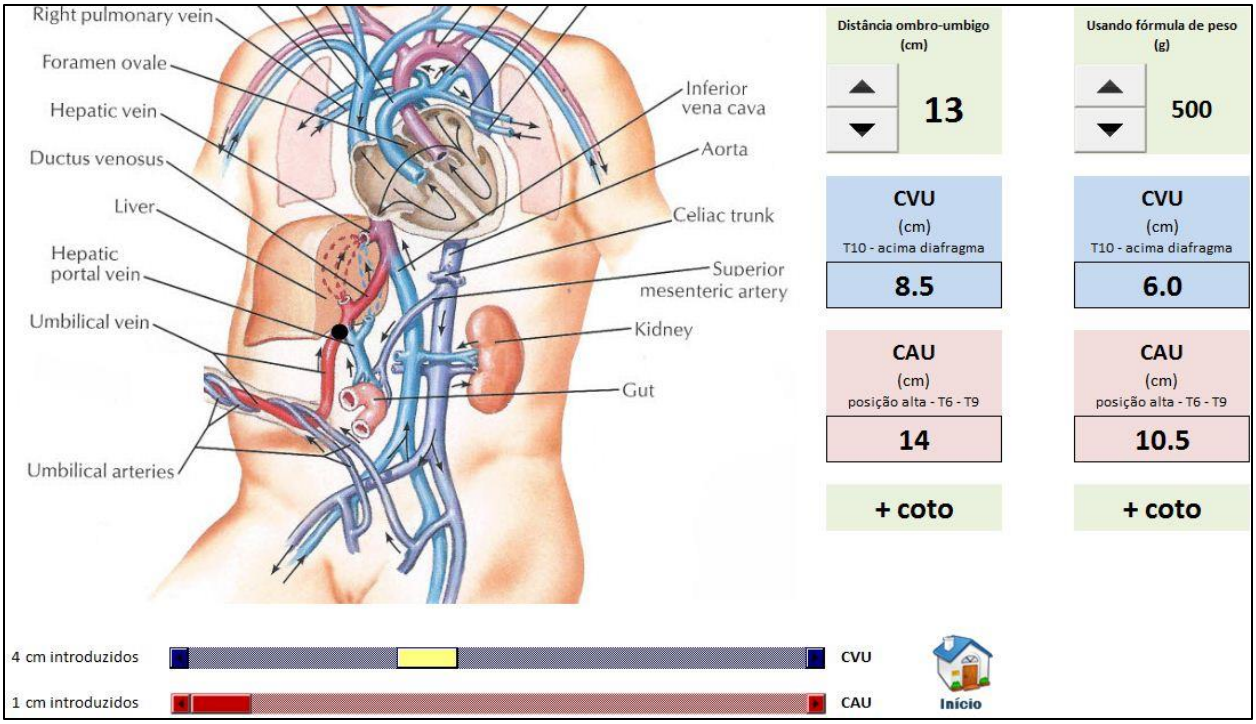


Figure 2.5 - Catheterization tool showing where the tip of umbilical venous catheter is, after inserting it 4 centimeters deep.

The jaundice tool helps in the diagnosis and treatment decision making. After introducing the bilirubin levels it calculates if any treatment is needed, for instance: phototherapy or exchange transfusion.

The acid-base tool helps the physician in the interpretation of the patient blood gases, mostly the respiratory and metabolic components of the acid-base equilibrium. It informs, for instance, if the patient is acidotic and if that acidosis is from respiratory or metabolic origin. It presents the results in two tables (with different entrances) for a more clear view. It's a great tool for teaching medical students too, once it allows experimentation with fictitious scenarios.

The date calculator tool shows the calendar of any year, since 1900 and also helps the pediatrician to easily calculate the corrected age and postmenstrual age for premature babies. It is also possible to calculate the number of years, months, weeks and days between two dates.

Another module allows the pediatrician to correctly evaluate the results of a traumatic lumbar puncture, indicating if the leukocytes measured are due to meningitis or resulted from an inadvertent punctured vessel.

The blood transfusion tool, that follows the Portuguese guidelines, helps in the decision of whether or not, a newborn should be transfused, and if so, it provides the amount and rhythm of the transfusion.

This application is freely available in the Portuguese neonatologists forum (http://lusoneo.portugueseforum.net/).

3.3 Paeds ED

Paeds ED is a pediatrics specific mobile app, currently only available for Apple iPhone/iPad devices. To reduce the number of emergency setting errors, Samiei and two colleagues designed the Paeds ED app, which allows docs to accurately predict a child's weight, review available drugs, and calculate doses with precision—quickly ("within three or four finger swipes") (Engelhart 2010).

The app initially asks for the patient's gender and 'known weight' or 'known age' (Bhansali & Armstrong 2012). If the weight is unknown it is calculated based on the 'guesstimate' formula, based on age in years, and then derive the correct doses of various drugs either from memory or formularies. Paeds ED converts

the age of the child in weeks, months, or years (including premature babies) into weight by referencing to WHO (World Health Organization) weight charts.

After that, the app displays 14 categories: Key Resuscitation Data, Anaphylaxis, Convulsion, Bronchospasm, Inotropic Support, Septicaemia, Anesthesia, Sedation, Fluids – Bolus, Fluid – Maintenance, Fluid – Rehydration, Fluid - Burns resuscitation, Pain Ladder and Drug formulary. All of the above sections are possible emergency situations a pediatrician may run into (Apps 2011a).

Figure 2.6 – Paeds ED mobile application for iPhone.

The fluids section provides maintenance requirements (based on 4-2-1 rule) and detailed information on bolus administration, management of burns and dehydration (Bhansali & Armstrong 2012).

Once selected each of these modules provides an aggregate of drugs & dosages or equipment sizes potentially required to resuscitate that child (Apps 2011a). All the drugs available in each of the categories can also be found in the formulary. It lists the drugs alphabetically with the correspondent dosages calculated, and includes 'quick links' to each letter (Bhansali & Armstrong 2012).

There is a free version of this app, called Paeds ED lite. The difference between this version and the paid one is that this has less emergency situations and 'drugs & equipment' available. It has only 2 emergency situations and 11 'Drugs & Equipment' available (Apps 2011b).

3.4 DrugDoses

The DrugDoses app is mainly a drug reference application. It is a mobile version of the Frank Shann's booklet, which contains more than 2000 drug dosages for both children and adults (Mosa et al. 2012). This application has 5 sections for different purposes. These sections are: Drugs, Code Blue, PedCalc, Laboratories and Cardio.

The Drugs section is an alphabetically ordered drug list, with quick links to each first letter and has the ability to select 'Favorites' (Bhansali & Armstrong 2012). It lists every drug commonly prescribed for systemic use in both adults and children. The iPhone version of the booklet provides an enormous amount of information, which is updated every 6 months. The final user will always have the latest drugs, which will only be available in the paper version in a year or two (D. O. Karam n.d.).

The Code Blue calculates properly the doses of several drugs commonly used in code blue situations. For example: Adrenaline, Fentanyl, Ketamine and Adenosine. The user is asked for the patient weight and then the doses are calculated and shown (D. O. Karam n.d.). However, this section contains no data on defibrillation or fluid administration (Bhansali & Armstrong 2012).

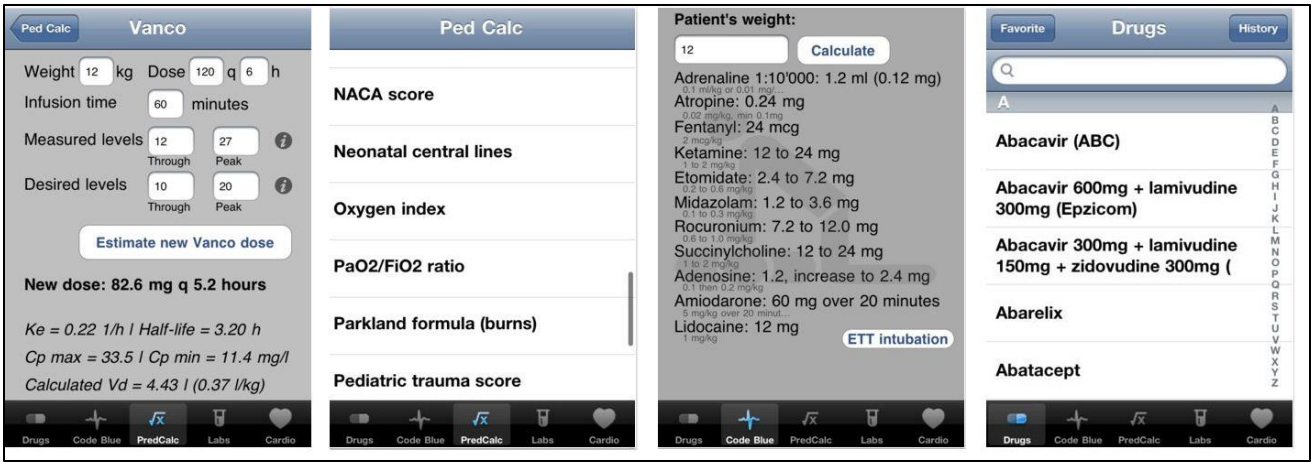


Figure 2.7 – DrugDoses mobile application for iPhone.

PedCalc is a Pediatric calculator that contains the most common scoring systems and formulas, such as: Glasgow Coma Score, Anion Gap, Body Mass Index, Maintenance fluids, Pediatric Trauma Score and Predicted Height.

The Laboratories section provides more than 150 current laboratory values. Lastly, the Cardio section provides useful cardiology information.

There's an inApp purchase that can be made to have access to the Frank Shann's Pediatric Critical Care Guidelines. It contains all the latest guidelines available for pediatric critical care, ranging from Analgesia & Sedation to Ventilation. It also provides a search field for a quicker access to its content (O. Karam 2012).

This app is currently available for iOS and Android platforms. The Windows Mobile and Palm OS also had a version of DrugDoses, but they're currently discontinued (D. O. Karam n.d.). One drawback is that it is the most expensive application of all the apps covered in this document. Once the user buys the product it will be given an initial 2 year license including all the updates. After that period, the user may buy a lifetime license (O. Karam 2012).

3.5 Epocrates RX and Essentials

Epocrates Inc. provides a set of medical applications for a wide variety of purposes. The company claims to have more than a million active members including 50% of U.S. physicians (Epocrates 2013a). In this master thesis only Epocrates RX and Epocrates Essentials applications are going to be discussed.

Epocrates RX is a drug reference application available for free. It was cited as the most commonly accessed drug-reference application (Mosa et al. 2012). The application provides the possibility of searching brand, generic and OTC (Over-the-Counter) medicines. It has a customizable homepage so the user can organize it the way he wants.

The drug information section has the following characteristics: adult and pediatric dosing for FDA (Food and Drug Administration) approved and off-label indications, black box warnings, adverse reactions, U.S.

healthcare insurance formularies, manufacturers contacts, approximate retail drugs full price, among others (Inc. 2013).

The drug-drug interaction-checking feature of drug reference applications like Epocrates is a very useful evidence-based resource at the point of care (Mosa et al. 2012). It is possible to check for potentially harmful interactions between a patient's multiple meds up to 30 drugs at a time. These interactions are organized by categories.

There's available a mobile Sample closet section, allowing the user to order free literature and drug samples from some pharmaceutical companies or healthcare organizations. It is only necessary to fill the request to get the desired items (Inc. 2013).

It has a section to identify a pill by indicating physical characteristics or imprint code. By selecting available characteristics it is possible to view drug names and images.

The App Directory section provides a wide variety of free and paid applications ranging from dosing tools to imaging atlases.

There is also available a mobile resources center that offers clinicians useful, up-to-date medical news and scientific abstracts, selected by a contributing editor (Inc. 2013).

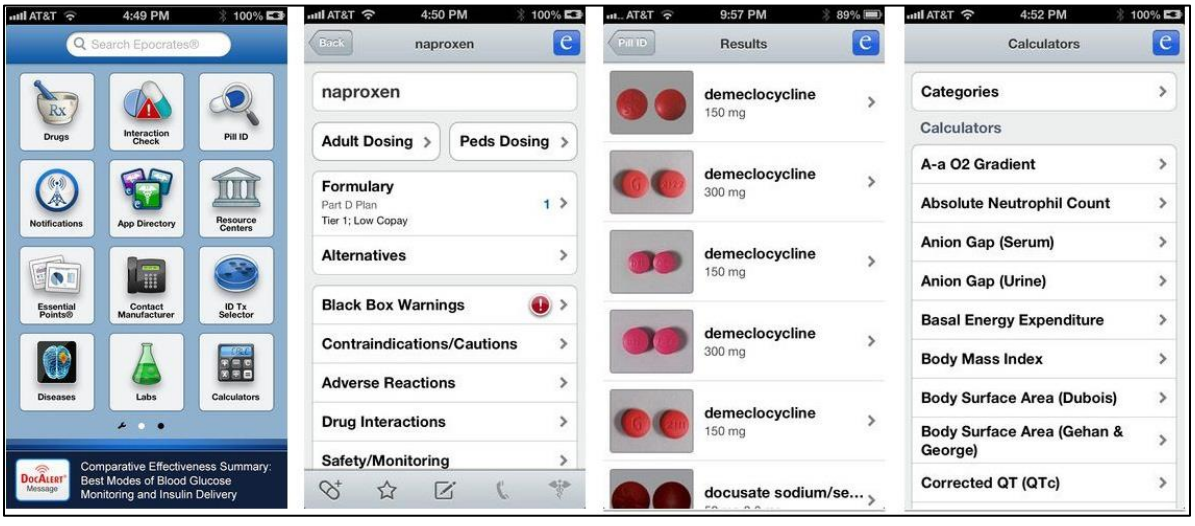


Figure 2.8 – Epocrates RX mobile application for iPhone.

Epocrates Essentials is an annually charged application that offers all the functionalities the Epocrates RX has, and also includes features related with: diagnosis and diseases.

The disease information section provides content developed in collaboration with the BMJ Group, such as: differential diagnosis with links to respective disease and condition topics, evidence-based treatment options searchable by patient group compiled from clinical references, disease state management guidelines and test categories including initial tests, tests to consider and emerging tests.

Another section is the Diagnostic and Lab tests section, which provides hundreds of lab tests and panels providing: integrated drug and disease information and follow-up recommendations, approximate tests costs, reference ranges, preparation and collection guides, among others (Epocrates 2013c).

There is also available information about alternative medicine.

EPocrates ID (Infectious Disease) is available in the Epocrates Essentials and has the most current specialty guidelines and primary literature resumed, providing a much smaller and comprehensive set of treatment recommendations for common and uncommon infections (Epocrates 2013b). A 2004 study evaluated ePocrates ID for treatment recommendations on 202 cases and reported that it provided treatment recommendations in every case (Mosa et al. 2012).

These applications are available for the iOS, Android and BlackBerry platforms.

4 Requirements Engineering

In this chapter, it is going to be addressed a set of features that are considered to be helpful in the development of application software. In order to develop the pediatrics mobile application, it was used the requirements elicitation and prioritization processes. Both processes are related with a wider subject called Requirements Engineering.

Requirements engineering on the project level is the process by which the requirements for a software project are gathered, documented and managed throughout the software development lifecycle. Software requirements are the critical determinants of software quality, given empirical studies showing that errors in requirements are the most numerous in the software life-cycle and also the most expensive and time-consuming to correct (Aybüke Aurum 2005).

A survey conducted with 350 USA organizations revealed that about half of the managers interviewed identified poor requirements as a major source of problems, along with other factors such as low user involvement and unclear objectives. In another survey involving twelve UK companies, requirements problems accounted for 48% of all software problems. In one healthcare related case study, it was observed that there was a huge gap between the daily operations of a hospital and software developer's domain knowledge of these operations, though every year healthcare organizations spend large amounts of money and resources on IT systems. The authors of the study (Tveito and Hasvold) argue that this gap is due to insufficient requirements gathering and misunderstanding requirements due to the lack of domain knowledge.

In order to deliver high quality software systems on time and on budget it is essential to have properly structured and controlled requirements specifications that are understandable, comprehensive and consistent. It is important to have a good understanding of stakeholder goals and ensure their involvement in the requirements engineering process (Aybüke Aurum 2005).

In the development of the pediatrics mobile application, it is going to be addressed a set of issues related to the three parts of requirements engineering, which as above mentioned are: the gathering,

documentation and management of requirements. The requirements gathering used a set of elicitation techniques that are presented in detail in section 3.1. The documentation of the requirements was done using a requirements specification template, named Volere (section 3.2). Requirements management is a discipline that involves multiple concepts, such as: requirements modeling, prioritization, negotiation, quality assurance, among others. In this work, it is only going to be studied the requirements prioritization, presented in section 3.3.

4.1 Requirements elicitation

Requirements elicitation is the process of seeking, uncovering, acquiring, and elaborating requirements for computer based systems. It is a complex process involving many activities with a variety of available techniques, approaches, and tools for performing them. The relative strengths and weaknesses of these determine when each is appropriate depending on the context and situation.

The elicitation of requirements begins in an initial stage of a project and is considered to be a critical part of the development process of a software system. These requirements may be found in a variety of sources, such as: stakeholders, documentation and other existing systems. Eliciting the right requirements is considered a vital but difficult part of software development projects (Aybüke Aurum 2005).

In reality requirements elicitation is an activity that relies heavily on the communication skills of requirements engineers and the commitment and cooperation of the system stakeholders. One of the major problems that were identified in the software development project teams is the communication barriers and agreement about the requirements. There are several elicitation techniques available to use. The choice of a elicitation technique(s) for a specific project depends on a variety of factors including time and cost, the availability of resources, the safety criticality of the system, and any legal or regulatory constraints.

There isn't yet a consensus in a standard definition of requirements elicitation. Requirements elicitation is concerned with learning and understanding the needs of users and project sponsors with the ultimate aim

of communicating these needs to the system developers. A substantial part of elicitation is dedicated to uncovering, extracting, and surfacing the wants of the potential stakeholders. It is a process, in which a set of requirements is gathered. After the elicitation of the requirements there are other techniques to filter the requirements, such as: requirements prioritization and requirements negotiation (Aybüke Aurum 2005).

The requirements elicitation process has several typical activities, such as: Understanding the application domain, identifying the sources of requirements, analyzing the stakeholders and selecting the Techniques to use and eliciting the requirements from stakeholders and other sources.

To understand the Application Domain it is important to investigate and examine in detail the situation in which the system will reside. It is necessary to explore the current environment, so it is possible to identify any existing constraints.

Requirements may be spread across many sources and exist in a variety of formats. In all software development projects a number of possible sources for requirements may be identified. Stakeholders represent the most obvious source of requirements for the system. Other sources of information may be: subject matter experts, existing systems and existing documentation (Aybüke Aurum 2005).

It is also necessary to analyze and find out the range of stakeholders of the project. Stakeholders are people who have an interest in the system or are affected in some way by the development and implementation of the system. The customer, and more specifically the project sponsor, is usually the most apparent stakeholder of the system. In some cases however the actual users of the system may be the most important.

It is generally accepted that an individual requirements elicitation technique or approach cannot possibly be suitable for all projects. The choice of techniques to be employed is dependent on the specific context of the project and is often a critical factor in the success of the elicitation process. A study regarding requirements elicitation reveals that the elicitation technique may be selected for a variety of reasons, such as: the technique selected is the only one the analyst knows, the technique selected is the analyst's favorite, the technique selected is the one that the analyst considers to be the most appropriate, among others. Clearly requirements elicitation is best performed using a variety of techniques. In the

majority of projects several methods are employed during and at different stages in the software development life cycle, often in cooperation where complementary (Aybüke Aurum 2005).

After the identification of the requirements sources and specific stakeholders, the actual elicitation of the core requirements then begins using the selected elicitation techniques, approaches, and tools. During this activity it is important to investigate in detail the needs and wants of the stakeholders, especially the users, as well as to determine the future processes the system will perform with respect to the business operations.

The requirements elicitation process is always limited by several aspects, such as: specific characteristics of the project, organization and environment, the budget or the schedule are some of the variables that directly affect the requirements elicitation. In reality its completion is often determined by time and cost constraints rather than achieving the required level of requirements quality and completeness. Typically the process begins with an informal and incomplete high-level mission statement for the project. This may be represented by a set of fundamental goals, functions, and constraints for the target system, or as an explanation of the problems to be solved. The result of this process forms the basis of further investigation and refinement of requirements in a typically iterative and incremental manner. At the end of the requirements elicitation, there normally is a set of detailed requirements in natural language text and simple diagrammatic representations with additional information including descriptions of the sources, priorities and rationales (Aybüke Aurum 2005).

Over the years a number of process models have been proposed for requirements elicitation. For the most part these models provide only a generic roadmap of the process with sufficient flexibility to accommodate the basic contextual differences of individual projects.

Frequently requirements engineers are responsible for documenting the requirements elicited. This role is particularly important as it represents the production of results from the elicitation process, and forms the foundation for the subsequent project phases. Evaluation of the elicitation process and the work performed by the analyst is based on these resultant artifacts, which in some cases may form the basis of contractual agreements (Aybüke Aurum 2005).

4.1.1 Requirements elicitation techniques

Following it is presented a set of techniques used for requirements elicitation nowadays. These techniques are only a small portion of all the available techniques for this purpose; however the techniques studied in this document are some of the most commonly used.

4.1.1.1 Interviewing

When interviewing, the requirements engineering team puts questions to stakeholders about the system that they currently use and the system to be developed. Requirements are derived from the answers to these questions. Most requirements engineering processes use this technique, which can take two forms: formal and informal. The formal ones are closed interviews, where the stakeholder is asked a pre-defined set of questions. On the other hand, informal interviews are made of a set of open questions. The requirements engineering team explores a range of issues with system stakeholders and hence develop a better understanding of their needs (Sommerville 2010).

In practice, interviews with stakeholders are normally a mixture of both of these. You may have to obtain the answer to certain questions but these usually lead on to other issues that are discussed in a less structured way. Completely open-ended discussions rarely work well.

Interviews are good for getting an overall understanding of what stakeholders do, how they might interact with the new system, and the difficulties that they face with current systems. However, interviews are not so helpful in understanding the requirements from the application domain. Mainly, the difficulty in eliciting domain knowledge with interviews is due to a couple of reasons:

- All application specialists use terminology and jargon that are specific to a domain. It is impossible for them to discuss domain requirements without using this terminology. They normally use terminology in a precise and subtle way that is easy for requirements engineers to misunderstand.
- Some domain knowledge is so familiar to stakeholders that they either find it difficult to explain or they think it is so fundamental that it isn't worth mentioning.

In order to be effective eliciting requirements with the interviewing technique, the requirements engineers must: be open-minded, avoid pre-conceived ideas about the requirements and be willing to listen to stakeholders; prompt the interviewee to get discussions going using a springboard question, a requirements proposal, or by working together on a prototype system (Sommerville 2010).

Interviewing on its own is liable to miss essential information and so it should be used in conjunction with other requirements elicitation techniques.

4.1.1.2 Domain analysis

Domain analysis consists in the examination of the existing and related documentation and applications. It is a very useful way of gathering early requirements as well as understanding and capturing domain knowledge, and identifying reusable concepts and components. These types of investigations are particularly important when the project involves the replacement or enhancement of an existing legacy system.

Some examples of documentation for domain analysis are design documents, instruction manuals, hardcopy forms and files used in the business process; while the application studies often include looking at both upstream and downstream systems, as well as competitive or like solutions (Aybüke Aurum 2005).

Analogies and abstractions of existing problem domains can be used as baselines to acquire specific and detailed information, identify and describe possible solution systems, and assist in creating a common understanding between the analyst and stakeholders.

This elicitation technique and its results are often used in conjunction with, and as the input to other elicitation techniques (Aybüke Aurum 2005).

4.1.1.3 Group Work

Group work is a very common technique for requirements elicitation. Groups are particularly effective because they involve and commit the stakeholders directly and promote cooperation. When the number of stakeholders is big, the sessions are difficult to organize. It is necessary to be aware and prevent situations where some individuals dominate the discussion. It is also important that stakeholder feel comfortable and confident in speaking openly and honestly (Aybüke Aurum 2005).

4.1.1.4 Ethnography

Software systems are used in certain social and organizational contexts, and the system requirements may be derived or constrained by that context. These requirements are often critical for the success of the system. One reason why many software systems are delivered but never used is that their requirements do not take proper account of how the social and organizational context affects the practical operation of the system.

Ethnography is an observational technique that can be used to understand operational processes and help derive support requirements for these processes. The requirements engineer must be in the working environment were the system will be used, with the objective of observing and taking notes of the tasks participants are working on. Social and organizational factors that affect the work, but which are not obvious to individuals, may only become clear when noticed by an external observer (Sommerville 2010).

After studying this technique, Suchman (1987) found that ethnography is effective for discovering two types of requirements: Requirements that are derived from the way in which people actually work, rather than

the way in which process definitions say they ought to work; and requirements that are derived from cooperation and awareness of other people's activities.

Ethnographic studies can reveal critical process details that are often missed by other requirements elicitation techniques. However, because of its focus on the end-user, this approach is not always appropriate for discovering organizational or domain requirements, and as a consequence shouldn't be the only technique to use in an elicitation process (Sommerville 2010).

4.1.1.5 Prototyping

A prototype is a representation of a potential product, or only part of it. It has the objective of visually present the user with a simulation of the requirements. There are two kinds of requirements prototypes: high-fidelity prototypes that use specialized software tools and result in a partially working piece of software, and Low-Fidelity prototypes using pencil and paper, whiteboards, or some other familiar means. Teams usually prefer Low-Fidelity prototypes because they can be quickly generated and the users enjoy the spontaneous nature and inventiveness of these prototypes (Suzanne Robertson 2006).

