Design and Prototypical Implementation of a Mobile Healthcare Application: +HealthExpress

BRAC UNIVERSITY

Inspiring Excellence

Design and Prototypical Implementation of a Mobile Healthcare Application: HealthExpress

Student Name	**Student ID**
Shahan Shawkat	14165002
Musa Nasrullah	14165003
Rahenul Islam	14365002

An Internship Report Submitted in fulfilment of the requirements for the award of the degree of Master in Computer Applications

Department of Computer Science and Engineering

BRAC University

December 2015

Abstract

Bangladesh has made significant progress to achieve millennium development goals on many frontiers. However the task is not finished yet and the country can overhaul the service quality of its healthcare sector taking assistance of various mobile based information technology tools. It is a common problem in many parts of Bangladesh that unregistered doctors taking advantage of people's ignorance provide medical consultation services. While such practice can inflict serious medical damages to the person being treated, hence it must stop. In order to help general public from mistakenly seeking medical consultation from an unregistered doctor we have tried to implement a prototypical mobile health care application that combines the possibilities to search nearest registered doctor, pharmacy, diagnostic centre, clinic and hospital. This application can also facilitate e-prescription transmission among doctors, patients and other relevant stakeholders. This paper describes the requirements for such a system, details its implementation. An in-depth literature review of the topics of mHealth and software engineering challenges to develop mobile application were also conducted to achieve the objectives of this paper.

Acknowledgements

Wow! Four months of hard work and we can't believe it's finally finished! All those persons who contributed in our work during the last few months we would like to thank them all.

First and foremost we would like to thank our supervisor Dr. Amitabha Chakrabarty. This internship report topic was proposed and realized following the visionary ideas of Dr. Amitabha Chakrabarty. We are very grateful for his valuable feedbacks and the constant motivation that he has extended to us ever since we started our post graduate education at BRAC University.

We would also like to thank all our friends and class mates who helped us a lot during our studies at the university.

Beyond friends there are families, we thank them very much for their love and support for us no matter what we decided to do and also for the financial support.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Bangladesh which is the ninth largest country [1] in terms of the size of its population in the world has made significant progress to achieve the millennium development goals [2] to improve the quality of life of its citizens. On its way to move forward with the development goals the country has made remarkable progress in the health care sector as well. However Bangladesh still has to achieve a lot, to offer its citizens the kind of healthcare benefits enjoyed by the countrymen of most developed high income nations. In this context, technology can be a key enabler to achieve Bangladesh its dream and the good news is so far the nation is utilizing various technologies such as mobile communication, internet etcetera, in the most positive way. Just to take an example Bangladesh has the twelfth largest mobile subscriber's base in the world [3]. Taking benefit of this large number of mobile subscribers, various mobile based solutions can be used to improve the service quality in the health care sector. mHealth is a key mobile based technology that will empower the future of the healthcare sector and it has a vital role to play especially in the case of a developing nation such as Bangladesh [4].

Often it comes to our attention from various national media that unregistered and unqualified doctors taking the advantage of people's ignorance provide medical services to general public [5, 6, 7]. Because of this undesirable practice some time people suffer damaging medical consequences to their health. From our research described in this paper we looked into possibilities to remove this malevolent practice from the society. Particularly we looked into opportunity to develop a mHealth app which can be used by general public to locate a registered doctor closed to his or her desired location with required medical services. We not only attempts to include the function to locate a registered doctor using our proposed mHealth app but also pharmacy, diagnostics centre, health clinic and also hospital so that the app can be used as a one stop service point for general public to locate desired medical service. Compared to how people find out desired medical entity, using the proposed health care mobile application can –

- increase the rate of successfully locating the nearest medical entity
- decrease the practice of unregistered doctor consultation service
- provide rating information to enhance the service level of individual medical entity

- facilitate the transmission of e-prescription using smart phones and store medical record of individuals in longer term.

Specifically within the context of mHealth the objectives of this paper is to-

- identify the major challenges to develop a mobile application from software engineering perspective.
- explore the prospect of mHealth app to solve healthcare challenges for a developing country like Bangladesh.
- develop system analysis and design of a mHealth application.
- create a prototypical implementation of the mobile application to demonstrate how such application can help increase healthcare service level.