When requirements are not properly formed, the users can't explain them or they are not clear for the analysts, this technique may be helpful in solving those problems (Suzanne Robertson 2006). This technique is also found useful when developing human-computer interfaces, or where the stakeholders are unfamiliar with the available solutions (Aybüke Aurum 2005).

4.1.1.6 Goal Based Approaches

This technique starts by collecting high-level goals that represent objectives for the system. Those goals are then refined into sub-goals, and the refinement of each sub-goal goes on until individual requirements are elicited. One of the disadvantages of this approach is that errors in the high-level goals of the system made early on can have a high negative impact. This technique has revealed to be helpful in situations that only

high level objectives of the system are known, and there is little or none understanding about the more specific problems the system is intended to resolve (Aybüke Aurum 2005).

4.1.1.7 Scenarios

People usually find it easier to relate to real-life examples rather than abstract descriptions. They can understand and explain what their interaction with the system might be. The discussion of the scenario(s) gives requirements engineers the opportunity to complete the requirements with more details. Each scenario usually covers one or a small number of possible interactions. Different forms of scenarios are developed and they provide different types of information at different levels of detail about the system. At its most general, a scenario may include a description of: what the system and users expects when the scenario starts, the normal flow of events in the scenario, what can go wrong and how this is handled, other activities that might be going on at the same time and the system state when the scenario finishes (Sommerville 2010).

4.1.1.8 View Points

Viewpoint approaches aim to model the domain from different perspectives in order to develop a complete and consistent description of the target system. The idea of this technique is to model the systems with the different points of view of its users or from the position of related systems. This approach is considered effective for projects where the system entities have detailed and complicated relationships with each other, as well as in the organization and prioritization of its requirements. One common criticism of viewpoint approaches is that they do not enable non-functional requirements to be represented easily, and are expensive to use in terms of the effort required (Aybüke Aurum 2005).

4.1.1.9 Apprenticing

Apprenticing involves the analyst actually learning and performing the current tasks under the instruction and supervision of an experienced user. In this technique the analyst is taught the operations and business processes by observing, asking questions, and physically doing, rather than being informed of them. Similar to Role Playing but more involved, apprenticing is very useful when the analyst is inexperienced with the domain, and when the users have difficulty in explaining their actions (Aybüke Aurum 2005).

4.1.1.10 Questionnaires

Questionnaires are mainly used during the early stages of requirements elicitation and may consist of open and/or closed questions. To be effective, the terms, concepts, and boundaries of the domain must be well established and understood by the participants and questionnaire designer. Questions must be focused to avoid gathering large amounts of redundant and irrelevant information. They provide an efficient way to collect information from multiple stakeholders quickly, however they are limited in the depth of knowledge requirements engineers are able to elicit. In the same way they provide no mechanism for the participants to request clarification or correct misunderstandings (Aybüke Aurum 2005).

4.1.2 Comparison of techniques

After the introduction of a set of some of the most common requirements elicitation techniques, it is important for one to decide what requirement(s) should be used in a given context.

In this section it is presented a table summarizing the different characteristics each technique covers. Figure 3.1 provides a quick overview of the techniques helping in the selection of the best one(s) to use in a certain context.

	Interviews	Domain	Groupwork	Ethnography	Prototyping	Goals	Scenarios	Viewpoints
Understanding the do-main	X	X	X	X		X	X	X
Identifying sources of requirements	X	X	X			X	X	X
Analyzing the Stakeholders	X	X	X	X	X	X	X	X
Selecting techniques and approaches	X	X	X					
Eliciting the Requirements	X	X	X	X	X	X	X	X

Figure 3.1 – Table identifying different characteristics each technique covers.

It is also presented a table (Figure 3.2) with the indication of which requirements fit well together as complementary or replaceable techniques. Those which can be used in cooperation are marked with a "C", and those which can be used as alternatives are marked with an "A" (Aybüke Aurum 2005).

	Interviews	Domain	Group work	Ethnography	Prototyping	Goals	Scenarios	Viewpoints
Interviews		C	A	A	A	C	C	C
Domain	C		C	A	A	A	A	A
Group-work	A	C		A	C	C	C	C
Ethnography	A	A	A		C	C	A	A
Prototyping	A	A	C	C		C	C	C
Goals	C	A	C	C	C		C	C
Scenarios	C	A	C	A	C	C		A
Viewpoints	C	A	C	A	C	C	A	

Figure 3.2 – Table with the possible combinations of techniques for the best outcome.

4.1.3 The techniques chosen for this project

After the study of a set of the most common requirements elicitation techniques, there were chosen the following techniques for this project: Interviews, Domain analysis and Apprenticing.

As it was already mentioned the mobile application to be developed in this project is pediatrics related. Its necessity was identified by a pediatrician (Dr. Simão) working in Santo António Hospital, who developed an application called Sabichão and has been asked by several of his coworkers for a mobile application.

With the cooperation of Dr. Simão it was possible to use the interview technique, using the Sabichão application as a starting point to understand the pediatrics context and its needs.

To complement that information, it was used the domain analysis technique for the Sabichão application as well as for several mobile application available in the markets. The analysis made for the Sabichão was focused in the exploration of the application because there is no documentation available. The analysis of the other applications was limited to the available documentation, mainly as a consequence of being paid. These analyses can the found in the Chapter "State of the Art" of this document.

Apprenticing was another technique used, and was found to be very helpful in the understanding of the application context. It was also useful to learn how to fully interact with some of the Sabichão modules that have a set of features considered important to the mobile application. It was performed under the supervision of a pediatrician.

4.2 Volere

4.2.1 The Volere template description

Volere is a requirements specification template that represents the most basic type of tool used by analysts to support the process of requirements elicitation (Aybüke Aurum 2005). This template can be used for any kind of product or project. Since the release of Volere in 1995, it has been used by thousands of projects in hundreds of countries. It is a distillation of literally hundreds of requirements specifications and is currently used by thousands of organizations all over the world (Suzanne Robertson 2006).

A requirements specification template makes the requirements writing easier to do and more convenient. The output of this requirements process is a written description of the requirements to be used as input to the design of the product.

The Volere is a template that sets out a complete description of the product's functionality and capabilities. The template itself is composed by five different sections: Project Drivers, Project Constraints, Functional Requirements, Nonfunctional Requirements and Project Issues. Each of these sections contains subsections separating the subjects into parts. It is up to the requirements analysts to carefully consider which sections and subsections to apply to a particular project (Suzanne Robertson 2006).

This template benefits from the shell when gathering individual requirements. The shell has the form of a card that contains a number of components related to a specific requirement. Each component gives its contribution to the understanding of the whole requirement. Although they may at first glance seem rather bureaucratic, it was found that their value repays the effort used to gather the information. The

requirement shell is completed progressively, because it is not practical to find all components of one requirement before moving on to the next stage. The following picture illustrates the aspect of the shell.

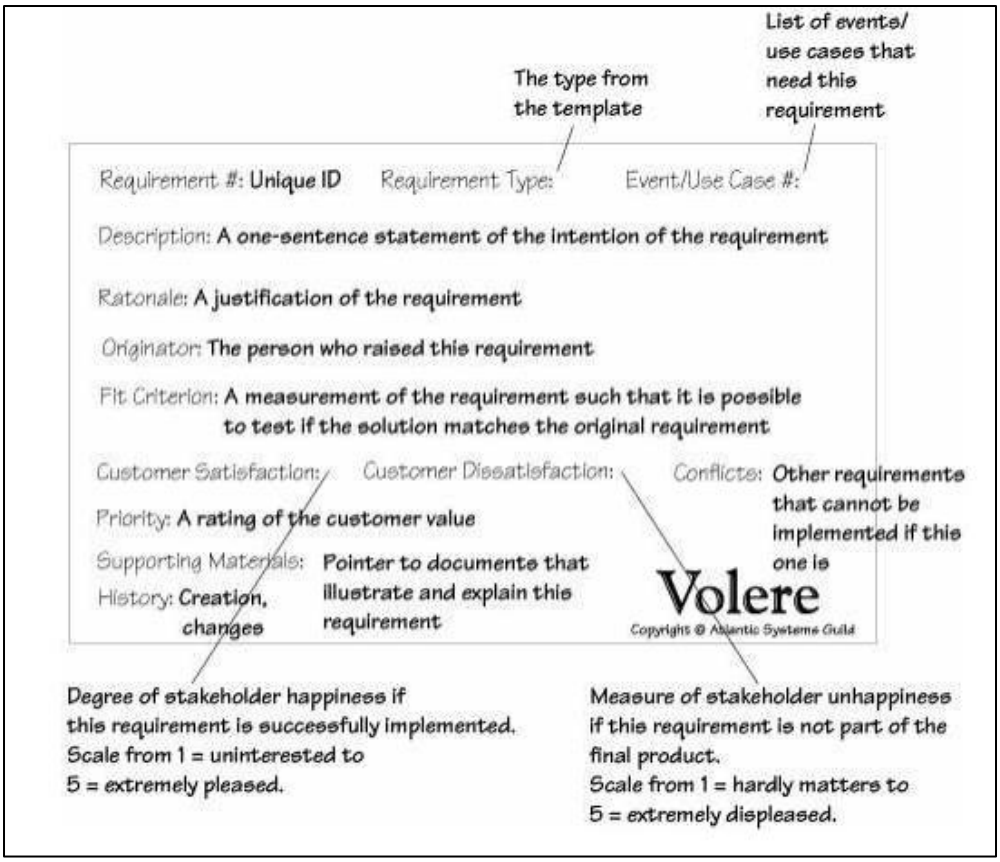


Figure 3.3 – The Volere requirements shell.

4.2.2 The utilization of Volere in this project

As it was said in this section, Volere is a requirements specification template that addresses a wide range of topics, like Project Drivers and Project Constraints. As many of these topics were already introduced and many others make much more sense in the corporate business, in this project it is only going to be used the shell presented in figure 3.3 in order to document the elicited requirements. These requirements can be found in Appendix I.

4.3 Requirements Prioritization

Requirements Prioritization is a discipline that concerns the prioritization of requirements of a given project, with the aim of developing the functionality that is most desired by the customers, as well as least risky, least costly, and so forth. Prioritization is a crucial step to make good decisions regarding product planning for single and multiple releases. The quality of a software product is often determined by the ability to satisfy the needs of the customers and users. Hence, eliciting and specifying the correct requirements and planning suitable releases with the right functionality is a major step towards the success of a project. If the wrong requirements are implemented and users resist using the product, it does not matter how solid the product is or how thoroughly it has been tested (Aybüke Aurum 2005).

This section introduces several important concepts when performing prioritizations, such as: prioritization techniques, several aspects to consider in the prioritizations, among other issues.

The result of prioritizations suggests which requirements should be implemented. Hence, the prioritization techniques could be a valuable help to get an understanding of what is important and what is not for a given project.

Decision-making becomes a complex task to perform when the number of different aspects to consider starts growing. A decision that is easy to do when only considering one aspect gets tougher if a few more aspects are also to be considered. For example, the functionality that is most important for the customers might not be as important when other aspects (e.g. price) are also considered (Aybüke Aurum 2005).

Most software projects have more candidate requirements than can be released within the time and cost constraints. Prioritization helps to identify the most valuable requirements from this set by distinguishing the critical few from the trivial many. Some activities requirements prioritization provides support are:

- In the stakeholder's decision on the core requirements for the system;
- In planning and selecting an ordered, optimal set of software requirements for implementation in successive releases;

- In the trade-off of desired project scope against sometimes conflicting constraints such as schedule, budget, resources, time to market, and quality;
- In the selection of only a subset of the requirements and still produce a system that will satisfy the customer(s);
- In establishing relative importance of each requirement to provide the greatest value at the lowest cost (Aybüke Aurum 2005).

The resulting prioritization might be used as a guide throughout the project.

4.3.1 Aspects of Prioritization

When prioritizing requirements there are many aspects that may be taken into account. An aspect is a property or attribute of a project and its requirements that can be used to prioritize requirements, such as: penalty, cost and risk. When prioritizing requirements based on a single aspect, it is easy to decide which one is most desirable. If there are other aspects to be considered, such as cost, customers can change their mind and high priority requirements may turn out to be less important. Often, the aspects interact and changes in one aspect could result in an impact on another aspect. Several aspects can be prioritized, and the decision of which aspects to consider depends on each specific project. Aspects are usually evaluated by stakeholders in a project (managers, users, developers, etc.). Bellow it is presented a list of aspects that may be important in a software development context (Aybüke Aurum 2005).

4.3.1.1 Importance

When considering importance, the stakeholders are expected to prioritize the most important requirements for the system. However, the term importance may mean different things for different stakeholders. Importance could, for example, be urgency of implementation, importance of a requirement for the product

architecture, strategic importance for the company, etc. Hence, it is necessary to specify the meaning of the word importance, or the different kinds of importance that should be considered (Aybüke Aurum 2005).

4.3.1.2 Penalty

The penalty aspect is considered to be the consequence that will occur if a requirement is not met. For example, failing to conform to a standard could incur a high penalty even if it is of low importance for the customer. The same goes for implicit requirements that users take for granted, and whose absence could make the product unsuitable for the market (Aybüke Aurum 2005).

4.3.1.3 Cost

The implementation cost is usually estimated by the developing organization. The cost is usually measured by: complexity of the requirement, the ability to reuse existing code, the amount of testing and documentation needed, etc. Cost is often expressed in spent hours by the staff (effort) since the main cost in software development is often related to the necessary time (Aybüke Aurum 2005).

4.3.1.4 Time

In software development the time and cost aspects are highly related, since cost is often measured in time. However, time is influenced by many other factors such as degree of parallelism in development, training needs, need to develop support infrastructure, complete industry standards, among others (Aybüke Aurum 2005).

4.3.1.5 Risk

Risk is part of every project, and may be divided in internal (technical and market risks) and external risks (e.g. regulations, suppliers), both taken in risk management account. Their probability and impact must be considered when evaluating the project tasks. Risk management is also useful for identifying possible problems of desirable requirements for a system that could arise in the future. Such risks could for example include performance risks, process risks, schedule risks etc. Based on the estimated risk likelihood and risk impact for each requirement, it is possible to calculate the risk level of a project (Aybüke Aurum 2005).

4.3.1.6 Volatility

The volatility and risk aspects are related at a certain level, and therefore there is no consensus on whether volatility should be considered separately or with the risk aspect. Either way, some volatility examples are: market changes, business requirements change, legislative changes, users change, or requirements become clearer during the software life cycle. Therefore this aspect may be the reason for the increase of projects costs later in the development phase, or may early require a more volatile architecture for the system and consequently more costly (Aybüke Aurum 2005).

4.3.2 Using multiple aspects

There should be considered multiple aspects when prioritizing requirements. It is possible to combine different aspects in many different ways, and the better choice depends on the specific situation.

In the Cost-Value approach there are considered the value (importance) and cost aspects resulting a prioritized requirements that give most value for the money. The Planning Game (PG) from eXtreme Programming (XP) uses a similar approach when importance, effort (cost), and risks are prioritized. In

Wiegers' approach, the relative value (importance) is divided by the relative cost and the relative risk in order to determine the requirements that have the most favorable balance of value, cost, and risk. This approach further allows different weights for different aspects in order to favor the most important aspect (in the specific situation) (Aybüke Aurum 2005).

4.3.3 Prioritization Techniques

The prioritization techniques have the objective of establishing a relative order between different requirements. There are several techniques available for this purpose, some more powerful than others.

Following it is presented a set of prioritization techniques. Some of the techniques assign a priority value to each of the requirements, while other techniques assign the priority values to a group of requirements (when examples are given, importance is used as the aspect to prioritize) (Aybüke Aurum 2005).

4.3.3.1 Analytical Hierarchy Process (AHP)

The Analytic Hierarchy Process (AHP) is a systematic decision-making method that has been adapted for prioritization of software requirements. This technique compares all possible pairs of requirements, in order to determine which has higher priority, and how much higher. Usually it is used a scale from one to nine where one represents equal importance and nine represents absolutely more important.

Several studies concluded that AHP is not suitable for large numbers of requirements. However, in its original form, the redundancy of the pair-wise comparisons allows a consistency check where judgment errors can be identified and a consistency ratio can be calculated (Aybüke Aurum 2005).

4.3.3.2 100-Dollar Test

The 100-dollar test is a very straightforward prioritization technique where the stakeholders are given 100 imaginary units (money, hours, etc.) to distribute between the requirements. The result of the prioritization is presented on a ratio scale (Aybüke Aurum 2005).

In spite of being a straightforward technique, there were identified a few problems that it is necessary to be aware of. If there are too many requirements it was found that the 100 imaginary units may be scarce to prioritize all the requirements. In these situations, it is suggested to give the stakeholder a larger amount of units (e.g. 100,000 units). Another problem that may happen is that at the end of the prioritization the total sum is not equal to 100 units, as a consequence of a miscalculation. This can be prevented by using a tool that keeps count of how many points have been used. The prioritization process should be done only once, since stakeholders may change their evaluation if they don't get one of theirs favorite requirements as a top priority. The same can happen if a stakeholder decides to spend all his credits in his favorite requirements, knowing that some other necessary requirements will get credits anyway. A possible solution may be limiting the amount of units that could be spent on a single requirement, although it may prevent the stakeholder from genuinely prioritize the requirements (Aybüke Aurum 2005).

4.3.3.3 Numerical Assignment (Grouping)

Numerical assignment is the most common prioritization technique. This technique groups requirements into different categories, usually three. The numbers of each group must be associated with a clear meaning (e.g. critical, standard, optional) instead of terms that may be different from stakeholder to stakeholder (e.g. high, medium, low) (Aybüke Aurum 2005).

One problem that arises with this technique is that stakeholders tend to consider most of their requirements as critical. A proposed resolution is to limit the number of requirements that may be in each group. Since this technique prioritizes the requirements into groups, requirements from the same group

have the same priority and as a consequence it is not possible to know the priority order of the requirements within a given group (Aybüke Aurum 2005).

4.3.3.4 Ranking

When using this technique every requirement is assigned with a numerical number, where the first is ranked 1 and the last is ranked *n* (for *n* elements). Each requirement has a unique rank (in comparison to numerical assignment) but it is not possible to see the relative difference between the ranked items (as in AHP or the 100-dollar test). Ranking is more appropriate for use when there is only one stakeholder, because it might be difficult to align several different stakeholders' views (Aybüke Aurum 2005).

4.3.3.5 Top-Ten Requirements

As the name suggests, in the Top-Ten Requirements technique the stakeholders are asked to pick their top ten requirements for the system, without any order between them. This technique is good for multiple stakeholders of equal importance.

Although some conflicts may arise while using this technique, such as: one stakeholder gets three requirements met while another gets six. To avoid that, it is important to not just do the average of all the stakeholders' prioritizations, but also try to include requirements from all the participants. At the same time, it is important to evaluate if the final result will satisfy at least a good percentage of the stakeholders (Aybüke Aurum 2005).

4.3.4 Comparison of techniques

Following there is a table that presents a comparison of the different techniques in what concerns the scale, granularity and sophistication.

Technique	Scale	Granularity	Sophistication
AHP	Ratio	Fine	Very Complex
Hundred-dollar test	Ratio	Fine	Complex
Ranking	Ordinal	Medium	Easy
Numerical Assignment	Ordinal	Coarse	Very Easy
Top-ten	-	Extremely Coarse	Extremely Easy

Figure 3.4 – Comparison of the different techniques.

In general, it is recommended to use the simplest possible technique to prioritize and resolve possible conflicts that may occur in a specific situation (Aybüke Aurum 2005).

There are some programing methods that used more than one technique for the prioritization. The Method Planning Game (PG) combines numerical assignment and ranking by first dividing the different requirements into priority groups and then ranking requirements within each group. Another example is doing requirements triage, where requirements are put into different categories (e.g. must be in the product, optional or requirements that need more attention). Then, for the most important requirements it is used a more sophisticated technique while a less sophisticated technique is used for the less important requirements (Aybüke Aurum 2005).

The amount of stakeholders involved in the prioritization process will make a difference in what techniques should be used. Every project is in one of these options: One customer, several known customers and mass-market. In the first option, there is only one person to do the prioritization. For example, eXtreme Programming has an “on-site customer” as one of the core practices (the focus is on having one customer even though this customer could represent a market). In this situation, one should verify if the end-user and the customer are the same and if not, it is important to consider including both in the prioritization process; since they might have different needs that should be met for the success of the product. When having several stakeholders, the prioritization is more difficult to do because stakeholders may have

conflicting demands for the product. In this situation, it is necessary to resolve the conflict in a way everyone wins. If the product is market-driven then it is impossible to get all stakeholders involved in the process. It is necessary to collect information from sources like marketing intelligence, competitor's intelligence, marketing research, personas, among others (Aybüke Aurum 2005).

4.3.5 The Aspects and Techniques chosen for this project

In this section there were introduced several aspects and techniques proper for requirements prioritization, each of those with its pros and cons. In this project the aspects that are going to be used are: importance and cost/time. The definition of importance in this project is: The requirements that would be most useful in the application for the pediatricians. In this context cost and time have the same meaning since cost is going to be measured in time.

The use of importance was considered crucial, because this application is for the pediatricians, and as a consequence it is important that the most desirable/useful functionalities are part of the final application. The importance is the main aspect of prioritization. In order to differentiate the requirements that were given the same level of importance, it was used a second aspect: the estimated cost/time. The objective of the second aspect is to favor the fastest requirements to implement while penalizing requirements that take more time. The cost/time value was only assigned to the groups of requirements with the same value of importance.

As it was not found in the bibliography a way of applying the cost/time aspect, it was used the following strategy: To each of the requirements that were given the same priority it was assigned a sequential value. The fastest to implement requirements are given a smaller value than the most time consuming ones.

In this project there is a particular stakeholder that has agreed to closely follow the project, and that is representing a wider group of stakeholders: the pediatricians.

The technique used to prioritize the requirements is the 100 dollar test. As it is said above, this technique is simple to understand and apply and its final results show how important each requirement really is. In

order to prevent the problems this technique may run into, the prioritization was made in a calculation sheet that indicated the amount that the stakeholder had already spent. Also, the prioritization was made only once, as it is suggested in the bibliography.

The final prioritization is available in Appendix II.

5 Analysis of current technologies

This chapter is intended to perform an analysis of the available technologies today to build applications for mobile devices. Initially an analysis of the smartphone operating systems market share is performed. The forecasts of the market share for the next years are also taken into account. With this information it is possible to makes decisions based on today and tomorrow most important smartphone operating systems. The next step is to study some of the available frameworks to develop cross-platform mobile applications, which provides an understanding of how each platform works. Finally, the technology to use in the development of the final application is chosen, given the conclusions of the performed study and the already known functionalities it is intended to develop.

5.1 Smartphones Market share

Looking into the smartphones market, it is not difficult to spot an operating system fragmentation issue. Making an analogy with the computers market it becomes clearer. Nearly 20 years of mobile devices existence, there are too many different operating systems available with a considerable market share (Freedman 2007). When an organization or a group of programmers decide to develop an application for a specific operating system (e.g. Android), their final product will inevitably reach a limited amount of the smartphones users.

There are several smartphone operating systems available on the market, such as: Android, iOS, Windows Phone, BlackBerry, Symbian OS, etc. Figure 4.1 illustrates the smartphone operating system market shares from 2009 and 2012.

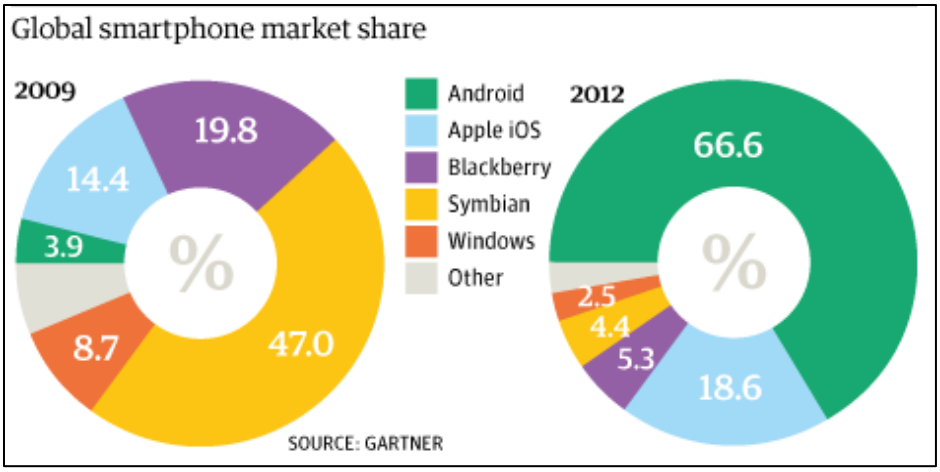


Figure 4.1 – Global smartphone operating systems market share from 2009 and 2012 (Garside 2013)**.**

There is also no sign of operating systems reduction over time. Recent news show that several worldwide recognized technology organizations have been working in releasing their own smartphone operating systems, willing to strive for their position in this market. Some examples are: Ubuntu for phones (Trenholm 2013), Firefox OS (Garside 2013) and Tizen (Lee 2013).