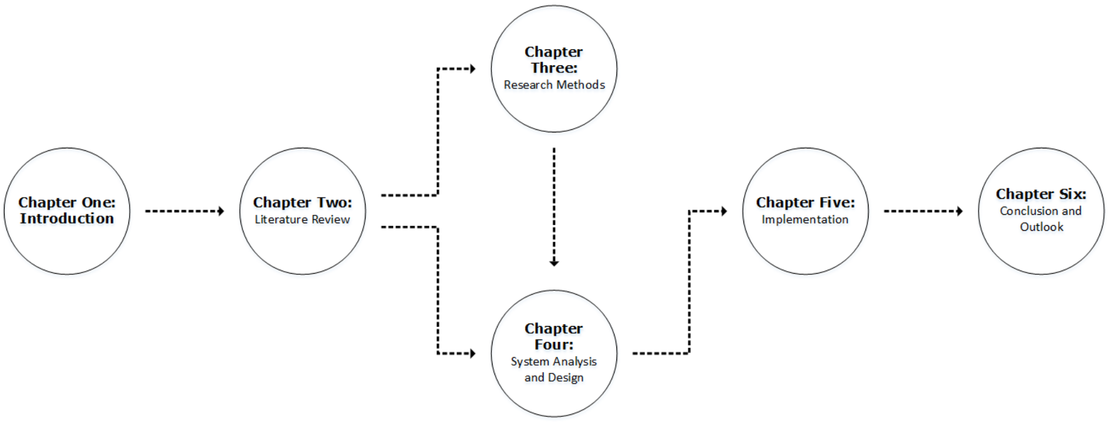


Figure 1.1 Report Outline

Figure 1.1 shows a quick overview of the report structure. To meet the first two objectives, a literature review was conducted, of which the results can be found in Chapter 2. The first part of the literature review focuses on overall concept of e-health and the factors that shape digital health scenario the second part focuses into software engineering challenges to develop a mobile application identified by current researches. Chapter 3 states the methodological approach to this paper. The functional and non-functional requirements as well as the core use cases of the proposed tool are introduced in Chapter 4. The implementation of the mobile application, including the data model, system design and user interface evolvement, is subject of Chapter 5. Chapter 6 concludes the report by discussing the findings and proposing future research endeavours in the area of mobile healthcare application.

Chapter 2: Literature Review

2.1 Digital Health

M-health, the use of mobile computing and communication technologies in health care and public health, is a rapidly expanding area of research and practice. M-health programs and interventions use mobile electronic devices (MEDs), such as personal digital assistants (PDAs) and mobile phones, for a range of functions from clinical decision support systems and data collection tools for health-care professionals [8, 9], to supporting health behaviour change and chronic disease management by patients in the community.

Current documented M-health interventions and programs include: mobile phone text messaging to support management of diabetes, hypertension, asthma, eating disorders and HIV treatment; mobile phone text messaging and PDAs as aids to smoking cessation, body weight loss, reducing alcohol consumption, sexually transmitted infection prevention and testing; PDAs for data collection in health care and health research and to support medical education and clinical practice. Whilst the majority of M-health interventions are reported from high income countries, there is an emerging literature on the application of mobile technologies in low income countries.

Mobile communication technology is the fastest growing sector of the communications industry in low income countries [10, 11]. In the last two decades, the global digital divide has narrowed most for mobile phone use, with many low income countries "leap-frogging" over fixed-line communications technologies, straight to expansion of wireless cellular communication networks. Whilst wireless communication network coverage and mobile phone ownership are not universal, or equally distributed, in low income countries, and the technology accessible to most of the population lags far behind that available in high income countries, there is still huge potential for M-health interventions and programs to have positive effects on health outcomes in resource-poor settings.

Mobile technologies have a number of key features that give them an advantage over other information and communication technologies in particular activities within health care and public health. Firstly, many MEDs have wireless cellular communication capability, providing the potential for continuous, interactive communication from any location e.g. telephone calls, text and multimedia messaging and also internet access via Wireless Application Protocol (WAP) or mobile broadband internet. Secondly, the devices are portable because of their small size, low weight and recharge ability, long-life battery power. Finally, many MEDs have sufficient computing power to support multimedia software applications. The combination of these features varies between specific devices and their relative importance will change with the health activity in which they are used. However, with advances in technology development, single devices increasingly possess many or all of these functions.

The main contribution of e-health is to provide an easy transmission and communication of information in health-care in forms of data or knowledge. During the 58th World Health Assembly held in Geneva in May 2005, the Ministers of Health of the 192 member states of the United Nations approved the so called e-health Resolution [12] that officially recognizes the added value of the information and communication technologies for health purposes.

E-health technologies opened the doorway to a new type of medical services where health-care professionals are able to utilize them fully for prevention and management of diseases, lifelong learning and communication with colleagues and patients. Moreover, education and use of e-health technologies can help to change a passive attitude of patients against their diseases towards a proactive attitude of informed citizens for managing their own health.

E-health concept has been the main topic of many books, papers in journals and presentations at conferences. New information and communication technologies (ICT) make possible to describe in a structured and unique way patient state, given procedures and the use of structured information for statistics and examination of quality of health-care services. For example, new tools make possible to transfer a structured electronic health record to the point of care, even in case that the transfer is provided at the point of care abroad. However, medical data can be extremely complicated due to the abundance of clinical terminology, as well as the structural complexity in the formation of the presented information. Thus, this information must be presented in a standardized format in order to ensure that the data is universally understood and organized.

2.2 E-Health concept

Nowadays health-care systems in economically developed countries are going through the process of a significant transmission. From the centrally controlled health-care, they are going to process controlled or shared health-care with the aim to reach personalized health-care. As soon as we place a patient or a citizen in the centre of a health-care system, the system will reflect its individual needs, expectations and wishes. However, such health-care systems should have three main features- that correspond to an electronic view, economics view and environmental view.

2.2.1 e-Health: Information and Communication Technologies view

Electronic health (e-health) is the first main feature of the modern health-care. We understand e-health according to a rather broad definition adopted by the European Commission. E-health is described as the application of information and communication technologies (ICT) across the entire range of functions that affect the health-care sector. E-health represents the interaction between patients and health/service providers, institution-to-institution transmission of data, or peer-to-peer communication between patients and/or health professionals. E-health tools play an important role in improving the health of citizens. If the e-health tools and services are used appropriately, they may provide better and more efficient health-care services for all people. Examples include health information networks, electronic health records, telemedicine services, wearable and portable systems which communicate, health portals, and many other ICT-based tools assisting disease prevention, diagnosis, treatment, health monitoring and lifestyle management. For this reason, the research in biomedical and health-care informatics is the prerequisite for the development of e-health applications.

E-health applications can be evaluated according to different levels of complexity, e.g. by the scheme proposed by J.H. van Bemmel [13]. The scheme can serve for structuring ICT applications in health-care into six levels. Each level contains different types of ICT applications with, from bottom to top, increasing complexity and a growing human involvement. According to this scheme we could assign each e-health application to one.

Level of complexity

Level 1 ”Communication and telematics” is the lowest level of complexity. On the lowest level of complexity, data enter the computers for information processing, are coded, transmitted, decoded and presented to users or other processing systems. E-health applications on level 1 are providing simple telematic applications. Therefore, e-health applications are transferring and share information in the form of data and knowledge on different levels of complexity. Level 2 “Storage and Retrieval” is extended with CT technologies offering storage and retrieval of data. Databases of patient data are used in hospitals, clinical departments and primary care practices, but databases in health-care can be also used to control the stock of goods necessary to run the institution. Nowadays, medical imaging systems deliver pictures in a digital form, and communication systems (PACS) are more used in health-care. Databases in health-care often store knowledge offered by different classification systems and nomenclature, e.g. Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT). The use of ICTs on level 3 “Processing and Automation” is much more dependent on people because the design of e-health applications at this level requires much more knowledge that is specific to health-care. Level 4 “Diagnosis and Decision Making” requires enough human knowledge and experience that should be incorporated in e-health applications. The e-health applications at this level are supporting diagnosis and decision/making in healthcare, and the methodology of the support should be fully justified. Level 5 “Treatment and Control” covers e-health applications, which involve humans, while the therapeutic and control processes are complex. E-health applications on level 5 need to use formalized medical knowledge, advanced information, and communication technologies. Level 6 “Research and development” reflects the creative human tasks in the development of new methods during research.