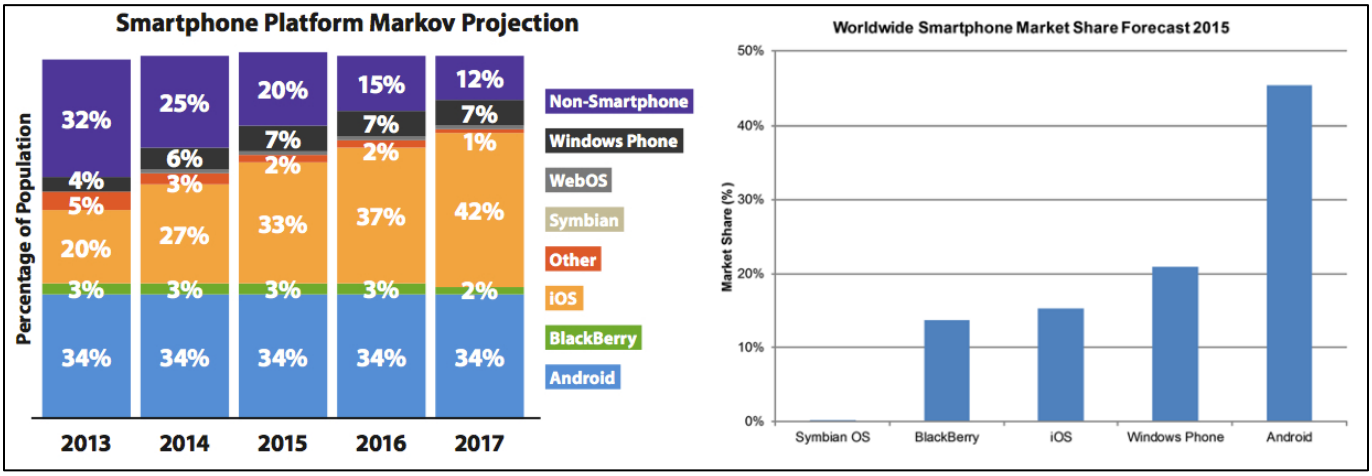


Figure 4.2 - Yankee Group report at the left and the IDC forecast at the right (Paczkowski 2013)**.**

In Figure 4.2 it is possible to see two forecast graphics for the smartphones market share. The graphic presented in the left is the result of a survey conducted by the Yankee Group company, in which it is concluded the iOS tendency to become the operating system with the highest market share in the next few

years (Paczkowski 2013). In the right side of the same figure it is presented a graphic with the forecast of the smartphones market share for the year of 2015, produced by the IDC Company. It predicts that Android is going to have by far the greatest market share, followed by Windows Phone.

With these two graphics it is possible to conclude that these two companies share predictions with considerable differences. Also, none of the companies included in their graphics the operating systems that are coming out. However, both forecasts seem to agree that it is not expected the smartphone operating systems market fragmentation to suffer a radical change.

In order to reach most of the users when developing a smartphone application, there are two possible approaches: the first is to develop a native application for each of the operating systems with the biggest market share; the second is to use a cross-platform development framework in order to develop the application only once, and deploy it to several operating systems.

In this project, it is intended to release a piece of software that may run on the majority of the smartphones in the market at the present moment as well as in the future. As a result it is going to be used a cross-platform development framework in order to shorten the development period of time.

5.2 Analysis of cross-platform development frameworks

In the past few years, many cross-platform development frameworks have emerged. There has been an explosion of activity in this area as mobile devices become faster and more widely adopted, and particularly with a fast-growing market for applications (Allen et al. 2010). Cross-platform development frameworks are intended to write an application once, and deploy it to several operating systems. This task may sound simple in theory, but is in fact highly complex, technically, due to a range of factors such as the highly fragmented mobile technology landscape, rapidly evolving standards, limitations imposed by the mobile devices themselves (screen size, input methods, display capabilities, etc.) and also constraints of the mobile network such as high latency and low bandwidth (Hartmann 2011).

As each vendor implements its own application development stack, achieving cross-platform and cross-device consistency is a non-trivial task. Fortunately as the web becomes ubiquitous and its technologies evolve, with more and more mobile browsers implementing new standards like HTML 5, CSS 3 and JavaScript, web applications are rapidly becoming an attractive and cost-efficient way of developing mobile applications. These can rival native apps in terms of rich user experience and access to advanced capabilities like storage and geo-location (Hartmann 2011).

The frameworks fall into two categories: those that let you create a native mobile application using cross-platform APIs (Application Programming Interfaces), and HTML/CSS/Javascript frameworks that let you build cross-platform interfaces that run in a web browser. It is common practice to combine these to create cross-platform native applications. Some examples of native cross-platform frameworks are Rhodes and Titanium; while some HTML/CSS/Javascript frameworks are PhoneGap and iWebKit (Allen et al. 2010).

Following, it is going to be presented a set of frameworks that facilitate the deployment of an application to several mobile operating systems. Due to the large amount of available options only a few will be discussed, which are believed to be part of the widely adopted frameworks by the development community. In spite of the existence of several commercial frameworks, there are going to be considered only technologies that are open source, or at least that have a set of development tools free to use.

5.2.1 Rhodes

Rhodes is a cross-platform smartphone application framework that was developed by Rhomobile in 2008 and currently is maintained by Motorolla Solutions. This framework supports all major smartphones, such as: Windows Phone, Android, iPhone, BlackBerry, among others. It achieves this by providing a runtime environment that executes on the device wrapped around a native app. This runtime Virtual Machine (VM), which is ported to the different platforms, abstracts the communication between the mobile app and the device (Hartmann 2011).

Rhodes allows the development of an application using the Ruby programming language as well as several well-known web technologies: HTML, CSS and JavaScript. It reveals to be an advantage for programmers that already have a programming background with these technologies since it is possible to create native mobile applications without having to learn the specific SDKs of each platform. The framework can be used under Windows, Linux and Macintosh. It is necessary to have the specific SDKs installed of each of the devices it is pretended to deploy the application. Android and Symbian devices run on Java and are cross-platform, while BlackBerry and Windows Mobile devices require Windows OS and iPhone devices require Macintosh OS (Allen et al. 2010).

Rhodes is an open source framework allowing developers to create consumer-type applications free of charge (Arcuri 2013). As a consequence of being open source, it is possible to see the lines of code that compose Rhodes. It is also possible to extend it, fix bugs or to create a personalized version of the framework.

Ruby code helps to structure and control business logic using the built in Model-View-Controller. It has several similarities with the Ruby on Rails programming language like the MVC pattern and the possibility of embedding Ruby in the views (ERB). Rhodes also provides a local Object Relational Manager, called Rhom. It includes code to persist local data and sync remote data using RhoSync. Therefore, Rhodes developers do not have to worry about writing data storage and sync logic into their applications and can focus instead on presentation and business logic.

At the end of the development, project files are compiled into a native executable that is installed on the device. Since Rhodes apps are native binary applications, they can be submitted and distributed through the Apple iTunes App Store, BlackBerry World, Android Marketplace, and other distribution channels. It is usually necessary to sign up in the different developer programs and acquire the cryptographic keys as it also happens when developing applications using the native SDKs (Allen et al. 2010).

Rhodes may be installed on the computer as a gem of Ruby and used from the command line. It may also be installed through the installation of RhoStudio, which is an Eclipse plug-in to faster the application development. It is also possible to install RhoStudio with a development suite called RhoMobile Suite. This Suite is a commercially-supported set of tools that contains RhoConnect (a tool to sync data with many

different data sources), RhoStudio and RhoElements (based on the Rhodes framework with an extended set of APIs) (Solutions 2013).

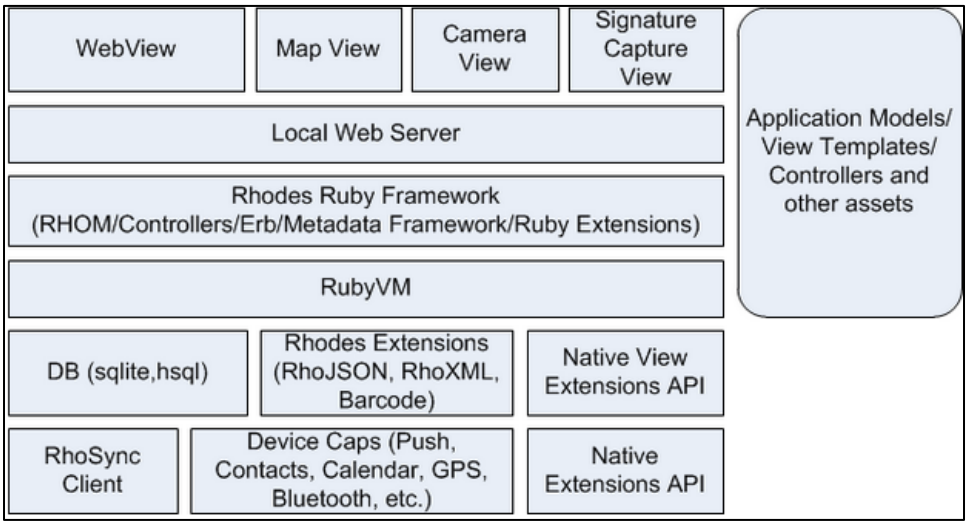


Figure 4.3 – Rhodes Framework Architecture.

Figure 4.3 presents the architecture of the Rhodes framework. An application developed with Rhodes is a web application that runs locally on the mobile device. A lightweight web server runs on the device in order to display the application. A Ruby virtual machine establishes the bridge between the project code and a set of functionalities available on the device, such as: Database, Calendar, GPS, among others.

5.2.2 PhoneGap

Phonegap is an open source framework for building mobile applications using web technologies. It was created by a company named Nitoby in 2008 and is currently supported by Adobe Systems. With HTML, CSS and JavaScript, this framework allows the applications deployment to a wide variety of platforms: iPhone, Android, Blackberry, WebOS, Windows Phone, Symbian and Bada (Systems 2013). Programmers with experience in those web technologies already have the needed knowledge to start developing, although it is necessary to use specific device SDKs and tools in order to build the mobile application (Allen et al. 2010).

PhoneGap applications are not purely HTML/JavaScript based, nor are they native. These are hybrid applications. Parts of the application, mainly the UI, the application logic, and communication with a server, is based on HTML/JavaScript. The other part of the application that communicates and controls the device (phone or tablet) is based on the native language for that platform. PhoneGap provides a bridge from the JavaScript world to the native world of the platform, which allows the JavaScript API to access and control the device. The JavaScript API has rich functionalities providing access to device capabilities, such as: accelerometer, camera, geolocation, network, storage, among others. Therefore, to take full advantage of this platform the programmer must be comfortable with the JavaScript programming language. PhoneGap does not come with an Integrated Development Environment (IDE) and as a consequence it is needed to use Eclipse with the Android SDKs to develop for Android, or Xcode if it is going to develop for iPhone (Rohit Ghatol 2012).

Figure 4.4 – PhoneGap architecture.

In spite of having a good set of capabilities, the PhoneGap API is not rich in features to work with the user interface. It is up to the developers to create their own styles in order to make the application have a good look. There are several user interface frameworks available to aid the interface development, such as: jQueryMobile and Sencha Touch (Hartmann 2011).

This framework imposes little structure and guidelines on how to best develop applications with it. The users are free to architect their solutions in a way that best suits their needs. This can be an advantage for experienced developers but it can also become challenging and promoting bad application design for the novice developers.

Phonegap may be a good choice when there is a web application that is necessary to port to a mobile environment. Since most code is already in a web format, the task of converting it to mobile web in most cases should be an easy task to perform (Hartmann 2011).

5.2.3 Titanium

Titanium is a cross-platform development framework that uses web technologies to deploy mobile applications. It was created in 2008, by the Appcelerator company and is a commercially supported product with its source code released under the Apache 2 license (Allen et al. 2010). Titanium is a Software Development Kit (SDK) with over 5,000 device and mobile operating system APIs, Studio, a powerful Eclipse-based IDE, Alloy, an MVC framework and Cloud Services for a ready-to-use mobile backend (Appcelerator 2013a). It may be installed under Windows, Macintosh and Linux. There is a free community edition to develop and deploy the applications as well as the Professional and Enterprise editions that offer additional support and services. It has a good documentation and other online resources, like training videos. It is also possible to purchase training courses and certifies in the Appcelerator website. All these resources allow new developers to quickly start developing their applications.

This framework is mainly focused on the Android and iPhone/iPad devices. It is also available a set of functionalities for BlackBerry devices, however there is no support for this platform and the APIs are not as mature as the other ones. In spite of being possible to use HTML, CSS and JavaScript to develop applications, it isn't mandatory. The look and feel of the different objects of the application may be customized just by modifying its parameters. Therefore, JavaScript is the only language that it is necessary

to use in Titanium. When using JavaScript in Titanium the developer may take advantage of its object-oriented model, with its methods and properties that Titanium provides via its API (J. Anderson 2013).

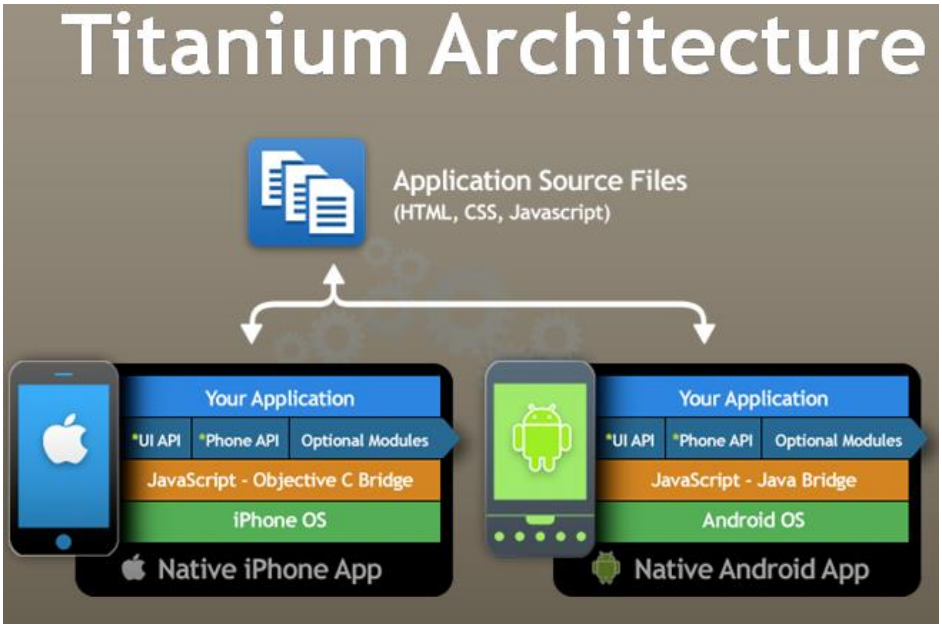


Figure 4.5 – Titanium architecture.

When a Titanium app is compiled, its engine processes the JavaScript and then builds an appropriate native project for the specified platform. For each iOS application an Xcode project is created and then compiled using Apple's compiler, while with Android it is created a native java application and compiled using the Android compiler. This means that the final project is compiled with the native APIs, in spite of the application being developed with the Titanium APIs. The end application is 100% native, using 100% native controls.

One of the big advantages of this framework is that due to its ability to compile the applications with the native APIs, it provides a wider set of native device functionalities that a web application can't provide (J. Anderson 2013).

5.2.4 Adobe Flash, Flex and Air

Adobe Flash platform contains several Flash-based runtime clients: Flash Player, Flash Lite and Adobe AIR (Adobe Integrated Runtime). Each of these clients has its own specific APIs. This platform also comes along with a component based framework, named Flex. These technologies work with the SWF format (J. G. Anderson 2011).

When it comes to mobile devices, both Flash Lite and Flash Player support Flash content in such devices. Flash Lite is intended to run on performance-limited mobile devices. On the other hand, Flash Player is intended to support the deployment of SWF (Shockwave Flash) content for web browsers and mobile devices.

The Flash technology is fully supported on Android and BlackBerry Tablet OS mobile platforms, while it is not on the Apple iOS platform. However, using Action Script 3 (AS3) and AIR, it is possible to target the applications to run on the platform via standalone applications.

AS3 is an object-oriented language for creating media content for playback in the Flash runtime clients Flash Player, Flash Lite, and Adobe AIR. Developers that are familiar with the Java and/or JavaScript languages should find AS3 familiar (J. G. Anderson 2011).

The Flex framework uses a set of AS3 classes to provide user interface (UI) components allowing the development of rich media applications. Adobe Flex is supported by the runtime APIs of Flash Player and Adobe AIR. It is available in the Adobe Flash Builder IDE, as well as through a SDK that allows the use of command line tools to create the application. Adobe Flash Builder is a commercial product, however it is possible to develop mobile applications using the Flex SDK with a free IDE, such as: FlashDevelop (Paananen 2011).

The Flex framework also uses the MXML language. MXML is an XML tag-based markup language, used in the layout and design of components and data assets for Flex-based user interfaces. In order to preserve the integrity of the MXML documents, it is a structured language with a set of rules that must be followed, just as it is seen in the XML language (J. G. Anderson 2011).

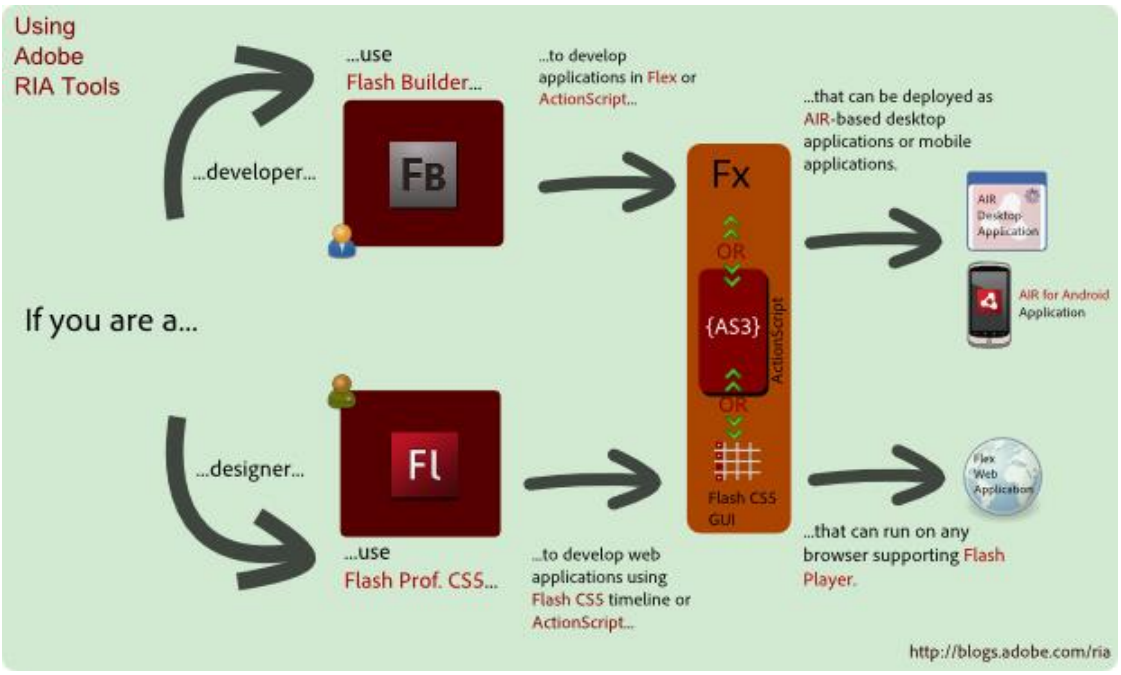


Figure 4.6 – A simplified schematic of the utilization of Adobe RIA tools.

Adobe Integrated Runtime (AIR) is a cross-platform run time that allows developers to create and deploy applications for a variety of operating systems outside of Internet browsers. AIR has the ability to install the applications in the same way as the native applications. With the AIR run time, developers may use the Flex framework to create mobile applications that are installed as standalone applications in the mobile devices (J. G. Anderson 2011). Adobe AIR SDK is also free and comes along with the tools necessary to build and deploy Adobe AIR applications (Paananen 2011).

5.3 The technology chosen for this project

In this project it is intended to develop a medical smartphone application that is going to have a set of pediatric related features, aiming to aid pediatricians in their professional daily practice.

As it was already said, the technologies presented in this document, are just a subset of the total amount of available technologies capable of producing such an application. It is believed that the chosen

technologies are representative of the wide variety of available choices. In order to choose the technology for this project, there were considered the following criterions:

- The technology used in this project should allow the freely development and distribution of the application;
- The application should be made available for the smartphone operating systems with the biggest market share;
- The technology involved should have the necessary capabilities to implement the desired functionalities;
- The technology should be as easy to learn and use as possible.

A limitation of the multi-platform frameworks is the operating systems that it is possible to deploy the applications. The frameworks that deploy to the larger amount of operating systems are Rhodes and PhoneGap, while those that deploy to a smaller amount are Flex and Titanium.

Another limitation found in the studied frameworks, is the ability they have to take advantage of device capabilities. It's generally accepted that frameworks with the capability of deploying native applications, are able to provide device capabilities related APIs that frameworks deploying web-based/hybrid applications cannot provide.

It is already known that there are functionalities that are going to involve calculations with dates. Some frameworks have limitations when it is necessary to work with calendars, such as: PhoneGap and Titanium. PhoneGap core APIs don't incorporate calendar functionalities in spite of already being available third party plugins for Android and iOS, while the Titanium framework only provides calendar capabilities for iPhone and iPad devices (Appcelerator 2013b) (Community 2012).

The Adobe Flash, Flex and Air technologies provide a powerful set of tools to develop mobile applications. This technology only allows the applications deployment to iOS, Android and Blackberry. Also, these development tools are normally used with the Adobe Flash Builder IDE that is commercially supported. The wide documentation and video training provided by Adobe presupposes its utilization. The use of the Flex

SDK with an open source IDE, like FlashDevelop, has very poor documentation and was also not found any active community debating/supporting the combination of such technologies.

Finally, Rhodes is the technology that is going to be used in this project. It was not found any kind of limitation for the requirements it is intended to develop. In spite of being part of a commercially supported development suite, it may also be used separately for free. It has an Eclipse plug-in (RhoStudio) that makes the development with this framework user friendly. Also, it has a good documentation online and a forum with an active community hosted in the Motorola Solutions website.

As a result, Rhodes is the technology that is going to be used to develop the pediatric smartphone application.

6 The Final application

This chapter discusses a few aspects of the implementation phase, such as: the use of the elicited requirements and its prioritization as well as the use of the Rhodes technology. It is also done a comment on the use of web services, database and the test phase of the application. Finally, it is presented the final structure of the app and the requirement each functionality corresponds.

The developed mobile application is called “Pediatria” (Pediatrics), and includes every elicited requirement for the application.

6.1 Implementation phase

During the development of the smartphone application, the elicited requirements revealed to be a helpful tool in order to easily identify the context of each requirement. These requirements were most helpful during the early stages of the implementation, complementing the lack of domain knowledge. Also, the requirements prioritization made it possible to measure how much each requirement really mattered. It also helped scheduling the requirements implementation. All the elicited requirements were successfully implemented and its implementation followed the requirements prioritization final results, every time that it was possible.

As there were several technologies involved in this project and little or none experience with them, it was necessary to spend some time learning them. The involved technologies are: Ruby, Ruby on Rails, Rhodes, jQuery Mobile and CSS. The learning process toke place both before and during the application development.

Throughout the development there were little changes to the previously gathered requirements, for example: Requirements 11 and 12 were joined and the data that used to calculate the final results was updated, due to an article that was recently published (Tanis R Fenton 2013).

The supervision and validation by Dr. Simão at the pace that the functionalities were being developed resulted in a refinement of the provided features as well as on how things were being presented.

6.1.1 Database

Every time it was necessary to work with large amounts of data, such as the tables of values used to calculate the anthropometric and blood pressure centiles, it was decided to store the data in a local database. If it was not the case, the values were stored in the controller of the functionality in order to avoid the required time to communicate with the database.

This procedure could easily be accomplished by creating and populating the database the first time the application was started. As it was found to be rather slow, the adopted solution was to package the final application with a pre-populated database. To perform this, it was necessary to create and populate the application database the first time it was started in each of the Operating Systems virtual devices and extract it from there.

Following is presented an image showing a table of the database with values of female blood pressure values.

	ano	perc	s5th	s10th	s25th	s50th	s75th
1	1	50th	83	84	85	86	88
2	1	90th	97	97	98	100	101
3	1	95th	100	101	102	104	105
4	1	99th	108	108	109	111	112
5	2	50th	85	85	87	88	89
6	2	90th	98	99	100	101	103
7	2	95th	102	103	104	105	107
8	2	99th	109	110	111	112	114
9	3	50th	86	87	88	89	91
10	3	90th	100	100	102	103	104
11	3	95th	104	104	105	107	108
12	3	99th	111	111	113	114	115
13	4	50th	88	88	90	91	92
14	4	90th	101	102	103	104	106
15	4	95th	105	106	107	108	110
16	4	99th	112	113	114	115	117
17	5	50th	89	90	91	93	94

Figure 5.1 - A database table with systolic and diastolic female blood pressure values.

6.1.2 Web Service

In order to securely gather patient information from the hospital it was necessary to communicate with a web service created specifically for this purpose. Since the hospital systems work with Microsoft databases, it was necessary to use the SOAP technology (Simple Object Access Protocol). As Rhodes by default does not have this capability, the simplest and lightest way found to accomplish this was by extending Rhodes with an adaptation of a Ruby SOAP controller available on the internet (Spritle 2011). It was also necessary to extend Rhodes in order to communicate via the HTTP protocol (Hypertext Transfer Protocol) and in order to parse XML (Extensible Markup Language) content.