2.2.2 e-Health: The economics view

The common views on economics use the distinction between microeconomics and macroeconomics. Macroeconomics studies the behaviour of an economy as a whole and addresses issues of policies (fiscal, monetary), unemployment etc. Microeconomics studies the behaviour of individuals, especially merits of their decision making. The microeconomic view of health-care may help to identify and understand reasons of unhealthy behaviour. This view may also help to find an appropriate model of health-care funding. The neoclassical theory formerly viewed personal and household behaviour as a way of searching for delights and enjoyments. Later the view on household microeconomic behaviour changed. It is commonly viewed as a manifestation of preferences no matter what they are based on.

Both views assume finality and exogenous character of recipient's benefit. The finality of benefit means that the rational decision-making is aimed to achieve a benefit with no influence to subsequent economic processes. The exogenous character of a benefit means that the mode of operation of the economic system has no influence on the process of achieving a benefit. The real economic system consumption surely contains a considerable amount of consumption with substantive influence on future economic processes [14]. The consumption of education usually influences abilities of the subject and therefore influences its ability of future production and subsequent consumption. The consumption of health-care services and the subject's lifestyle influences the future health state of the subject and therefore influences its abilities to produce.

Building and maintenance of social contacts surely influences the relationships among subjects within the group defined by interpersonal relationships. The productive aspect of consumption may be represented in a way hinted by Milton Friedman [15]. No matter if households consume or save they are building their portfolios of actives and they usually behave the way to maximize their future income [14]. Health-care is one of major industries in the human society. The accessibility of health-care is not only a matter of available technologies but is also a subject of social and political processes. The main political issue of health-care is the interrelationship between co-equality and efficiency. Current theories believe that the relation between co-equality and efficiency is substitutive [16].

Health Economics studies resources on pharmacoeconomics, outcomes research, and managed care, value in medicine, health-related quality of life, performance assessment, and quality of care. Economic feature in a modern health-care offers solutions that can bring enormous savings. If properly deployed, it could contribute to the transformation of the health sector and change substantially business models of health-care facilities. Limits on resources – both in budgetary and staffing terms – weigh constantly on health-care providers. E-health tools and services enable more efficient organization of resources and care provision leading to greater productivity. Therefore, economic feature in health-care means an optimal allocation of restricted sources both human and financial.

2.2.3 e-Health: The environmental view

Environmental health in the narrow sense addresses all the physical, chemical, and biological factors external to a person, and all the related factors impacting behaviours. It encompasses the assessment and control of those environmental factors that can potentially affect health. It is targeted towards preventing disease and creating health-supportive environments.

Environmental health in the broad sense covers also all environmental factors developed in the society that belong to legal, social and cultural environment that can potentially affect health. Further we use the term environmental health in the broad sense. We assume that the concept of the environmental health in the broad sense will play a very important role in future. Similarly as epidemiology in its origin focused on communicable diseases only, the changing distribution of diseases in population changed the focus of the epidemiology to non-communicable diseases. The use of information and communication technologies in health-care is strongly dependent on the legal, social and cultural status of the society. Therefore, e-health applications have to consider differences in culture, language, geographic position, health-care system, legislative and social environment and other characteristics as age, sex and other features specifying the target population.

Smart, mobile devices are the fastest growing computing platform with an estimated 1.6 billion mobile device users by 2013 (compared to a current estimate of 2 billion PC users). This rapid proliferation of mobile devices over the last five years has dramatically altered the platform that is utilized for social, business, entertainment, gaming, productivity and marketing using software applications. Containing global positioning sensors, wireless connectivity, photo/video capabilities, built-in web browsers, voice recognition, among other sensors, mobile devices have enabled the development of mobile applications that can provide rich, highly-localized, context-aware content to users in handheld devices equipped with similar computational power as a standard PC. Yet, these same novel features/sensors found in mobile devices present new challenges and requirements to application developers that are not found traditional software applications. The combination of computing power, access to novel on board sensors and the ease in which applications can be monetized and transferred to the marketplace has made mobile application the new IT computing platform for development. However, the rapid proliferation of mobile devices and applications has outpaced the software engineering approaches tailored to mobile application software engineering.

Traditional software engineering approaches may not directly apply in a mobile device context. First, mobile device user interfaces (UI) provide a new paradigm for new human-computer interaction sequences (e.g., multi-touch interfaces, QR code scanning, image recognition, augmented reality, etc.) that have not been previously explored in research and of which no established UI guidelines exist. Second, the divergent mobile platforms (e.g., iOS, Android, Windows 10, etc.), differing hardware makers for platforms (e.g., Android versions found on HTC, Google, Samsung) and mobile phone and tablet platforms (e.g., Apple's iPhone and iPad) have necessitated developers to make a series of the same application tailored for each type of device. Third, the novelty of a truly mobile computing platform provides both unique opportunities and challenges. For example, Roman, Picco and Murphy assert that "mobility represents a total meltdown of all the stability assumptions" made in software engineering. In this paper, we discuss how these three factors present four significant challenges to mobile application software engineering that are critical to enable the design and development of quality mobile application utilizing the capabilities provided by mobile device hardware and platforms.

2.3 Mobile Application Software Engineering

Based on the three factors novel to mobile application development outlined in previous section, we outline the following fundamental, unique challenges to the state-of practice in mobile application software engineering:

2.3.1 Creating Universal User Interfaces

There has been some preliminary research in creating a universal user interface for mobile devices. Each mobile platform has a unique guide to address developer user interface requirements. The user interface guidelines have several overlapping themes.

A significant consideration for mobile UI development relates to screen size and resolution. For example, Apple devices are limited to two sizes based on the size of the iPhone and the iPad whereas Windows 10, Android, and Blackberry provide screens of varying sizes and screen resolutions. As a result, UI design is difficult and mobile application developers must anticipate the targeted device(s).

Shneiderman's "8 Golden Rules of Interface Design" have been well received since their introduction [17]. However, these rules may not equally apply to mobile devices. Research by Gong and Tarasewich suggest that four of Shneiderman's guidelines readily translate to mobile devices, including: enabling frequent users to use shortcuts, offering informative feedback, designing dialogs to yield closure, and supporting internal locus of control. The remaining rules must be modified to be made applicable to mobile development.

As these challenges continue to evolve, further research should focus on streamlining application development efforts regardless of the mobile platform or device. Significant effort should be directed towards anticipating the diverse landscape of user capabilities, user interfaces and user input techniques.

2.3.2 Enabling Software Reuse across Mobile Platforms

Mobile applications currently span several different operating system platforms (e.g., iOS, Android, Windows 10, etc.), different hardware makers (Apple, HTC, Samsung, Google, etc.), delivery methods (i.e., native application, mobile web application) and computing platforms (i.e., smartphone, tablet). Each of these options must be considered during mobile application development as they have a direct influence on the software requirements. Companies currently need to make a business decision to target a single mobile device platform with rich features, multiple platforms through a mobile website with less rich features or spend the resources necessary to broadly target the gamut of mobile devices with rich, native applications. If targeting a single platform, developers may decide to build a single application for all platforms at the risk of some functional inconsistencies or instead consider building multiple version targeting each hardware/computing platform.

Within this development environment, many companies have separate development teams or separately contracted out the development efforts for different platforms (e.g., iOS and Android) essentially redoubling the software engineering effort needed for functionally similar mobile applications. Even when development is coordinated amongst development teams targeting different platforms, it is often in an ad hoc basis without a concerted effort to reduce the development time and cost through existing, reuse conscious software engineering methodologies.

Recent efforts in adapting HTML5 with tools like PhoneGap aim to reduce the development effort to produce nearly native applications across multiple platforms by rendering native applications interfaces through webviews. However, this approach does not allow for rich features that have access to the mobile device's API and is a technological solution rather the desired software engineering approach to reuse early software engineering assets.