6.2 The Rhodes framework

Rhodes was the technology used to develop this application. As it was already said, Rhodes provides the possibility of building mobile devices applications, using a Ruby on Rails similar programming language (web programming language). This framework makes it possible to develop an application once and deploy it to several platforms. It also has an active community that discusses the technology on the internet, and for several times the problems this project faced were already asked and answered by other community members. In order to facilitate the development, it was used RhoStudio (an Eclipse plugin), taking advantage of the Eclipse IDE. It comes with a fast mobile device simulator that can be used to test the application. The application can also be tested on the Operating System's virtual devices. However, for several times the application was completely functional in the RhoStudio simulator and at the same time not completely functional on a real device or virtual device.

Rhodes already comes with jQuery mobile by default. It is a widely known multi-browser JavaScript library responsible to generate a nice and user friendly layout. However, it was found that the calendar functionality doesn't behave in the same way in every Android OS version. The calendar of most recent jQuery mobile versions (example: version 1.3.1) didn't work on the older Android devices (example: version 2.2); while older jQuery mobile versions (example: 1.0a4.1) have a calendar that doesn't work properly

(example: complete application crash) on all the earlier Android devices (example: Galaxy S 3 version 4.1.2). In spite of the calendar issues, in general, jQuery Mobile revealed to work well in conjunction with Rhodes.

6.3 Test phase

In order to guarantee the accuracy of each of the mobile functionalities, this application was tested during the whole implementation phase. Every application functionality was also tested by Dr. Simão (pediatrician) who closely followed the project from the beginning, as well as by Dr. Barros Oliveira (family physician).

As it was possible to test the application not only by a pediatrician but also by a family physician, it was found that the application had a lot of features that make more sense in the hospital setting. This realization was followed by a suggestion of an extra feature that is frequently performed by family physicians when seeing a child patient: the calculation of pediatric doses of the most common used medicines in the primary care setting, some of them also used in the Hospital. With this feature, the application is helpful for the hospital and non-hospital setting of pediatrics care.

Once the application was successfully finished in time, and there was availability by the family physician to explain this particular feature and test it on the mobile application, it is also part of the final application, and is called “Cálculo de doses pediátricas”.

6.4 The final application

This application has a set of core functionalities that were organized in different categories in order to make it easier to find a specific tool. In order to faster the final user adaptation to this application, each section has an explanation of its purpose.

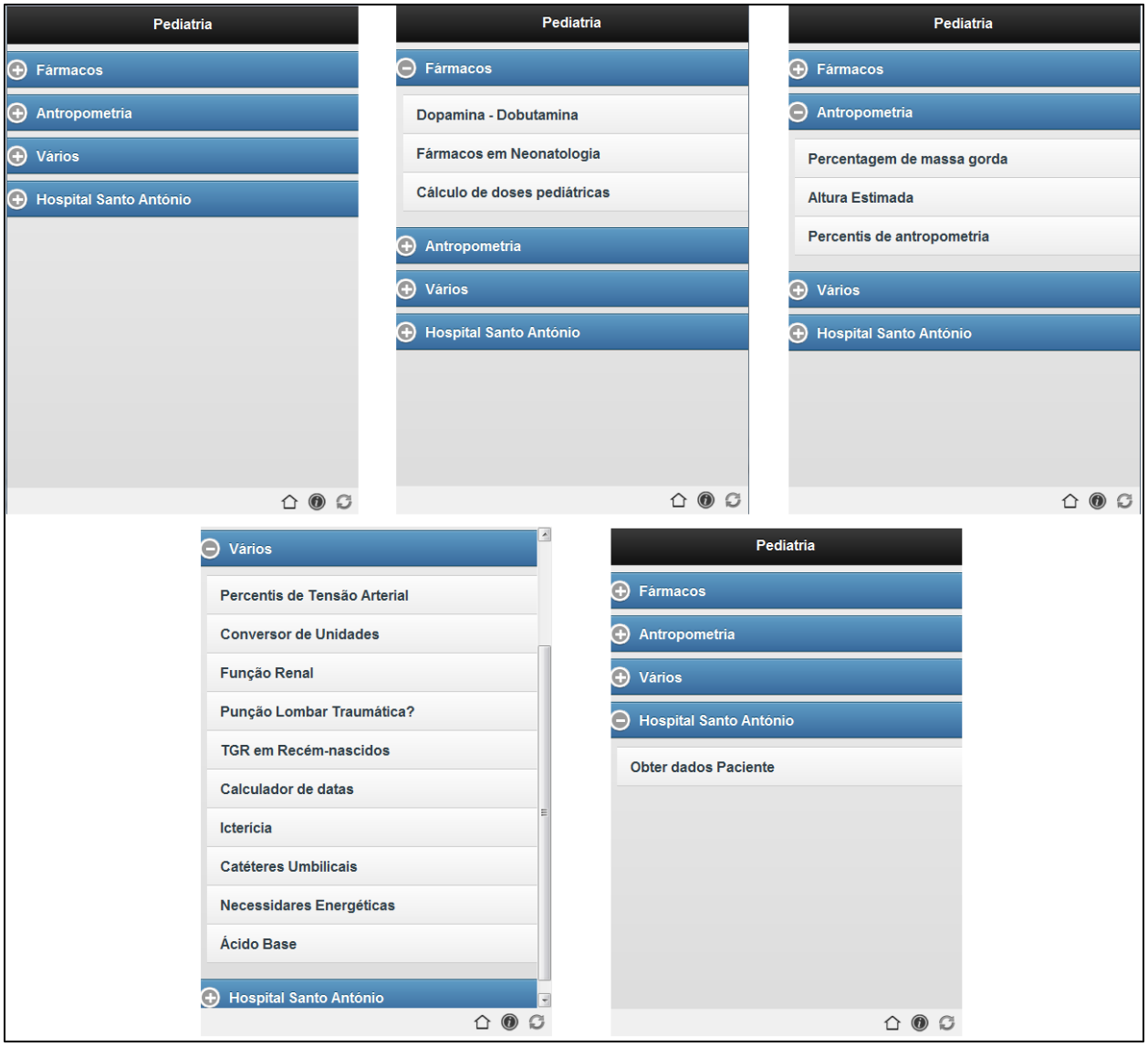


Figure 5.2 – This figure presents the application sections, and the features that fit in each of them.

As it is shown in figure 5.1 the application has four distinct sections:

- "Fármacos";
- "Antropometria";
- "Vários";
- "Hospital Santo António".

In the first section, there are three functionalities related with drugs dosages administration, which are:

- "Dopamina - Dobutamina" (Requirement 15);
- "Fármacos em Neonatologia" (Requirement 15);
- "Cálculo de doses pediátricas" (Extra functionality).

The second section has three functionalities related with anthropometric data. Those functionalities are:

- "Percentagem de massa gorda" (Requirement 14);
- "Altura estimada" (Requirement 13);
- "Percentis de Antropometria" (Requirements 11 & 12).

The next section has ten functionalities that belong to several areas:

- "Percentis de Tensão Arterial" (Requirement 1);
- "Conversor de Unidades" (Requirement 2);
- "Função Renal" (Requirement 3);
- "Punção Lombar Traumática?" (Requirement 9);
- "Transfusão de Glóbulos Rubros em Recém-nascidos" (Requirement 10);
- "Calculador de datas" (Requirement 8);
- "Icterícia" (Requirement 5);
- "Cateteres Umbilicais" (Requirement 4);
- "Necessidades Energéticas" (Requirement 7);
- "Ácido Base" (Requirement 6).

In the last section, there is a functionality that is specific to Santo António Hospital:

- "Obter dados Paciente" (Requirement 16).

In Appendix III there are presented pictures with all the steps of each functionality. The application is fully functional for both Android (version 2.2 or above) and iOS devices (version 6.1 or above). With the Android and iOS fragmentation it was not possible to test the application in every of its versions, however the

application is fully functional in every device it was tested until the date of this project delivery. It is also possible to launch the application for the Windows Phone and Backberry devices, although it still needs some adjustments in order to be fully compatible.

6.5 Questionnaire

It was created a questionnaire about the application, in order to get an evaluation of the app from its final users. The questionnaire also collects suggestions to improve the application in the future. After the release of the application this questionnaire was made available to the physicians that cooperated with this project as well as released in the Portuguese neonatologists forum (http://lusoneo.portugueseforum.net/).

At the moment of the project delivery, responses were still being accepted. The questionnaire can be found in Appendix IV.

7 Conclusion

This project had the initial objective of creating a mobile application for aiding pediatricians performing their work at the point of care. The necessity of such an application was identified by a pediatrician. There are several reasons for the usefulness of this application, such as: the wide variety of development stages and weights of the patient, drugs that don't come with pediatric presentation requiring additional calculations, the necessity of performing units conversions and performing complex calculations with very small values.

In order to better understand the pediatrics domain knowledge and know what kind of applications are already available to pediatricians, it was made a search for similar applications. Applications like Epocrates and PAEDS ED clearly show how delicate some of the pediatricians tasks are while attempting to aid them in their daily work. That work is available in Chapter 2 (State of the Art).

It was still necessary to make sure the application was going to have the functionalities that mattered the most for pediatricians in a mobile device. In order to achieve that, there was made a study on gathering and prioritizing requirements. There is already a field called "Requirements Engineering", that among others topics include the "Requirements elicitation" and "Requirements prioritization" study areas. Both these areas were addressed in Chapter 3 and were chosen the elicitation and prioritization techniques and tools that were found to be best choices for this particular project. To elicit the requirements there were used three techniques: Interviewing, Domain Analysis and Apprenticing. The interviewing was made with a pediatrician of Santo António Hospital and allowed to get an insight of what a pediatrician needed in a smartphone, but also to get a quick explanation of the necessary pediatric concepts to understand the requirements context. Domain analysis and Apprenticing were made with the Sabichão application and gave a clear understanding of what the application is for and how it worked. In order to document the gathered requirements it was used the Volere shell cards, which are part of the widely used Volere requirements specification. Only the shell cards were used because the main goal was not to create a

rigorous document about the application, but instead to register the requirements in a simple and objective way.

The prioritization of the gathered requirements was made with the 100-Dollar test technique. It is a simple technique to understand and apply. The aspect used with this technique was the importance. The 100-Dollar test allowed the stakeholder not only to prioritize the requirements but also to demonstrate how important each requirement really was. Because there were some requirements with the same value, it was used the cost/time aspect to prioritize these specific cases. To differentiate those situations, it was decided to favor the requirements that were expected to take less time to develop, in order to achieve the best cost/value solution possible. Even if the application was not 100% completed, this approach tries to maximize the number of implemented requirements while at the same time selecting first the most important ones.

At this point, it was still necessary to decide which mobile platforms the application was going to be developed and what framework(s) to use to achieve that. As a consequence, it was made a brief study on the recent behavior of the smartphone market share as well as on some projections for the following years. In 2012 the top three smartphone operating systems with the biggest market share were: Android, iOS and Blackberry. These 3 Operating Systems had 90.5% of the whole market. On the other hand, the projections for the following years include on the top 3 Android, iOS and Windows Phone, with position and percentages variations. Both the studied projections expect a fragmented smartphone market in the future as well.

In order to develop an application that could be ready to be launched as quickly as possible for most of the today's and tomorrow's smartphones on the market, it was decided to develop the application in a cross-platform development framework.

There were studied a few mobile cross-platform development frameworks that allowed freely applications development. Those frameworks are: Rhodes, PhoneGap, Titanium and Adobe Flex. The choice of the framework to develop the pediatric application was based on several aspects, but the most important ones were: a framework that was not found any limitation that could compromise the development of the gathered requirements, and a framework that was capable of compiling the final application to the top

smartphone market share operating systems. Then there were considered other aspects, such as: the available documentation of the framework or the activity of the frameworks community online. The final choice was the Rhodes framework because it was not found any limitation to develop the application, it was possible to compile the application to the top smartphone OSs in the 2012 market share analysis as well as in the projections for the next years, it has a good documentation and an active community online, among other factors.

In the implementation phase, it was used the work that was already done during the gathering and prioritization phase, being already available the requirements in the Volere shell cards and its development priority already defined. It was also necessary to go throgh a learning phase of the new technologies involved in this project, such as: Ruby, Ruby on Rails, Rhodes and jQuery Mobile. It was important to spend some time learning each of the involved technologies, in order to develop the application without many technical knowledge related issues on the way. During the implementation phase, this project was being followed by a pediatrician that saw its evolution and tested the application, which lead to a refinement of the application functionalities over time. All of the gathered requirements were successfully implemented.

After the application development was finished, the application was tested in order to find bugs and perform final adjustments. These tests were made not only by a pediatrician but also by a family physician that suggested an extra functionality family physicians perform a lot while taking care of children. With the cooperation of the family physician, this feature was also included in the application. As a consequence the application is not only for pediatricians but also for family physicians.

When the application was released a questionnaire was also made available, in order to evaluate and improve the application in the future. At the moment of the project delivery, responses were still being accepted.

The development of this project led to the writing of two articles, both accepted to be presented in conferences. The article “Step Towards M-Health In Pediatrics” was accepted in the HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies; while the article “Analysis of Cross-Platform Development Frameworks for a Smartphone Pediatric Application” was

accepted in the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM2013).

As future work, it is necessary to maintain the application through time by correcting the bugs when they are found or adding new useful features if suggested. The application can be translated to the English language, in order to target a wider audience. The application can also be launched to the Windows Phone and Blackberry devices, especially if it is reported interest for those markets.

8 Bibliography

Allen, S., Graupera, V. & Lundrigan, L., 2010. *Pro Smartphone Cross-Platform Development*, Berkeley, CA: Apress.

Anderson, J., 2013. *Appcelerator Titanium: Up and Running*, O'Reilly Media.

Anderson, J.G., 2011. *Beginning Flash, Flex, and AIR Development for Mobile Devices*, Wrox.

Appcelerator, 2013a. Appcelerator Titanium. Available at: http://www.appcelerator.com/ [Accessed May 6, 2013].

Appcelerator, 2013b. Titanium.Calendar. Available at: http://docs.appcelerator.com/titanium/latest/#!/api/Titanium.Calendar [Accessed April 21, 2013].

Apps, I., 2011a. Paeds ED iTunes Page. Available at: https://itunes.apple.com/gb/app/paeds-ed/id326619999?mt=8 [Accessed January 20, 2013].

Apps, I., 2011b. Peds ED lite iTunes Page. Available at: https://itunes.apple.com/gb/app/paeds-ed-lite/id331122046?mt=8 [Accessed January 20, 2013].

Arcuri, P., 2013. Rhomobile vs. rhodes. Available at: https://developer.motorolasolutions.com/message/7880#7880 [Accessed May 17, 2013].

Aybüke Aurum, C.W., 2005. *Engineering and Managing Software Requirements*, Springer.

Bhansali, R. & Armstrong, J., 2012. Smartphone applications for pediatric anesthesia. *Paediatric anaesthesia*, 22(4), pp.400–4.

Community, P., 2012. Phonegap calendar. Available at: http://community.phonegap.com/nitobi/topics/phonegap_calendar [Accessed April 21, 2013].

Douglas K. Richardson, John D. Corcoran, Gabriel J. Escobar, S.K.L., 2001. SNAP-II and SNAPPE-II: Simplified newborn illness severity and mortality risk scores. *The Journal of Pediatrics*, 138(1), pp.92–100.

Engelhart, K., 2010. Killer Apps that save lives. Available at: http://www2.macleans.ca/2010/03/18/killer-apps-that-save-lives/ [Accessed December 20, 2012].

Epocrates, 2013a. Epocrates RX iTunes Page. Available at: https://itunes.apple.com/us/app/epocrates/id281935788?mt=8 [Accessed February 3, 2013].

Epocrates, I., 2013b. Epocrates Content Sources Page. Available at: http://www.epocrates.com/clinical/sources [Accessed February 3, 2013].

Epocrates, I., 2013c. Epocrates essentials Official Page. Available at: http://www.epocrates.com/mobile/iphone/essentials [Accessed February 3, 2013].

Federal, U. et al., 2011. Erros de medicação em Pediatria. , 64(3), pp.563–569.

Freedman, E., 2007. UGH! 3 MILLION PLATFORMS WITH 100 USERS EACH. Available at: http://eliainsider.com/2007/10/21/ugh-3-million-platforms-with-100-users-each/ [Accessed April 18, 2013].

Garside, J., 2013. Firefox maker Mozilla to launch smartphone operating system. Available at: http://www.guardian.co.uk/technology/2013/feb/22/firefox-mozilla-smartphone-operating-system [Accessed April 18, 2013].

Hartmann, G., 2011. Cross-platform mobile development. , (March), pp.1–18.

Inc., E., 2013. Epocrates RX Official Page. Available at: http://www.epocrates.com/mobile/iphone/rx [Accessed February 3, 2013].

Isabelhealthcare, 2012a. Isabel User Guide 2012. , pp.1–4.

Isabelhealthcare, 2012b. The Isabel System Overview. , p.http://www.isabelhealthcare.com/home/products.

James E. Gray, Douglas K. Richardson, Marie C. McCormick, Kathryn Workman-Daniels, D.A.G., 1992. Neonatal Therapeutic Intervention Scoring System: A Therapy-Based Severity-of-Illness Index. , 90(4), pp.561 –567.

Karam, D.O., Drug Doses Official Page. Available at: http://drugdoses.net/ [Accessed February 1, 2013].

Karam, O., 2012. Drug Doses iTunes Page. Available at: https://itunes.apple.com/app/drugdoses/id322681972?mt=8 [Accessed February 1, 2013].

Kaushal, R. et al., 2001. Medication Errors and Adverse Drug Events in Pediatric Inpatients. , 285(16), pp.2114–2120.

Lee, J., 2013. Samsung Will Release Tizen-Based Smartphone This Year. Available at: http://www.bloomberg.com/news/2013-03-14/samsung-will-release-tizen-based-smartphone-this-year.html [Accessed April 18, 2013].

Manicone, P.E. et al., 2011. DIAGNOSTIC DILEMMAS IN PEDIATRICS: HOW DOES A WEB BASED DIAGNOSTIC CHECKLIST PERFORM? , 1(1), p.2.

Mosa, A.S.M., Yoo, I. & Sheets, L., 2012. A systematic review of healthcare applications for smartphones. *BMC medical informatics and decision making*, 12(1), p.67.

Network, T.I.N., 1993. The CRIB (clinical risk index for babies) score: a tool for assessing initial neonatal risk and comparing performance of neonatal intensive care units. The International Neonatal. Network. *Lancet.*, (342), pp.193–198.

Paananen, T., 2011. *SMARTPHONE CROSS-PLATFORM FRAMEWORKS*. JAMK University of Applied Sciences.

Paczkowski, J., 2013. Android's Leaky Bucket: Loyalty Gives Apple the Edge Over Time. Available at: http://allthingsd.com/20130426/androids-leaky-bucket-loyalty-gives-apple-the-edge-over-time/ [Accessed April 29, 2013].

Ramnarayan, P & Britto, J, 2002. Paediatric clinical decision support systems. *Archives of disease in childhood*, 87(5), pp.361–2.

Ramnarayan, Padmanabhan et al., 2007. Validation of a diagnostic reminder system in emergency medicine: a multi-centre study. *Emergency medicine journal: EMJ*, 24(9), pp.619–24.

Rohit Ghatol, Y.P., 2012. *Beginning PhoneGap: Mobile Web Framework for JavaScript and HTML5*, Apress.

Solutions, M., 2013. RhoMobile Suite documentation center. Available at: http://docs.rhomobile.com/.

Sommerville, I., 2010. *Software Engineering*, Addison-Wesley.

Spritle, 2011. Consume SOAP from Rhodes using raw SOAP Request - Initial code. Available at: https://github.com/spritle/rhodes_soap_test/blob/master/app/Soap/soap_controller.rb.

Suzanne Robertson, J.C.R., 2006. *Mastering the Requirements Process*, Addison-Wesley Professional.

Systems, A., 2013. PhoneGap Website. Available at: http://phonegap.com/ [Accessed May 4, 2013].

Tanis R Fenton, J.H.K., 2013. A systematic review and meta-analysis to revise the Fenton growth chart for preterm infants. *BMC Pediatrics*, 13(59), p.13.

Trenholm, R., 2013. Ubuntu for phones delayed until 2014, but on Nexus 4 now. Available at: http://crave.cnet.co.uk/mobiles/ubuntu-for-phones-delayed-until-2014-but-on-nexus-4-now-50010475/ [Accessed April 18, 2013].

Appendix

Appendix I – Elicited requirements in the Volere Requirements shell

Requirement #: 1 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: Given the age, height and sex of an under 18 year's old child, the system shall provide the values of arterial hypertension tables.

Rationale: The checking of the arterial hypertension values by the pediatricians, require more time if done by hand.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the blood pressure module of Sabichão.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão

History:

Requirement #: 2 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall provide a section to convert the most commonly used units by pediatricians.

Rationale: Pediatricians need to convert units (e.g. for medicines administration) many times.

Originator: Dr. Simão.

Fit Criterion: The system shall provide the same functionalities that the conversion module of Sabichão. provides.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão

History:

Requirement #: 3 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall provide a section to perform renal function related calculations.

Rationale: The diagnosis of renal function is a task often performed by pediatricians that involve a certain amount of calculations that may be automated.

Originator: Dr. Simão.

Fit Criterion: The system shall provide the same functionalities that the renal function module of Sabichão offers.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão

History:

Requirement #: 4 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall provide a visual image of the umbilical and arterial venous catheterization that shows the position of the catheter in the patient's body during the procedure.

Rationale: The umbilical and arterial venous catheterization is a complex task to do, because its performing varies with several patient characteristics.

Originator: Dr. Simão.

Fit Criterion: The system shall provide the same functionalities that the umbilical and arterial venous catheterization module of Sabichão offers.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão

History:

Requirement #: 5 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall help the pediatricians interpreting the results of jaundice exams and in the decision of additional procedures to perform if needed.

Rationale: The interpretation of the exams and decision of additional procedures to perform involve several mathematical operations that may be automated.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the jaundice module of Sabichão.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão

History:

Requirement #: 6 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: Given one of three elements (pH, CO2 and HCO3-), the system shall indicate the normal values of the others elements. The system shall also display information about respiratory alkalosis.

Rationale: This functionality is useful mainly for teaching purposes.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the Acid – Base module of Sabichão.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão

History:

Requirement #: 7 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall calculate the post-surgical energy needs of a patient.

Rationale: The checking of the energy needs of a patient is a commonly performed task, that requires performing mathematical operations.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the energy needs module of Sabichão.

Customer Satisfaction: 3 **Customer Dissatisfaction:** 3 **Conflicts:**

Priority:

Supporting Materials: Sabichão

History:

Requirement #: 8 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall have a calculator of dates that provides the calendar of a given year. Given the gestational age and birth date of the patient, it is provided the date of: last menstrual period, estimated birth date, today's corrected age and postmenstrual age.

Rationale: These are calculations that are commonly done by pediatricians.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the dates calculation module of Sabichão.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão.

History:

Requirement #: 9 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall provide a tool to aid pediatricians in a traumatic lumbar puncture.

Rationale: When a pediatrician suspects a patient has meningitis it is necessary to check the number of leukocytes, and there are some mathematical operations involved that may me automated.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the traumatic lumbar puncture module of Sabichão.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão.

History:

Requirement #: 10 **Requirement Type:** 1 **Event/Use Cases #'s:**

Description: The system shall provide a tool to supports pediatricians in the decision of performing an erythrocytes transfusion in newborns or not.

Rationale: The decision process of an erythrocytes transfusion has several calculations as well as some clinical factors that must be verified as true or not.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the erythrocytes transfusion in newborns module of Sabichão.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão.

History:

Requirement #: 11 **Requirement Type:** **Event/Use Cases #'s:**

Description: The system shall provide a section to calculate the estimated weight, length and cephalic perimeter of a baby with his gestational age between 22 and 44 weeks.

Rationale: It is a common calculation for pediatricians that may be automated.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the percentiles of weight, length and cephalic perimeter module of Sabichão.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão and http://www.peditools.org/.

History:

Requirement #: 12 **Requirement Type:** **Event/Use Cases #'s:**

Description: The system shall provide a section to calculate the estimated weight and height of a patient with his age between 0 and 17 years old.

Rationale: It is a common calculation for pediatricians that may be automated.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the percentiles of weight and height module of Sabichão.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão and http://www.peditools.org/.

History:

Requirement #: 13 **Requirement Type:** **Event/Use Cases #'s:**

Description: Given the baby gender and parents height, the system shall provide the estimated height of the patient.

Rationale: Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the estimated height module of Sabichão.

Customer Satisfaction: 3 **Customer Dissatisfaction:** 2 **Conflicts:**

Priority:

Supporting Materials: Sabichão.

History:

Requirement #: 14 **Requirement Type:** **Event/Use Cases #'s:**

Description: The system shall provide a section to calculate the fat mass percentage of both male and female patients.

Rationale: It is a commonly performed task by pediatricians that requires mathematical calculations and may be automated.

Originator: Dr. Simão.

Fit Criterion: The outcome shall be equal to the fat mass percentage module of Sabichão.

Customer Satisfaction: 3 **Customer Dissatisfaction:** 2 **Conflicts:**

Priority:

Supporting Materials: Sabichão.

History:

Requirement #: 15 **Requirement Type:** **Event/Use Cases #'s:**

Description: The system shall have a section to provide administration information (dosage administration, interval between administrations, administration via and administration observations) of the most commonly used medicines available in NEOFAX 2011.

Rationale: The NEOFAX is a reference book for pediatrician usage.

Originator: Several pediatrician articles.