2.3.3 Designing Context-Aware Mobile Applications

Mobile devices represent a dramatic departure from traditional computing platforms as they no longer represent a "static notion of context, where changes are absent, small or predictable". Rather, mobile devices are highly personalized and must continuously monitor its environment, thereby making mobile applications inherently context aware (collectively time-aware, location-aware, device-aware, etc.). Mobile applications are now contextualizing proximity, location, weather, time, etc. to deliver hyper specialized, dynamic, rich content to users through context-aware applications. Previously, web applications would often provide contextualized content based on time, detected location and language. However, the extent of context-awareness currently possible in mobile applications is beyond what software engineering approaches have encountered outside of agent-oriented software engineering. The consideration of context-awareness as a first-class feature in mobile application software engineering is needed so that the requisite attention is paid by developers when analysing these requirements resulting in better designed context-aware applications.

2.3.4 Balancing Agility and Uncertainty in Requirements

While most mobile application developers utilize an agile approach or a nearly ad hoc approach, the growing demand for context-aware applications, competition amongst mobile applications and low tolerance by users for unstable and/or unresponsive mobile applications (even if free) necessitates a more semi-formal approach. This should be integrated into agile engineering to specify and analyse mobile application requirements. The dynamic, contextual nature of mobile application content (e.g., location based applications) allows for situations in which the application's behaviour may not be able to fully satisfy the specified functional and non-functional requirements thereby necessitating that the application be self-adaptive. In this scenario the software will then provide less rich content satisfying less stringent requirements. For some mobile applications, this may arise if, as determined in the requirements, it is better for the application to run continuously and, when necessary, to autonomously modify its behaviour and provide reduced functionality rather than provide no functionality at all. For example, in a location-based application several factors (e.g., low battery, GPS sensor disabled, etc.) may affect the granularity and recentness of its content. In some location-based applications, it may be better to provide old content (i.e., content based on a previous location) rather than displaying an error message or risk slow or no response from the application.

Within mobile application software engineering, the need for an application to self-adapt, depending on context, has been constructed using ad hoc approaches. Yet, as mobile applications become more context aware, self-adaptive requirements will need to be more formally integrated into agile development so that developers more rigorously consider the behaviour of an application when its full requirements cannot be satisfied dynamically and how it can self-adapt to partially satisfy the requirements.

This paper briefly described four current challenges that we see for mobile application software engineering: designing universal UIs, developing for mobile application product lines, supporting context-aware applications and balancing agility with specifying requirements uncertainty. This paper asserts that mobile application software engineering research efforts need to focus on development approaches emphasizing UI design, proactive reuse at early software engineering phases, and attention to context-awareness and sensitivity to specifying requirements to handle requirements uncertainty within the existing agile development approaches used for development applications. In addition, software engineering research needs to emphasize education initiatives in these four areas to ensure that these approaches are disseminated to those doing actual mobile application development.

Chapter 3: Research Methods

3.1 Methods

The basic timeline of the process of writing this paper is depicted in below table. The official start date was September 6th and the date of the submission was December 17th. Originally, a waterfall-like step-by-step approach was laid out where one part of the paper would be completed before starting another part. However, mainly due to a vast amount of interesting literature, many of these time periods had to be extended and moved.