Fit Criterion: The outcome shall be equal to the NEOFAX 2011 module of Sabichão.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 1 **Conflicts:**

Priority:

Supporting Materials: Sabichão and NEOFAX 2011.

History:

Requirement #: 16 **Requirement Type:** **Event/Use Cases #'s:**

Description: The system shall provide a functionality that lets the pediatricians working in Hospital de Santo António request patient's data that are admitted to the Hospital.

Rationale: This functionality allows pediatricians working in the hospital to access to patient data, instead of having to insert it by hand every time it is necessary.

Originator: Questionnaire done to pediatricians that are members of the forum lusoneo.portugueseforum.net.

Fit Criterion: The system shall automatically fill the selected patient information in the sections that requires it.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 2 **Conflicts:**

Priority:

Supporting Materials:

History:

Appendix II – Requirements Prioritization

100 Euro Test

The 100-euro test is a very straightforward prioritization technique where the stakeholders are given 100 imaginary units (money, hours, etc.) to distribute between the requirements.

The result of the prioritization is presented on a ratio scale.

Requirements:	Invested €'s
Requirement 1	10
Requirement 2	9
Requirement 3	9
Requirement 4	6
Requirement 5	6
Requirement 6	6
Requirement 7	1
Requirement 8	6
Requirement 9	6
Requirement 10	6
Requirement 11	8
Requirement 12	8
Requirement 13	1
Requirement 14	1
Requirement 15	9
Requirement 16	8

Total Sum:	100

The final value should be equal to 100 €.

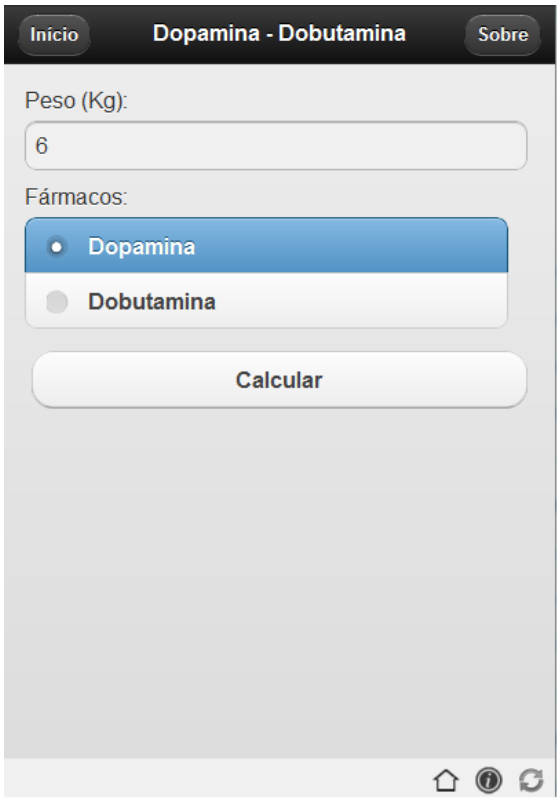
Requirements prioritized by the importance and cost/time aspects

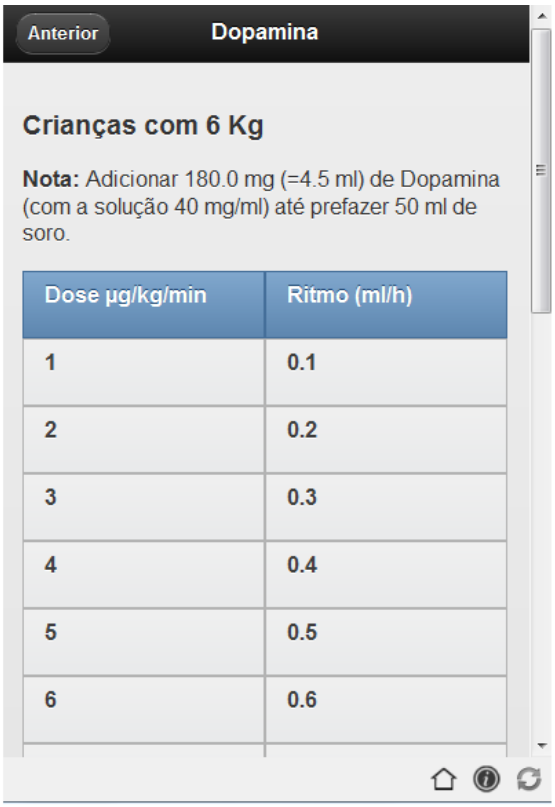
	Aspect		
	Importance	Cost/Time	
Requirements:	Invested €'s		
Requirement 1	10		
Requirement 3	9	1	
Requirement 2	9	2	
Requirement 15	9	3	
Requirement 16	8	1	
Requirement 11	8	2	Joined and updated during
Requirement 12	8	2	the development phase.
Requirement 8	6	1	
Requirement 9	6	2	
Requirement 10	6	3	
Requirement 5	6	4	
Requirement 4	6	5	
Requirement 6	6	6	
Requirement 13	1	1	
Requirement 14	1	2	
Requirement 7	1	3	

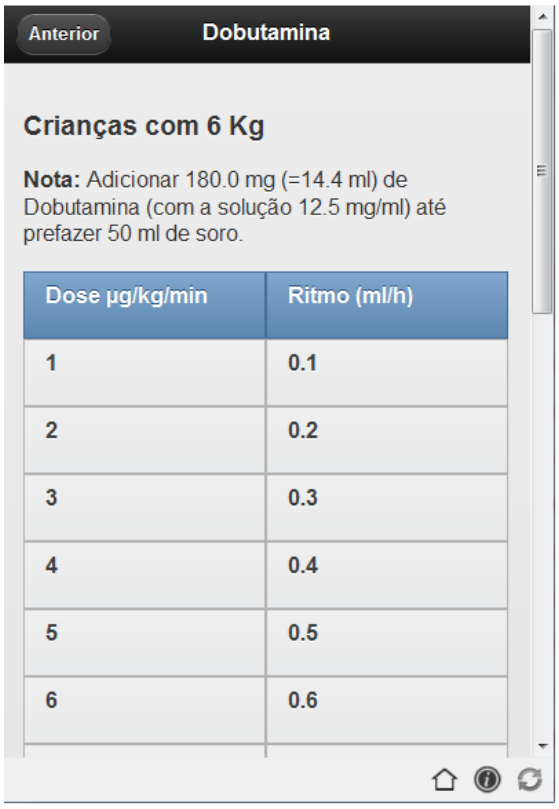
Appendix III – Application pictures by functionality

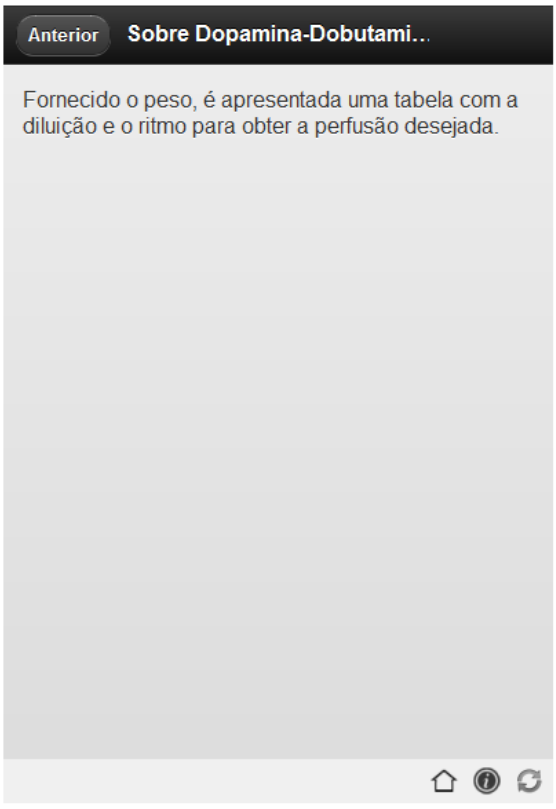
Note: The inserted data in the screenshots in this appendix is random and may not correspond to a possible scenario. The purpose of this appendix is to present each of the application functionalities and its different steps. The functionalities are presented in the same order found in the application.

Requirement 15 – "Dopamina - Dobutamina"

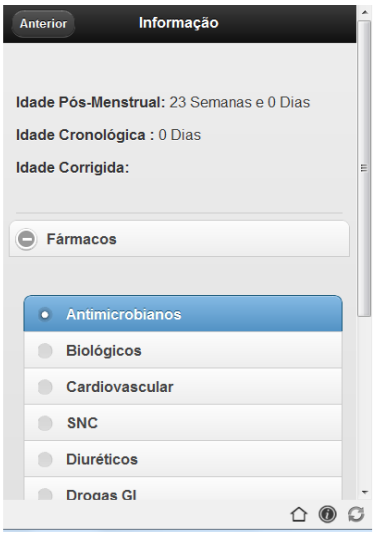


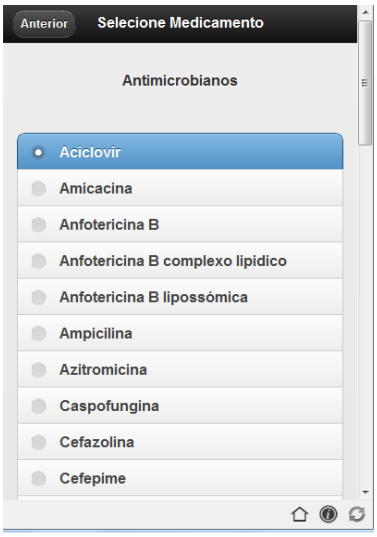


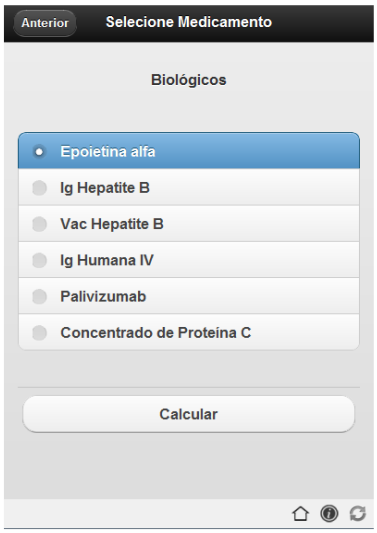


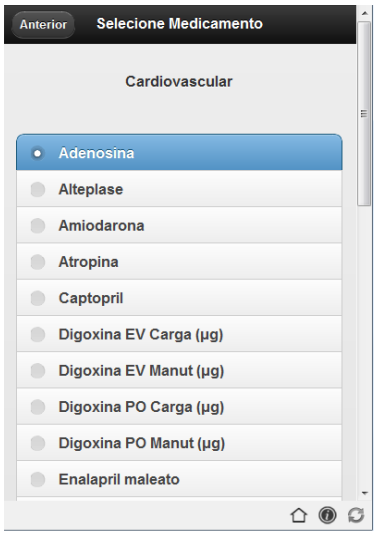


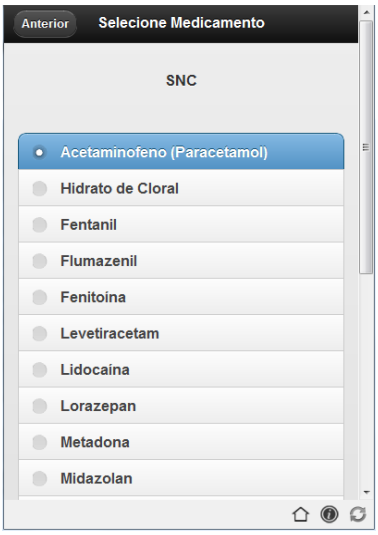
Requirement 15 - “Fármacos em Neonatologia”

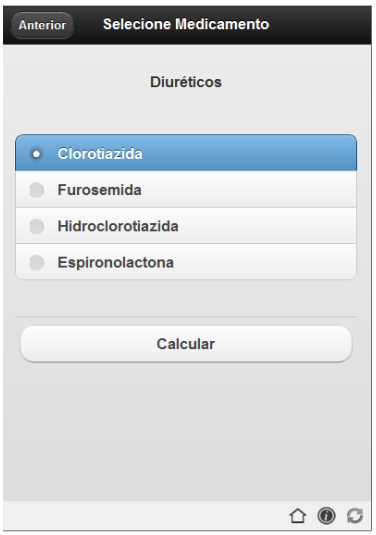


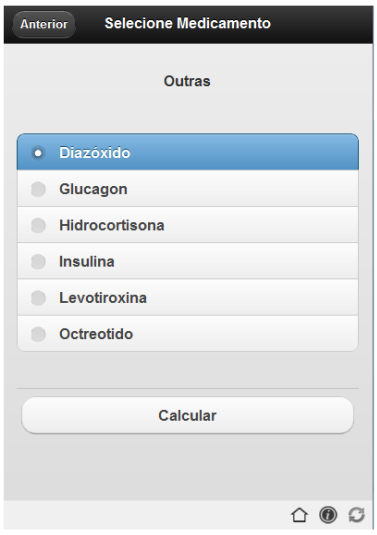


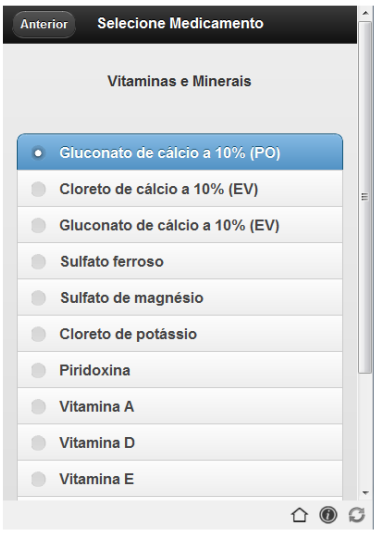




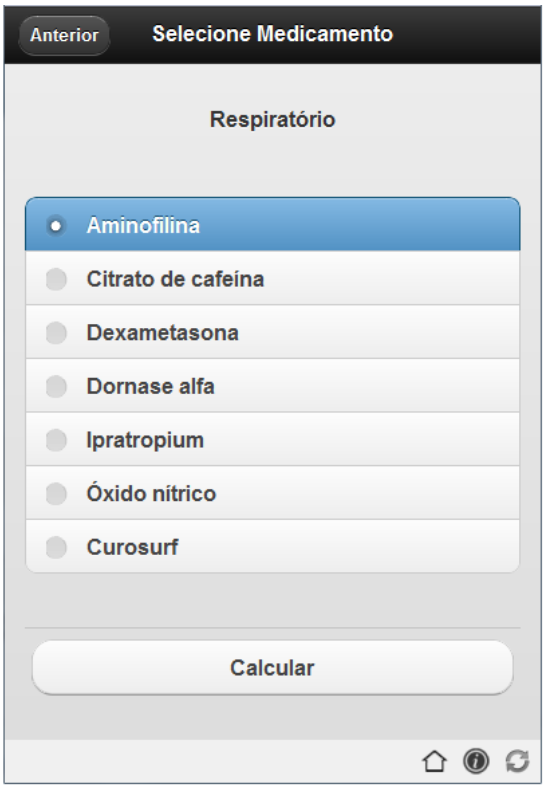


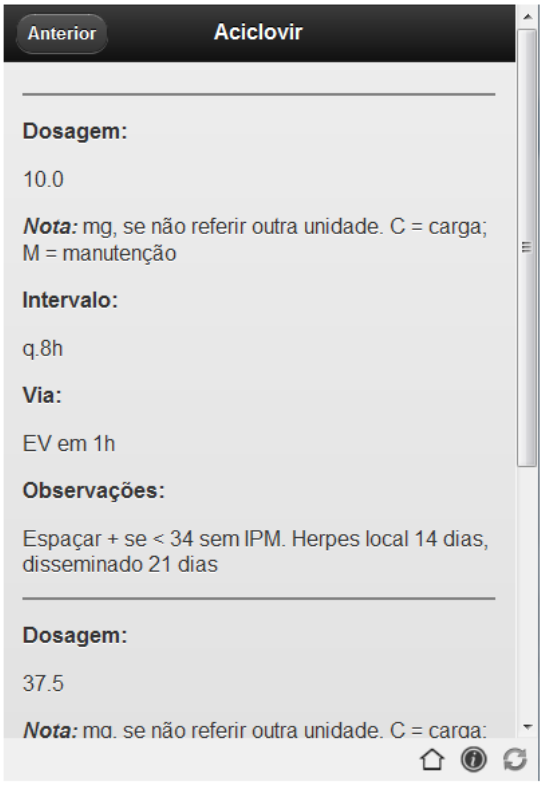


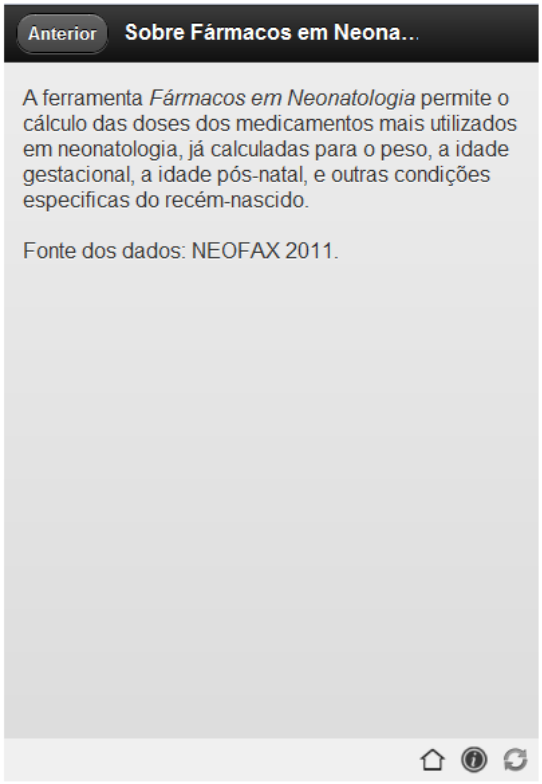




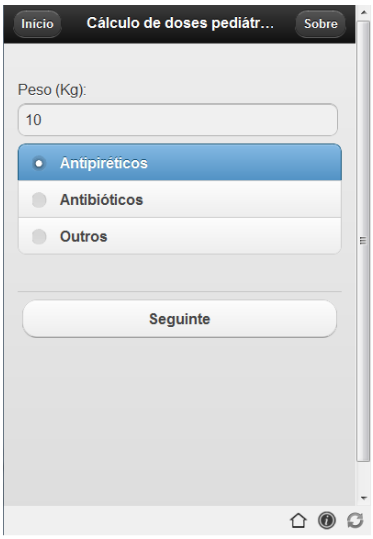

Anterior
Selecione Medicamento
Drogas GI
Cimetidina
Famotidina
Lansoprazol
Metoclopramida
Omeprazol
Ranitidina
Ursodiol
Calcular

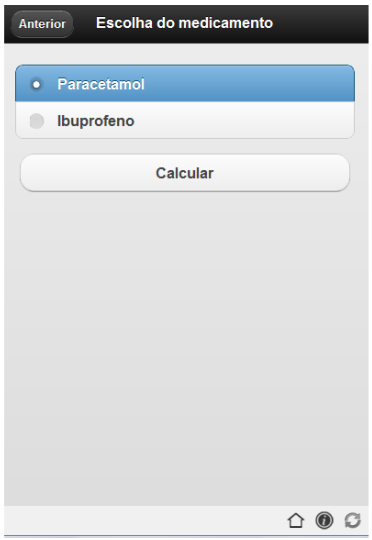

Anterior
Selecione Medicamento
Respiratório
Aminofilina
Citrato de cafeína
Dexametasona
Dornase alfa
Ipratropium
Óxido nítrico
Curosurf
Calcular

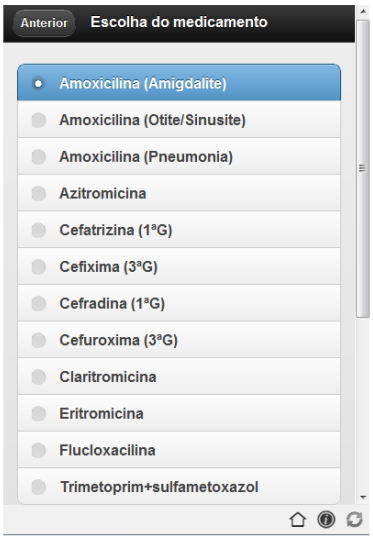

Anterior
Aciclovir
Dosagem:
10.0
Nota: mg, se não referir outra unidade. C = carga; M = manutenção
Intervalo:
q.8h
Via:
EV em 1h
Observações:
Espaçar + se < 34 sem IPM. Herpes local 14 dias, disseminado 21 dias
Dosagem:
37.5
Nota: mg, se não referir outra unidade. C = carga;

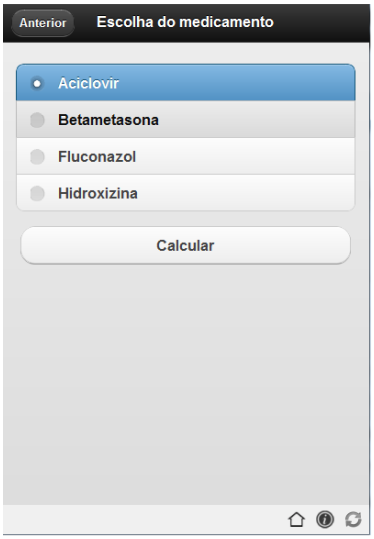

Anterior
Sobre Fármacos em Neona...
A ferramenta Fármacos em Neonatologia permite o cálculo das doses dos medicamentos mais utilizados em neonatologia, já calculadas para o peso, a idade gestacional, a idade pós-natal, e outras condições especificas do recém-nascido.
Fonte dos dados: NEOFAX 2011.

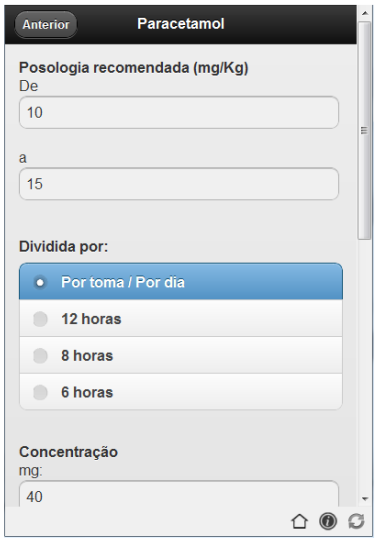
Extra Functionality - “Cálculo de doses pediátricas”

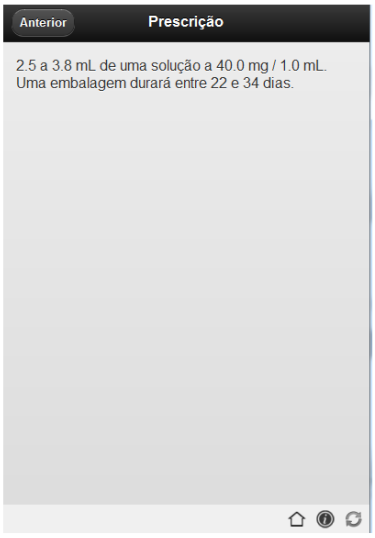


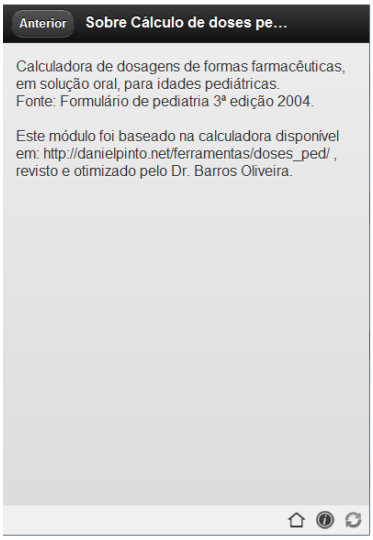




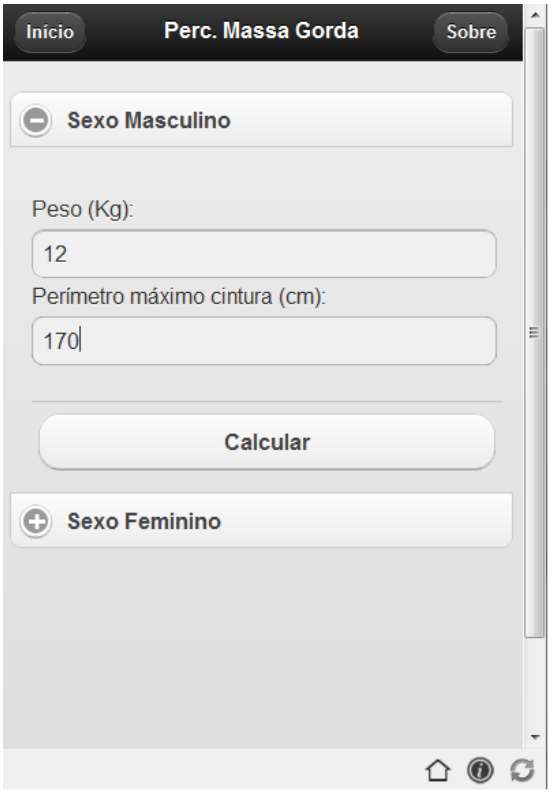


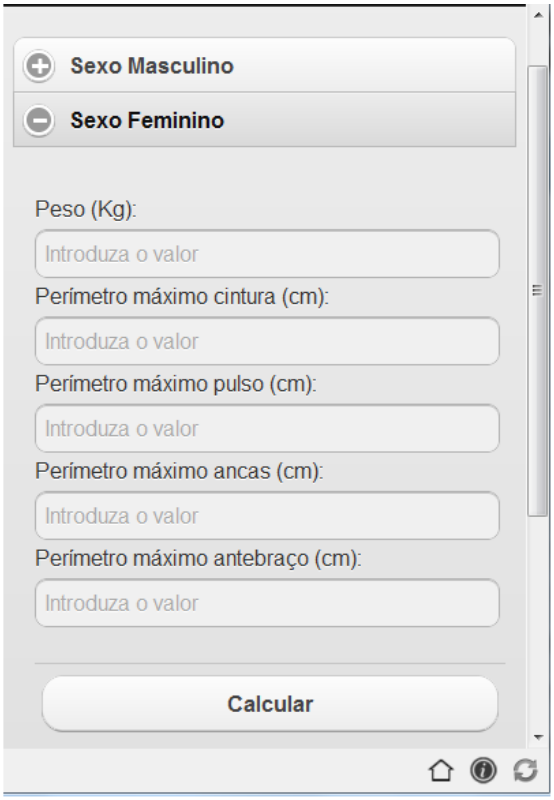


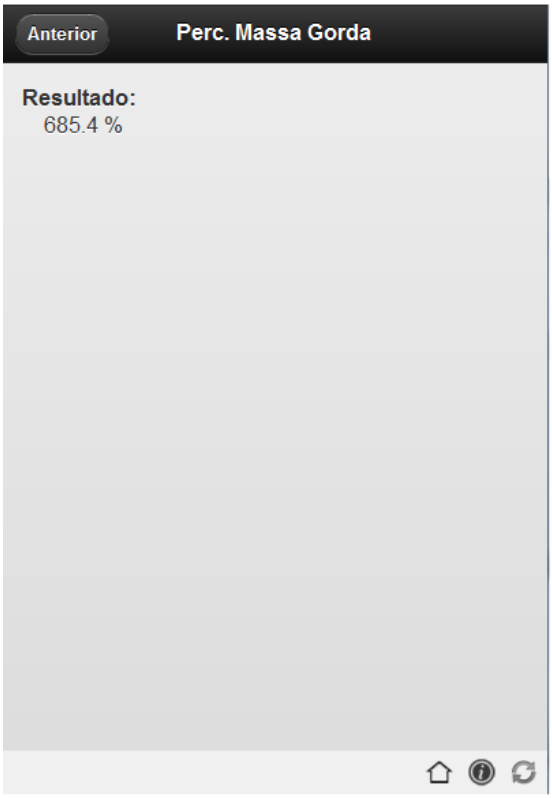


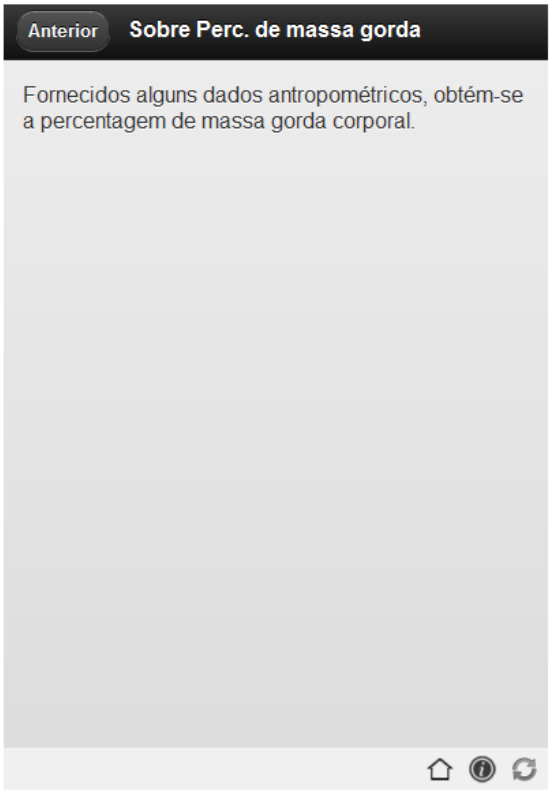


Requirement 14 - “Percentagem de massa gorda”

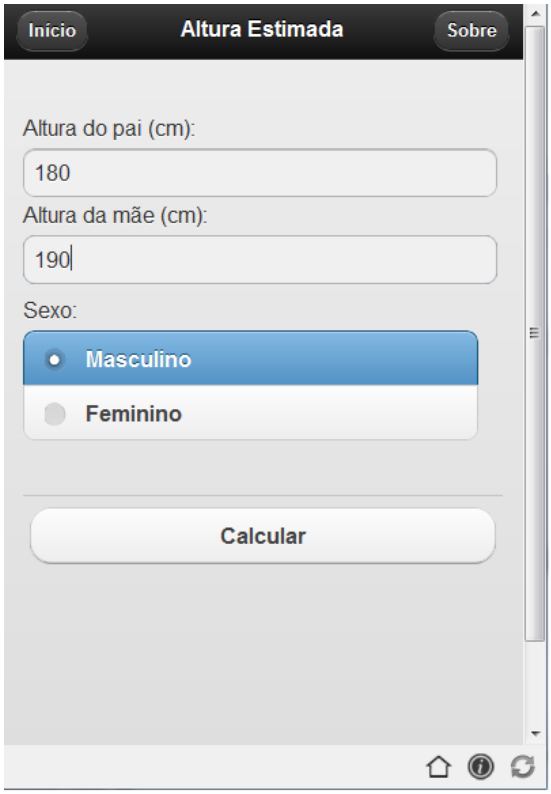


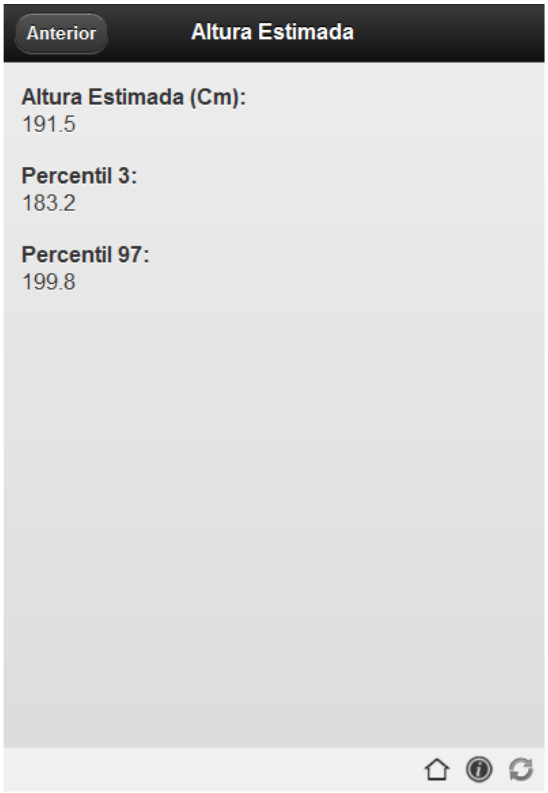


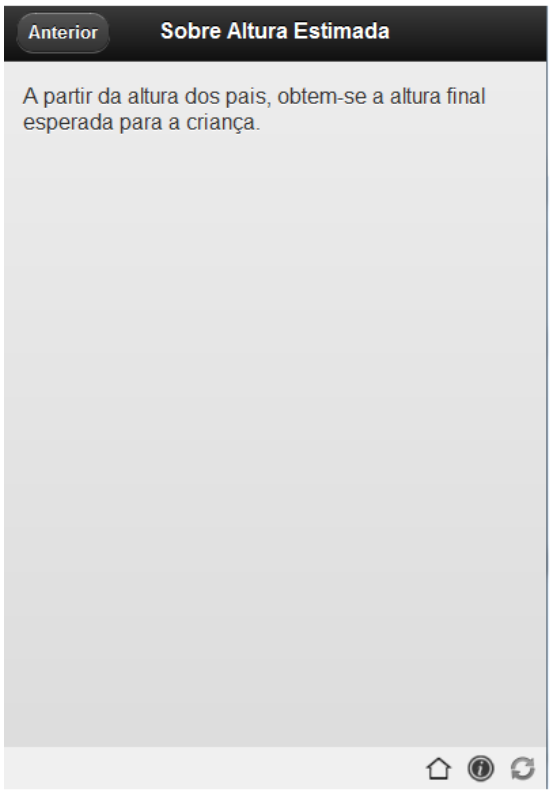




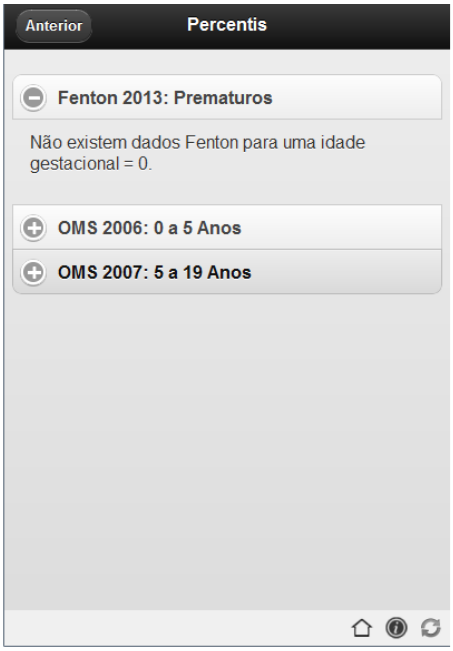
Requirement 13 - “Altura estimada”

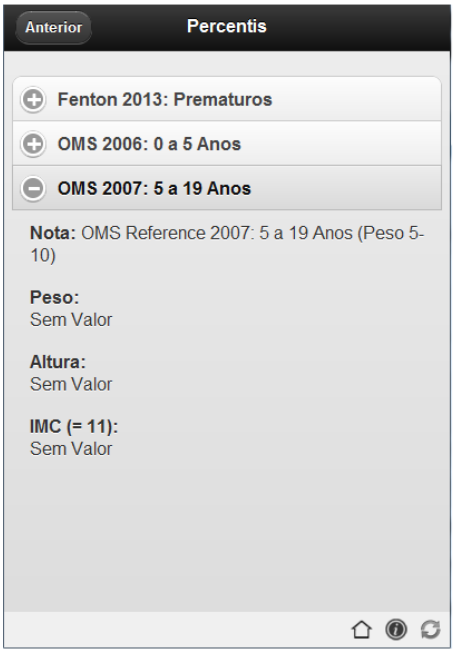


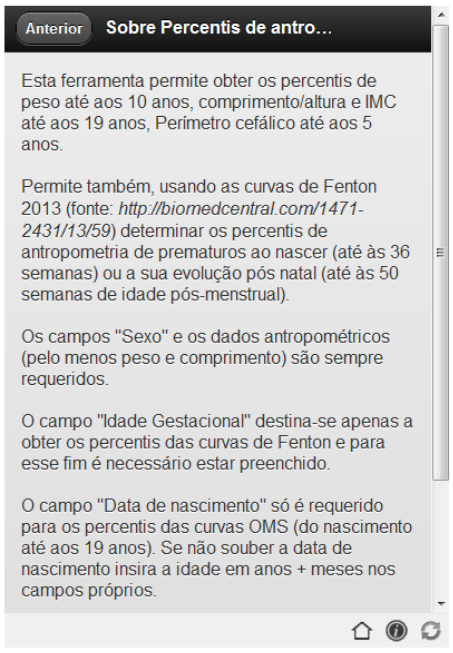




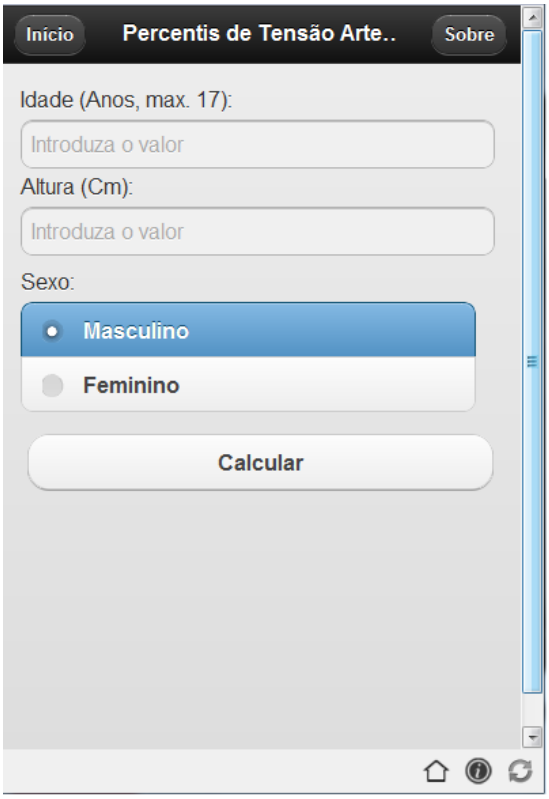
Requirements 11 & 12 - “Percentis de Antropometria”

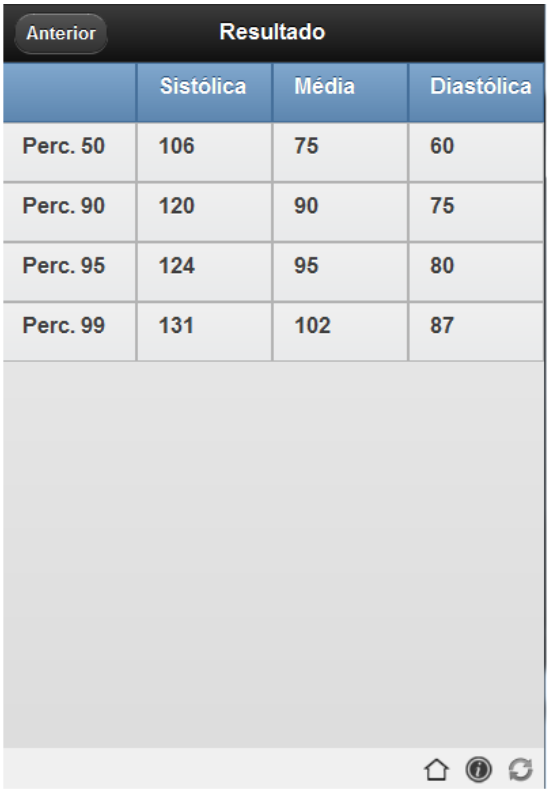


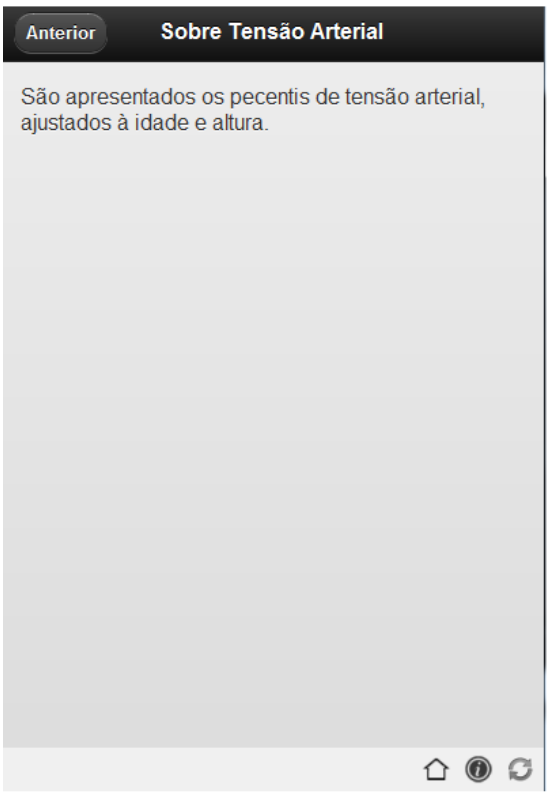




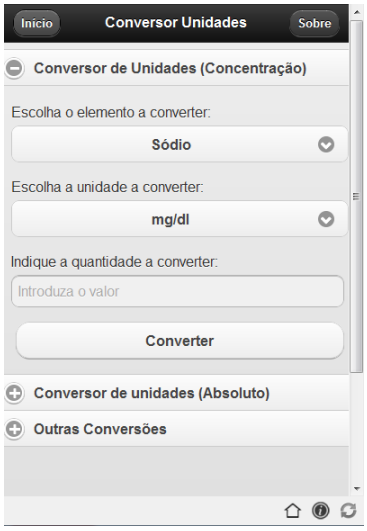
Requirement 1 - “Percentis de Tensão Arterial”

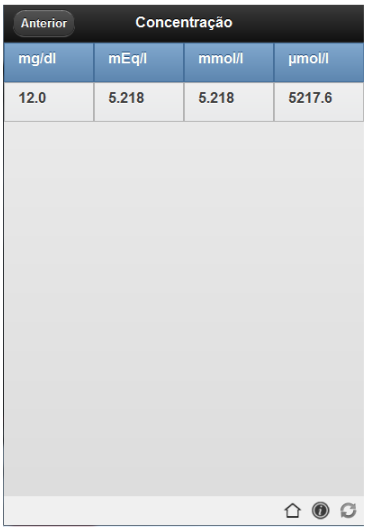


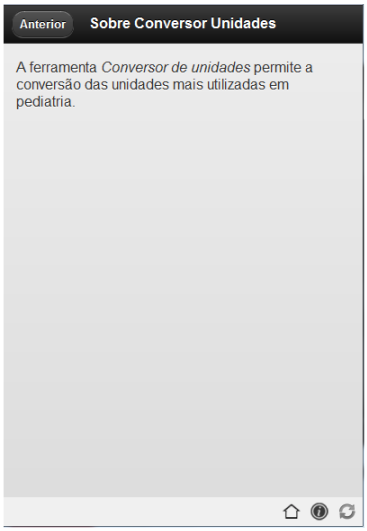


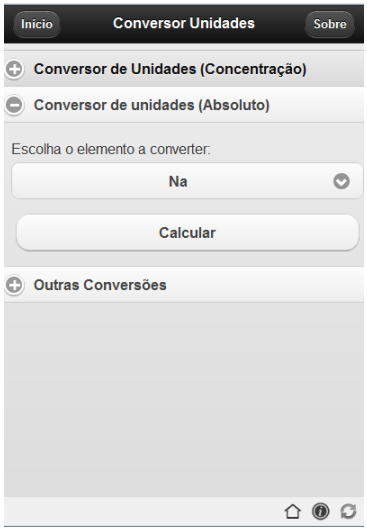


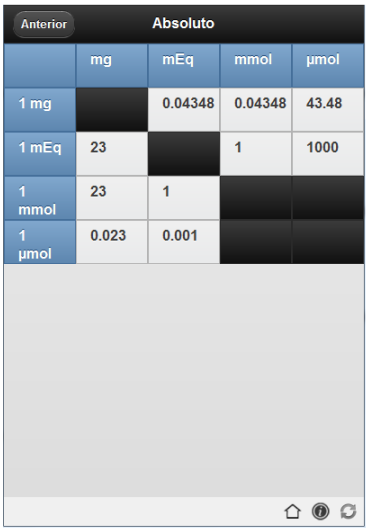
Requirement 2 - “Conversor de Unidades” ”

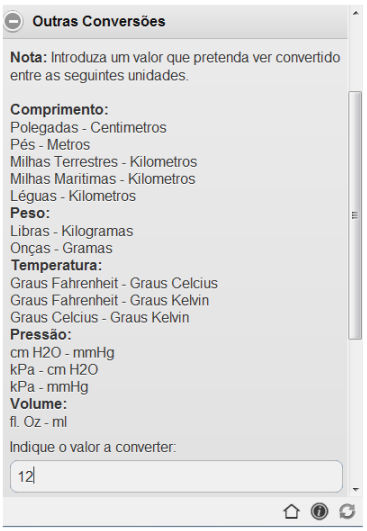


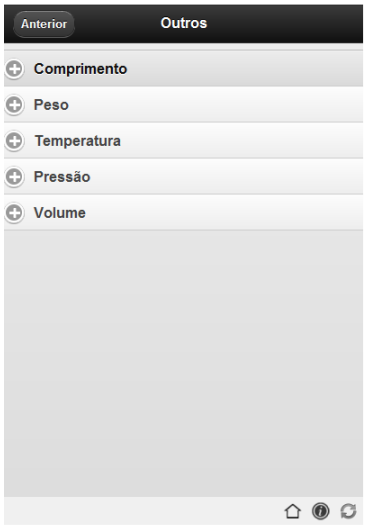




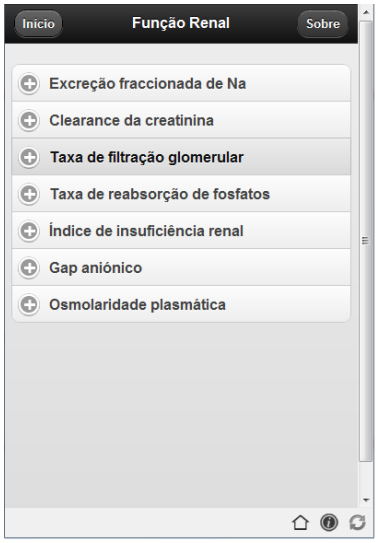


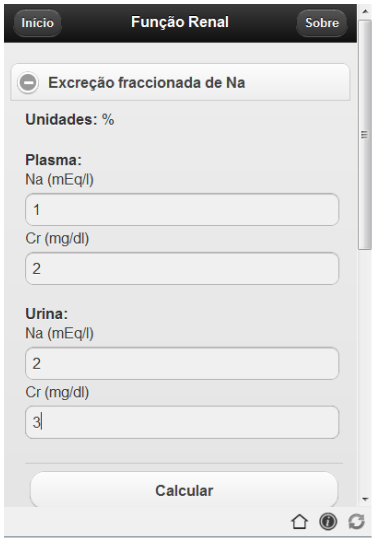


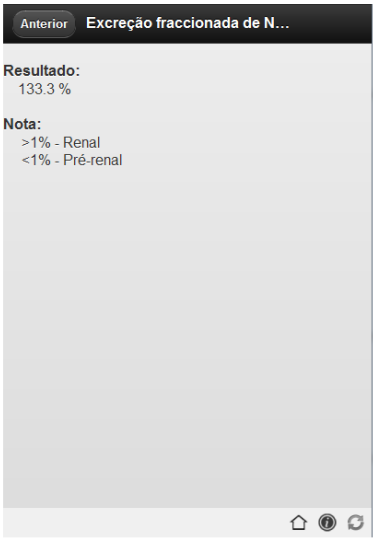


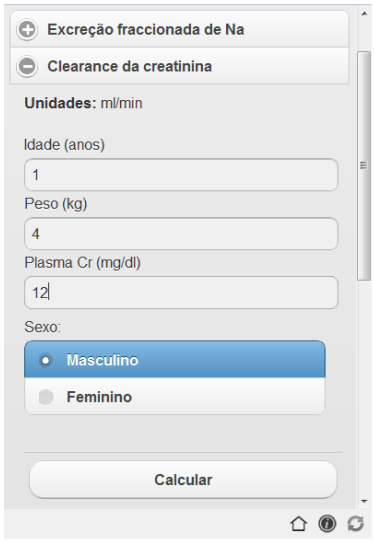


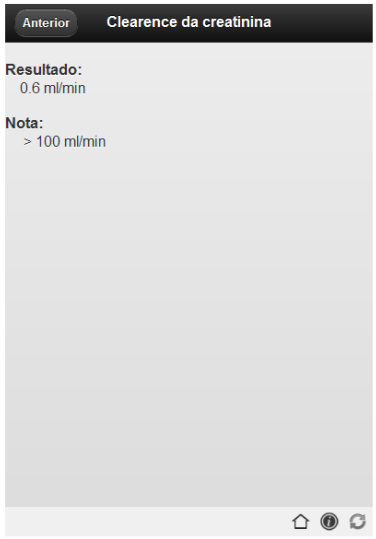
Requirement 3 - "Função Renal"