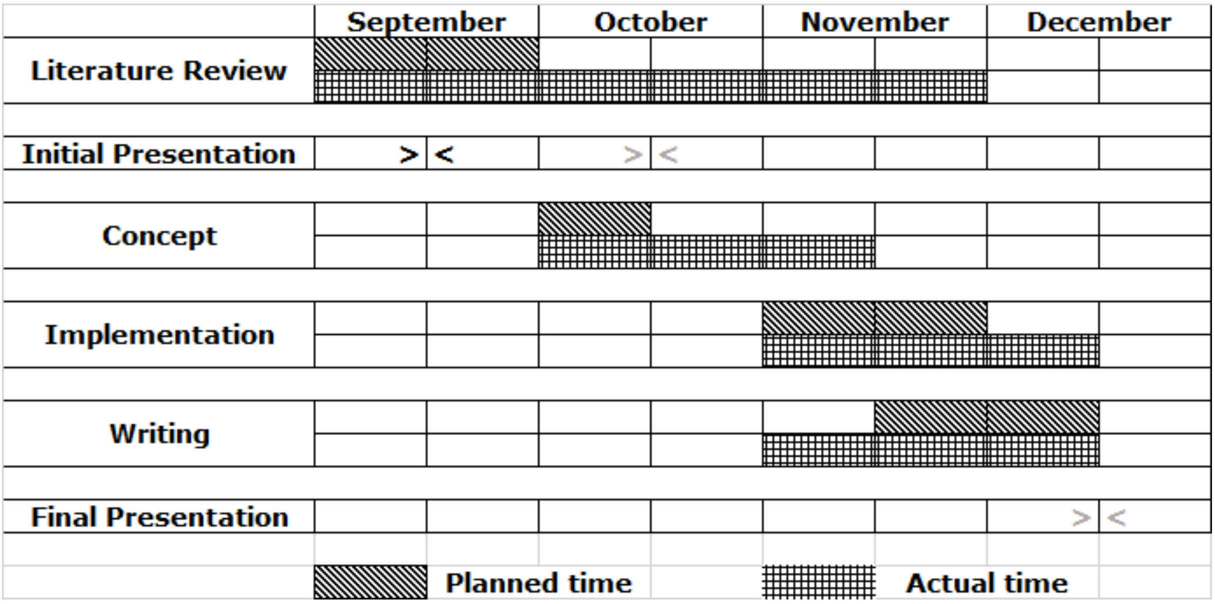


Table 3.1 Gantt chart of the research process

In the above diagram an overview of the research process is shown in a Gantt chart. The following sub-sections further detail how the particular areas of this internship project were executed.

3.2 Literature Review

At the beginning of the research project, a literature search was conducted. Based on possible search terms and key word combinations, Google Scholar and Emerald Insight were used for in-depth searches on the topic of mobile software engineering and mHealth from the beginning of September 2015 to end of October 2015. In the below diagram the process of filtering out suitable research publications from a list of search results was shown. First, the title of each search result was examined in order to determine whether the paper could have a possible connection to this particular internship project and the research questions. Publications fitting into the research topic were added to the search log. Then, the abstracts of all log entries were further analysed and not suitable papers were marked as such and excluded from any further analysis. The next step for the residual log entries was to locate the articles using the on- and offline access possibilities. However, quite a number of potentially interesting publications could not be accessed and were therefore eliminated. After skimming the content of all remaining log entries, a last few were excluded and the rest was used to construct the literature review found in section 2.

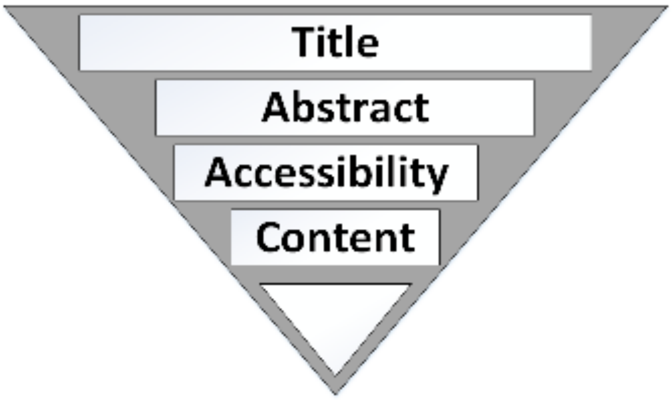


Figure 3.1 The process of filtering literature search results

3.2.1 Requirements Elicitation

The requirements that will be presented in section 4 were based upon the functionality of already existing medical healthcare applications and approaches. The Ionic Framework and various health care web applications were analysed with respect to their functionalities, capabilities and limitations. These tools all implement four main use cases: creating medical entities, searching for medical entities, giving feedback via ratings and user authentication.

3.2.2 Concept and Implementation

Based on the requirements elicited in Chapter 4, the basic system design, data structure and user interface were created (see Chapter 5 for more details). A prioritized backlog of all needed functionalities was generated (Table below) and its elements were continuously implemented until all were fulfilled.