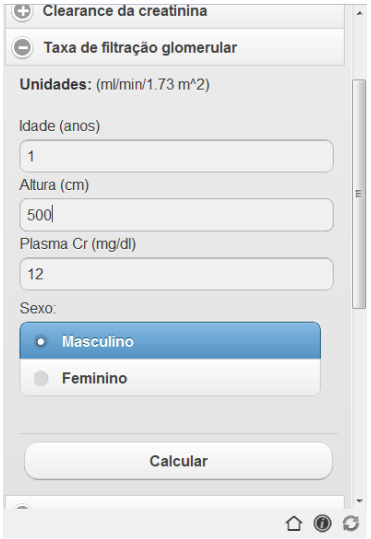


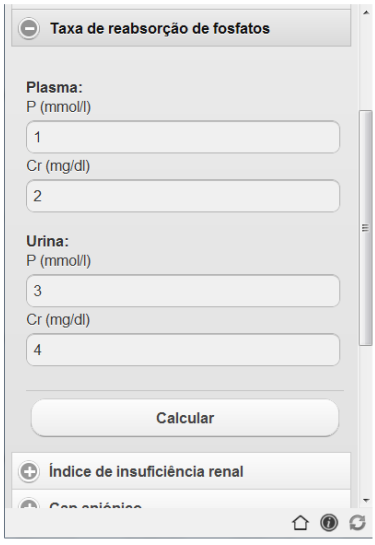


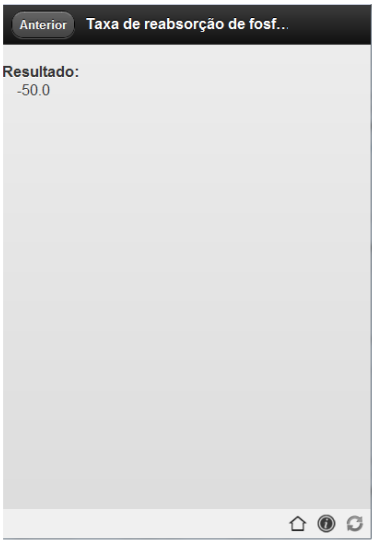


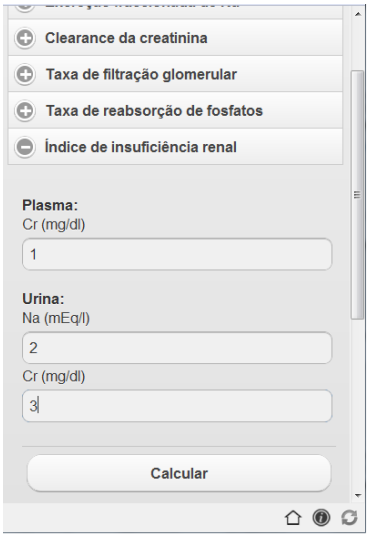


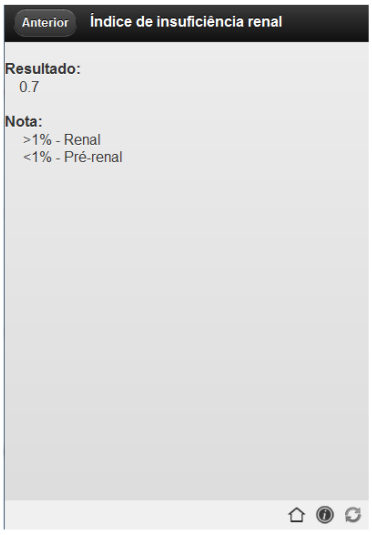


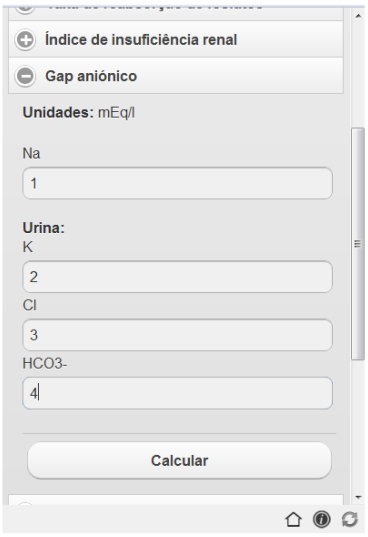


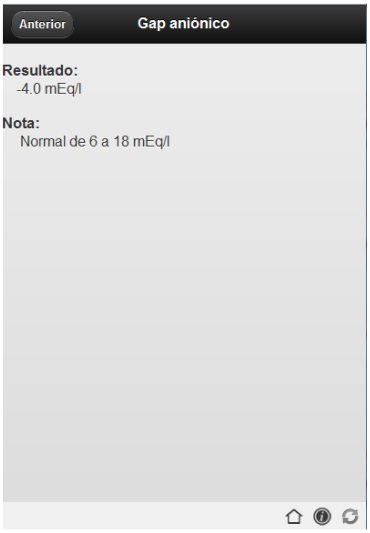


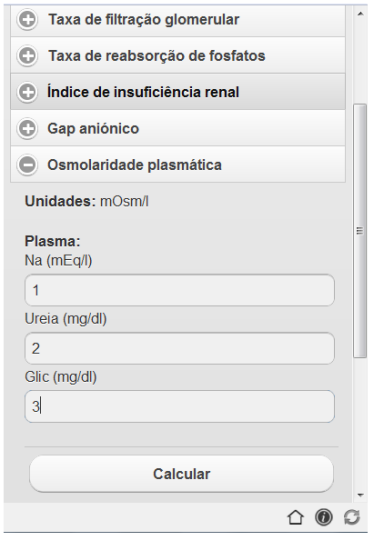


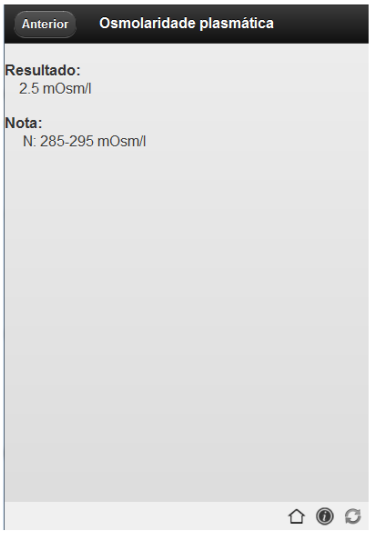

Clearance da creatinina
Taxa de filtração glomerular
Taxa de reabsorção de fosfatos
Índice de insuficiência renal
Plasma:
Cr (mg/dl)
1
Urina:
Na (mEq/l)
2
Cr (mg/dl)
3
Calcular

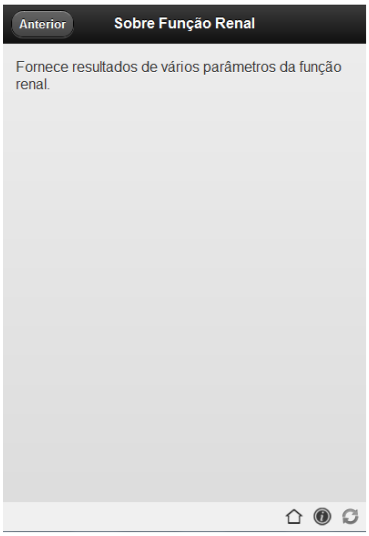

Anterior
Índice de insuficiência renal
Resultado:
0.7
Nota:
>1% - Renal
<1% - Pré-renal


Índice de insuficiência renal
Gap aniónico
Unidades: mEq/l
Na
1
Urina:
K
2
Cl
3
HCO3-
4
Calcular

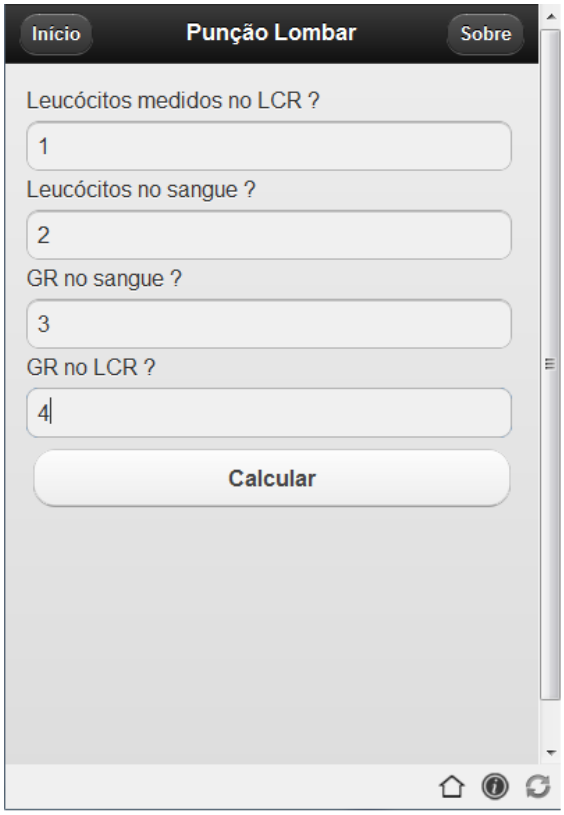

Anterior
Gap aniónico
Resultado:
-4.0 mEq/l
Nota:
Normal de 6 a 18 mEq/l

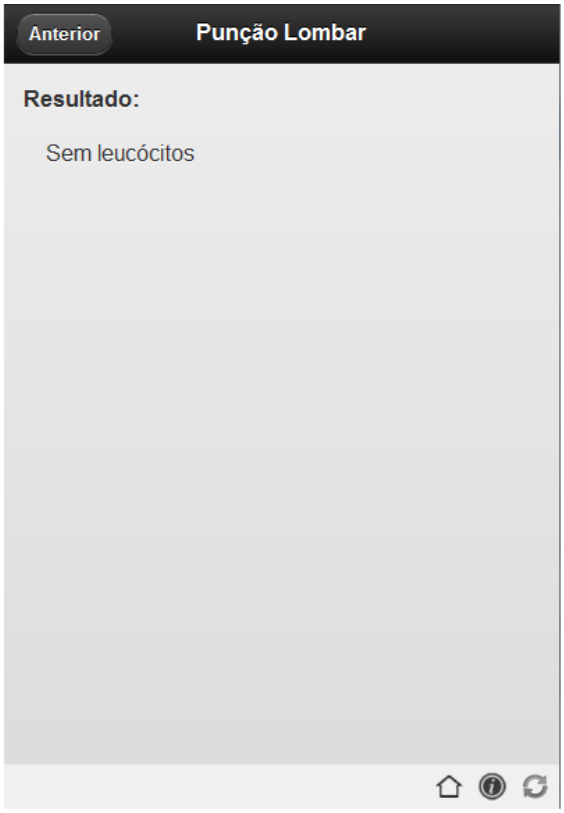

Taxa de filtração glomerular
Taxa de reabsorção de fosfatos
Índice de insuficiência renal
Gap aniónico
Osmolaridade plasmática
Unidades: mOsm/l
Plasma:
Na (mEq/l)
1
Ureia (mg/dl)
2
Glic (mg/dl)
3
Calcular

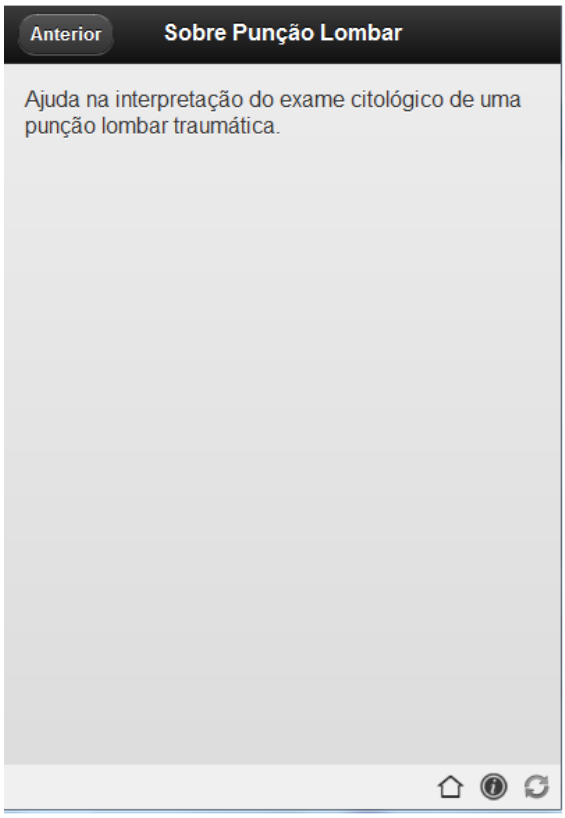

Anterior
Osmolaridade plasmática
Resultado:
2.5 mOsm/l
Nota:
N: 285-295 mOsm/l


Anterior
Sobre Função Renal
Fornece resultados de vários parâmetros da função renal.

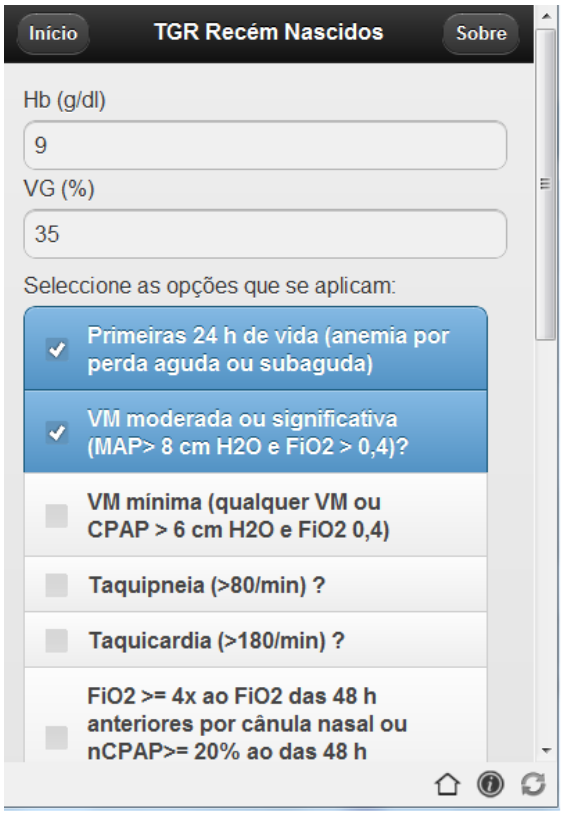
Requirement 9 - “Punção Lombar Traumática?”

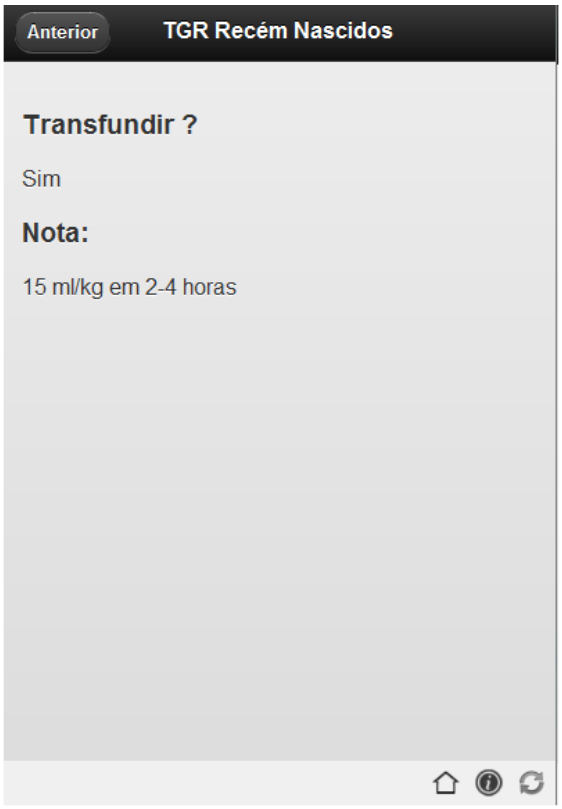


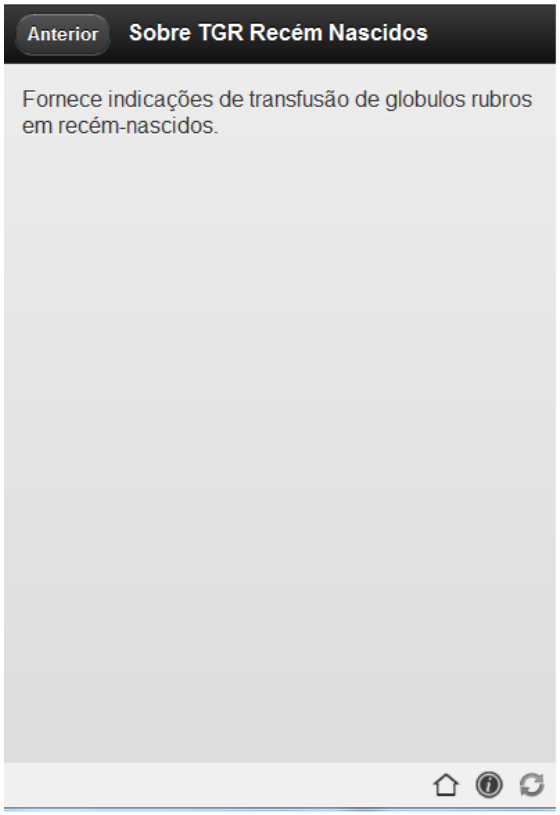




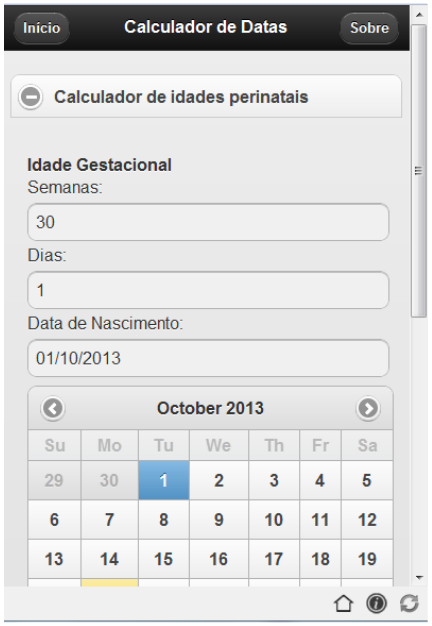
Requirement 10 - “Transfusão de Glóbulos Rubros em Recém-nascidos”

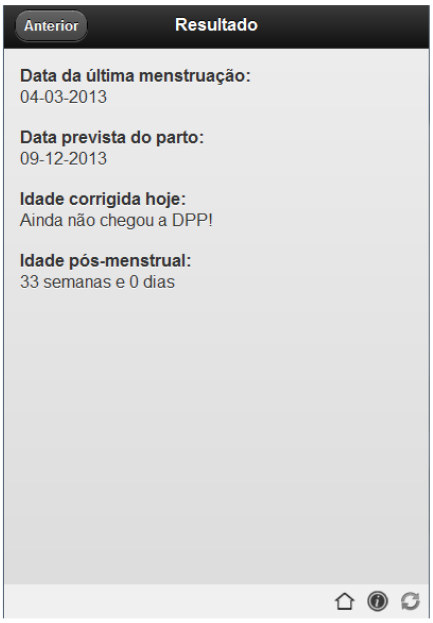


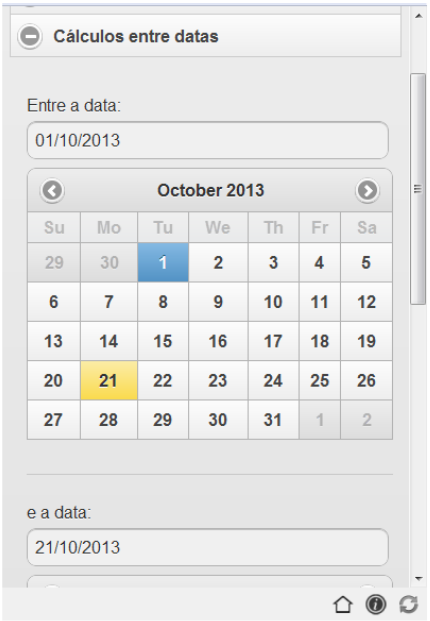


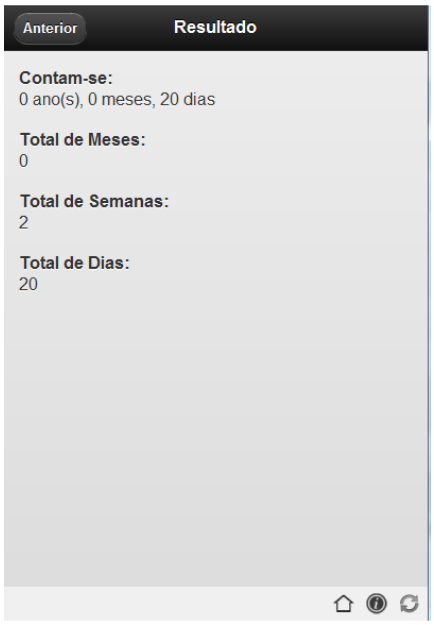


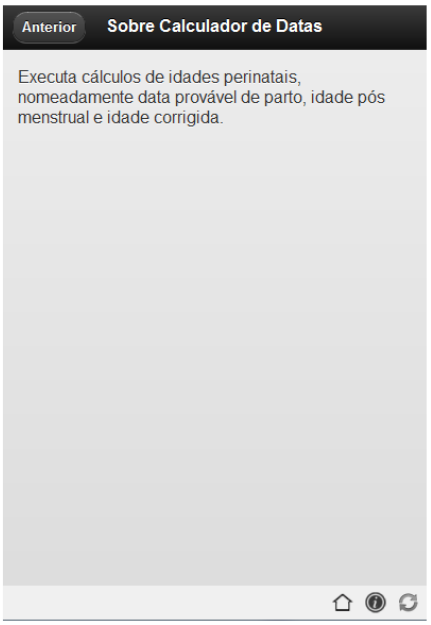
Requirement 8 - “Calculador de datas”



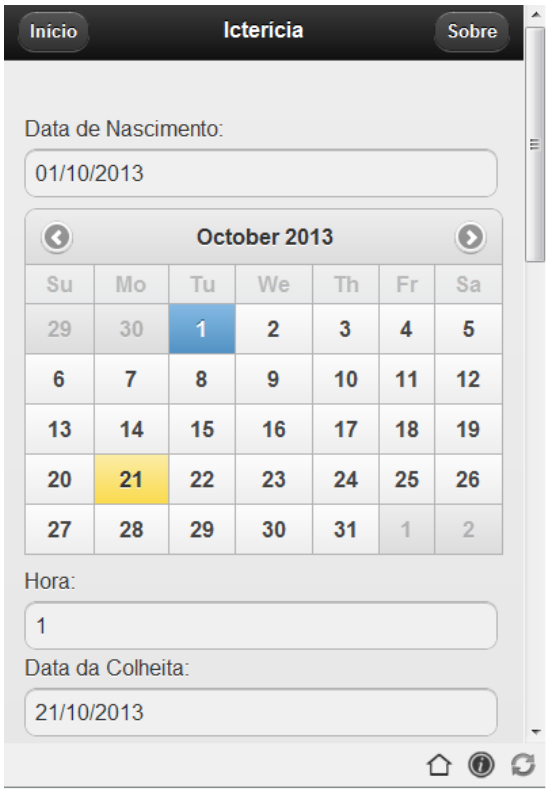


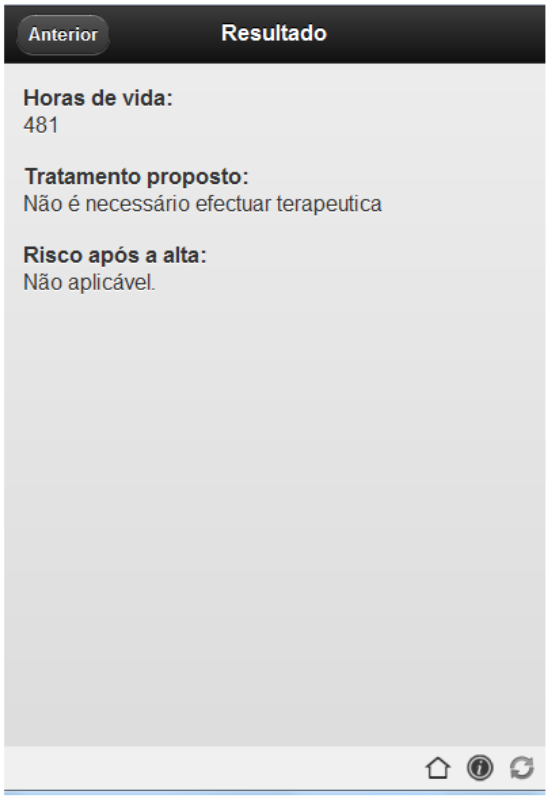


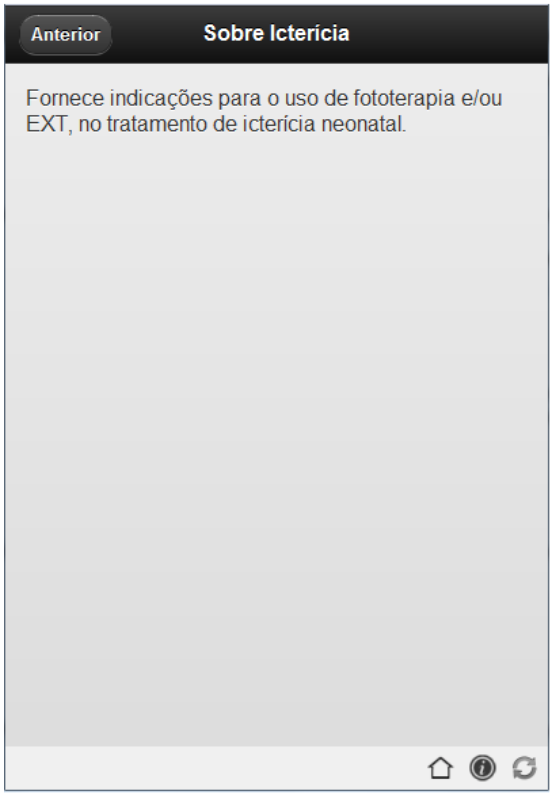




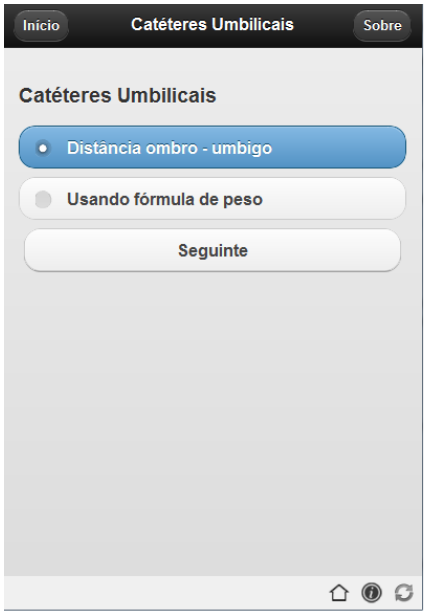
Requirement 5 - “Icterícia”

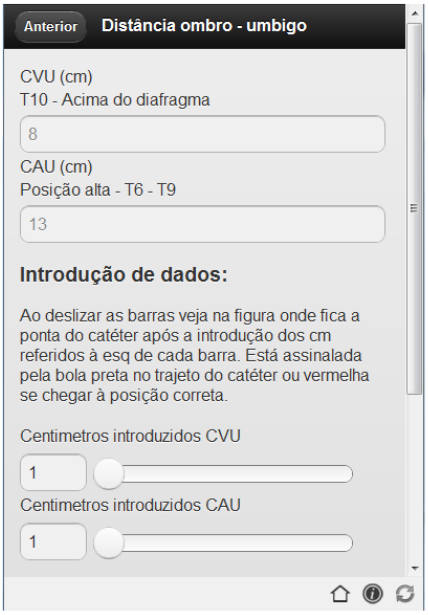


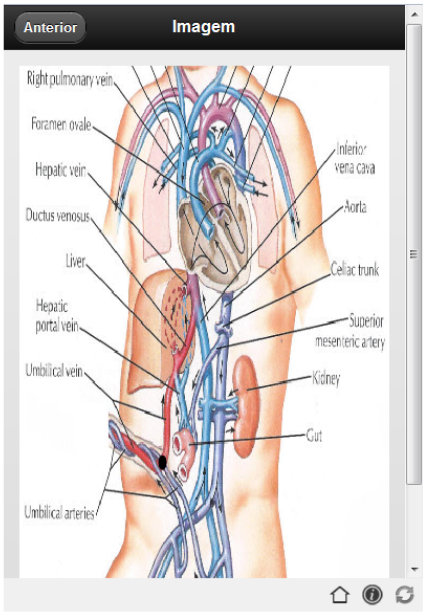


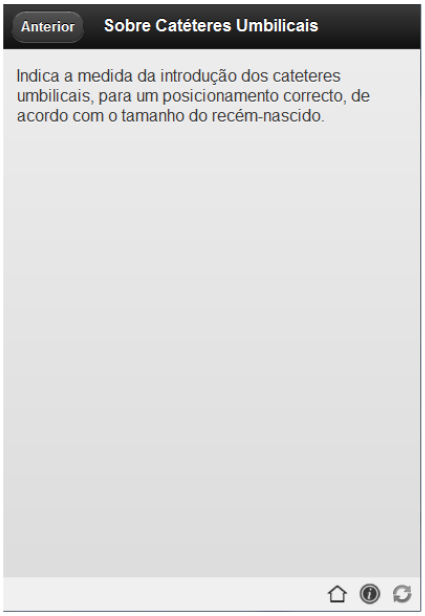


Requirement 4 - “Cateteres Umbilicais”

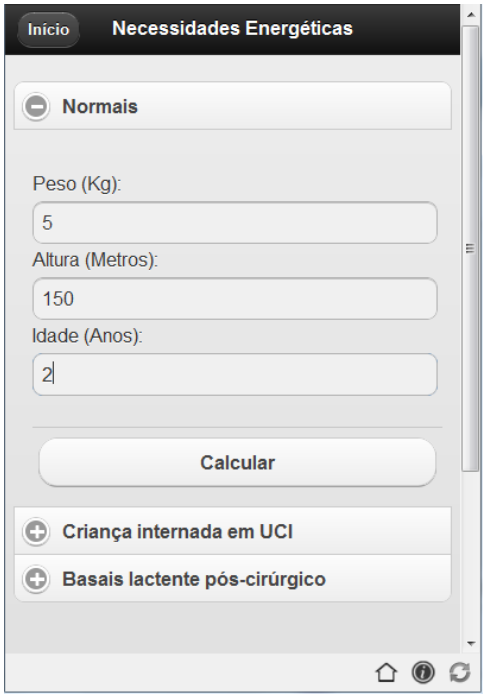


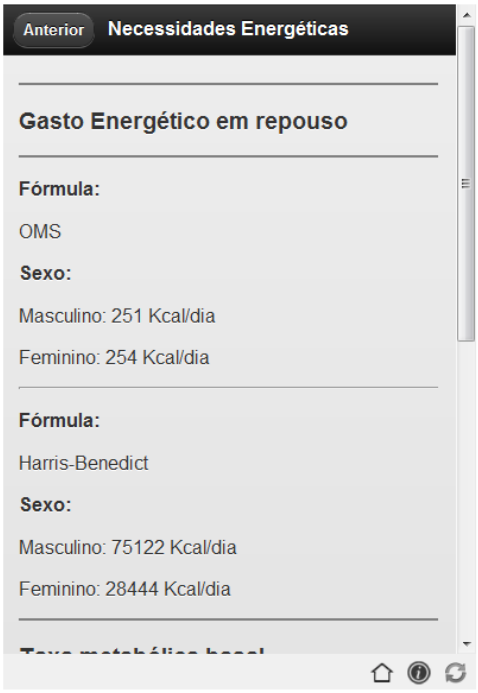


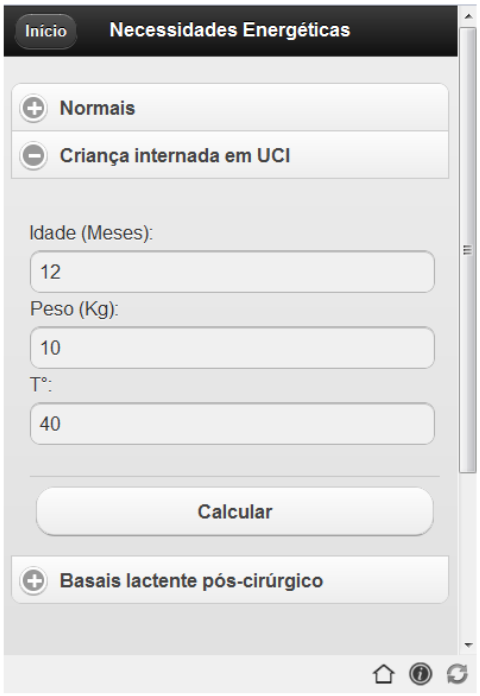


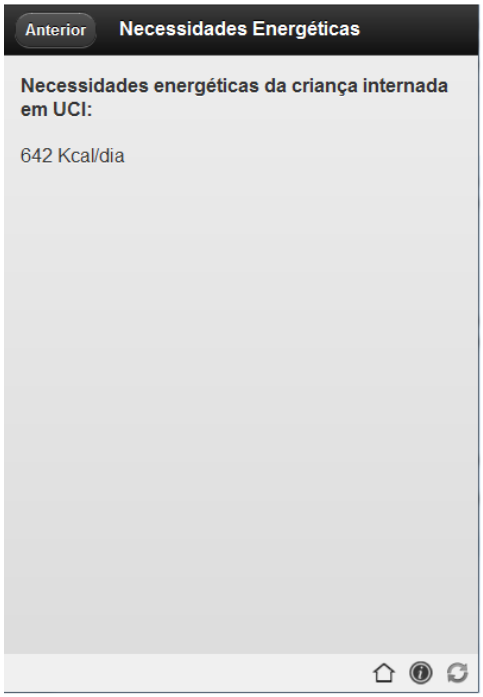


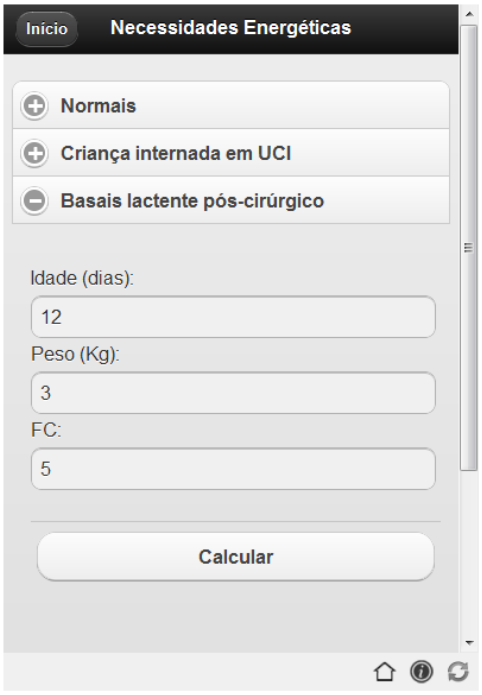
Requirement 7 - “Necessidades Energéticas”

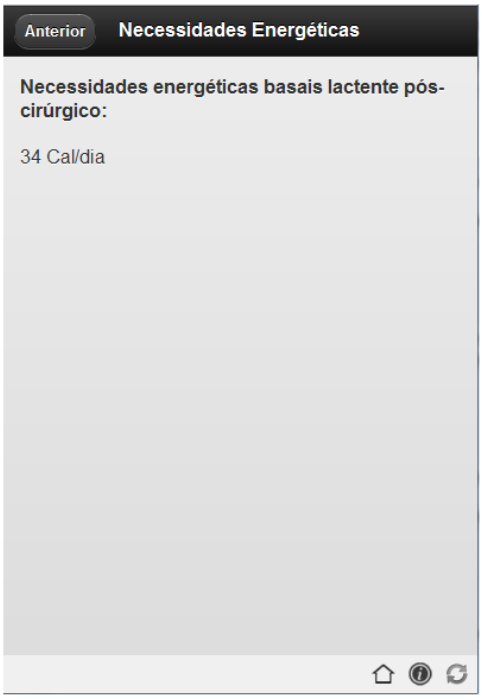




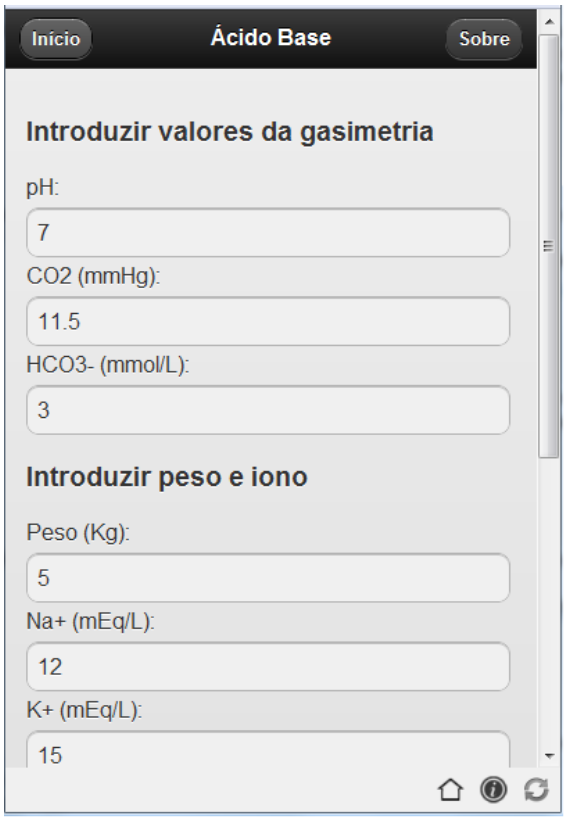


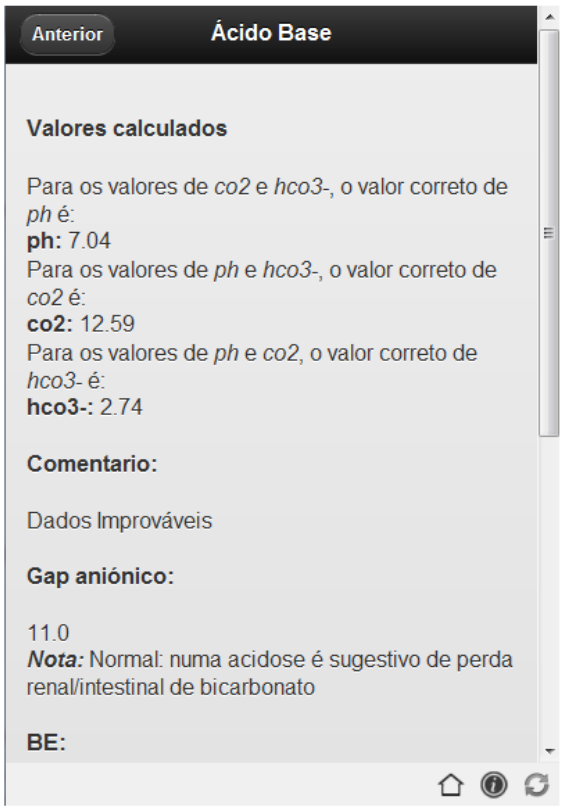


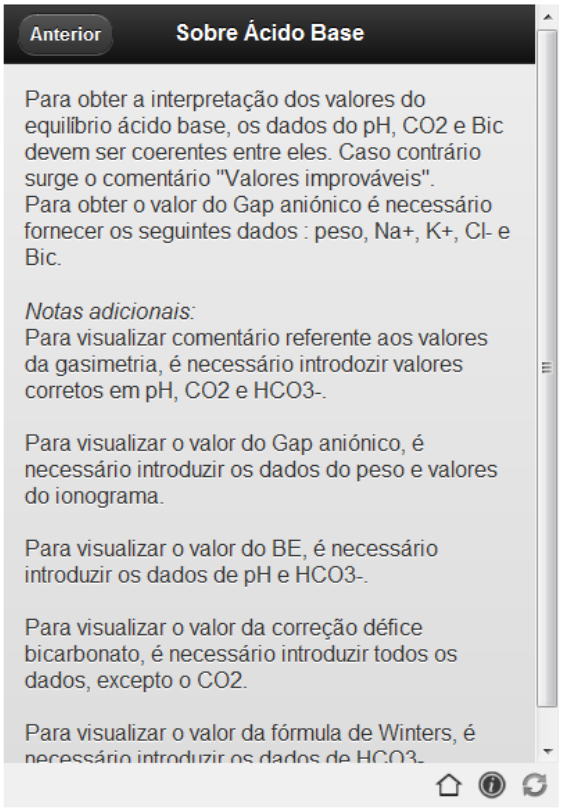




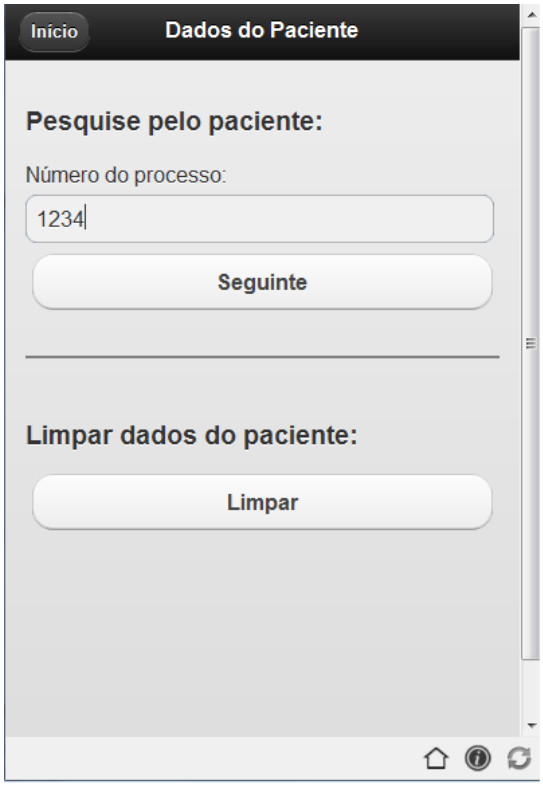
Requirement 6 - “Ácido Base”

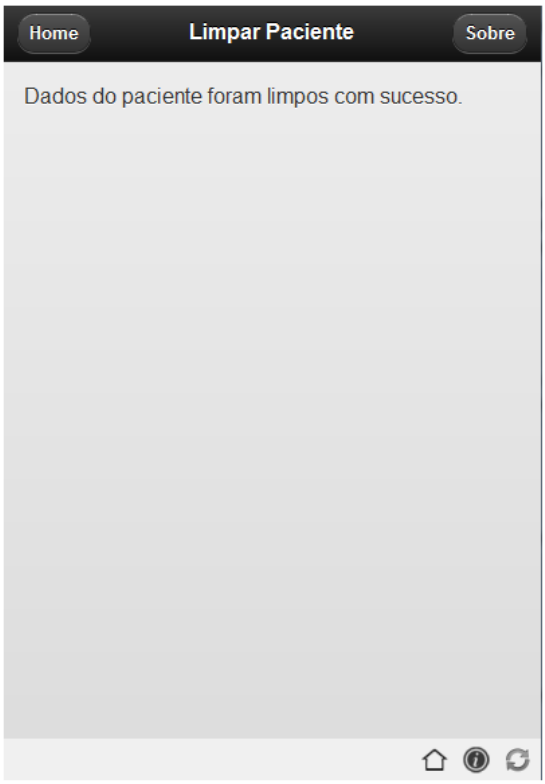


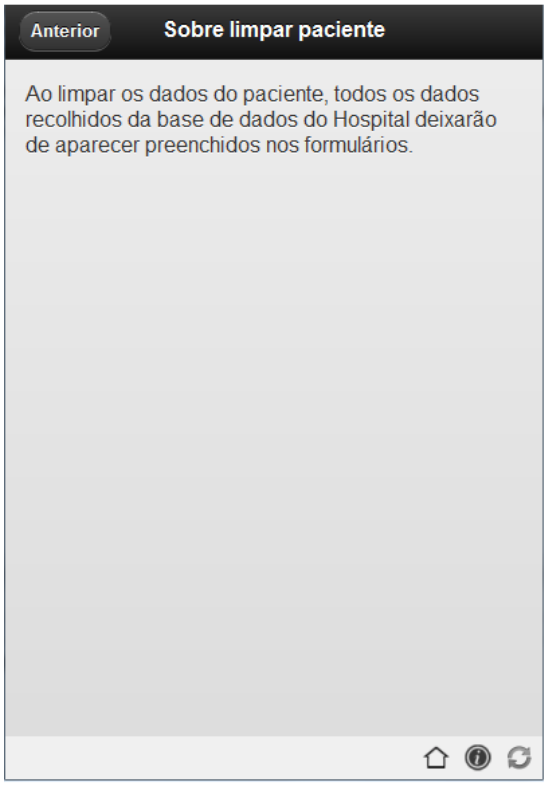




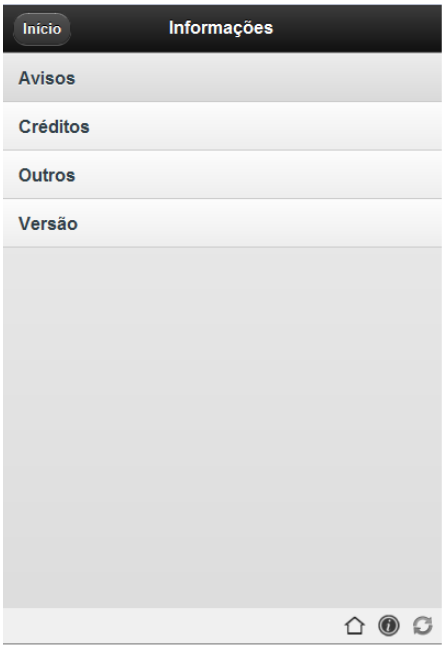
Requirement 16 - “Obter dados Paciente”

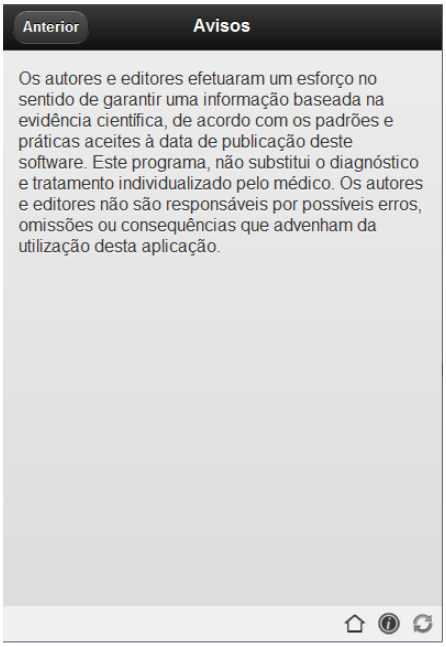


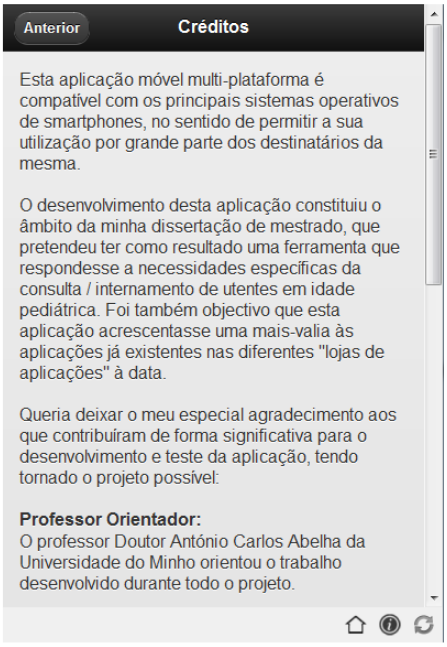


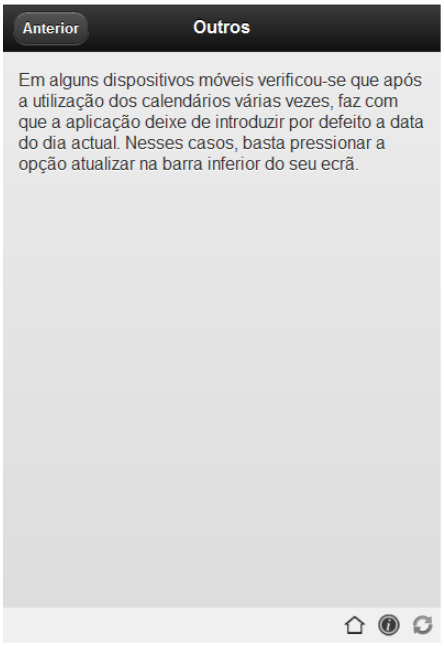


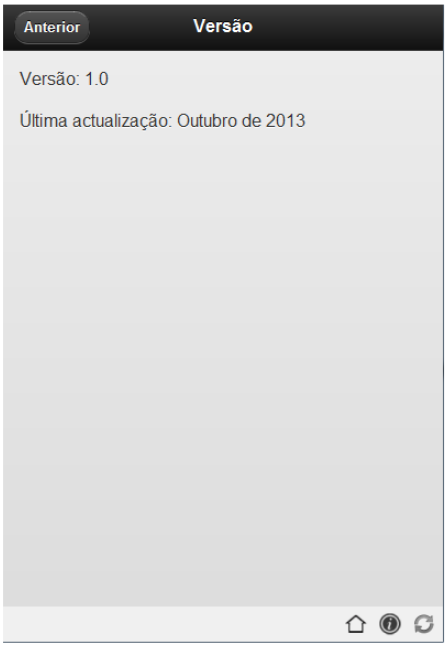
Aditional informations











Appendix IV – Application Questionnaire

Avaliação da aplicação móvel Pediatria (Pediatria mobile app Evaluation)

Este questionário tem por objectivo receber o feedback dos utilizadores finais da aplicação móvel Pediatria, para fins de avaliação e melhoria da aplicação. Por favor, leia com atenção as questões a seguir e em caso de dúvida solicite esclarecimento com o avaliador.

The objective of this questionnaire is to get the Pediatria application final users feedback, in order to evaluate and improve the application. Please, read carefully the following questions and in case of doubt contact the evaluator.

* Required

Em que grupo profissional se insere? *
Which professional group are you part of?

- ○ Medicina geral e familiar (Family medicine)
- ○ Pediatria (Pediatrics)
- ○ Estudante de medicina (Medicine Student)

Em que faixa etária se encontra? *
What is your age?

- ○ Menos de 25 (Less than 25)
- ○ 25-30
- ○ 31-40
- ○ Mais de 40 (More than 40)

Qual é a relevância das aplicações móveis na sua vida profissional? *
What is the relevance of mobile applications in your professional life?

- ○ Baixa (Low)
- ○ Média (Medium)
- ○ Alta (High)
- ○ Absolutamente essencial (Absolutely essential)

Quantas utilizações da aplicação considerou necessárias para se adaptar à aplicação? *
How many times did you use the app until you felt adapted to it?

- 1-3
- 4-6
- 7 ou mais (7 or more)

Classifique de 1 a 5 o aspecto da aplicação? *
Classify from 1 to 5 the aspect of the app?

	1	2	3	4	5	
Muito pouco agradável (Very little nice)	○	○	○	○	○	Muito agradável (Very nice)

Classifique de 1 a 5 a linguagem e os termos utilizados na aplicação? *
Classify from 1 to 5 the language and terms used in this app?

	1	2	3	4	5	
Muito pouco correcto (Very little correct)	○	○	○	○	○	Muito correcto (Very correct)

Classifique de 1 a 5 a organização da aplicação? *
Classify from 1 to 5 the app organization?

	1	2	3	4	5	
Muito mal organizado (Very poorly organized)	○	○	○	○	○	Muito bem organizado(Very well organized)

Classifique de 1 a 5 a rapidez da aplicação? *
Classify from 1 to 5 the app speed?

	1	2	3	4	5	
Muito lenta (Very Slow)	○	○	○	○	○	Muito rápida (Very fast)

Classifique de 1 a 5 a facilidade de interação com a aplicação? *
Classify from 1 to 5 the ease of interaction with the app?

	1	2	3	4	5	
Muito difícil (Very hard)	○	○	○	○	○	Muito fácil (Very easy)

Classifique de 1 a 5 o seu grau de satisfação com a aplicação? *
Classify from 1 to 5 your satisfaction level with this app?

	1	2	3	4	5	
Muito pouco satisfeito (Very little satisfied)	○	○	○	○	○	Muito satisfeito (Very satisfied)

Tem alguma sugestão de melhoria ou funcionalidade que gostasse de ver na aplicação?
Do you have any suggestion on improvements or functionalities you'd like to see in the application?

Submit

Never submit passwords through Google Forms.