Title	Size	Priority	Title	Size	Priority
Create Medical Entity	XL	1	Edit Medical Entity	S	1
- Medical Entity Title	S	1	Search Medical Entity	M	1
- Medical Entity Description	S	1	User Authentication	M	1
- Medical Entity Image	M	2	Rate Medical Entities	M	2
- Medical Entity Tags	S	4	Favourite Medical Entities	M	2

Table 3.2 Basic backlog (size S → XL, priority 1→ 5)

During the time of conception and implementation of the prototype, weekly meetings with the supervisor acted as small review meetings of the current development status. Drawbacks of user interface elements, ideas for the database to be used as well as re-prioritizations of certain backlog items were discussed during the meetings.

Chapter 4: System Analysis and Design

4.1 Requirements

This section introduces the minimal requirements for the prototypical implementation of a mHealth app healthExpress.

4.1.1 Roles

The system should be developed to support minimum three types of roles. On one side we have general user of the application who searches for doctors, pharmacies, diagnostics centres, clinics and hospitals. On the other side we have doctors who interact with the application to send e-prescriptions to patients or view patient's history. The another role is for administration purpose whose primary purpose is to add, update and remove medical entities from database and provide support to application users using comments.

4.1.2 Use Cases and Functional Requirements

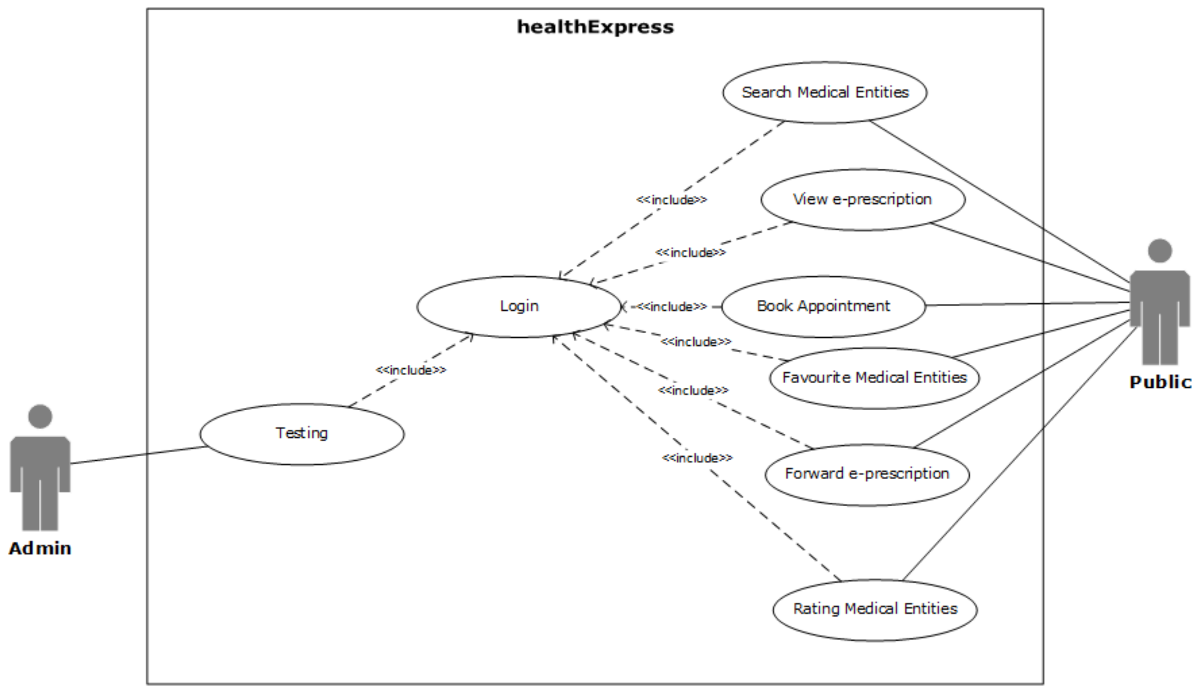


Figure 4.1 Mobile Application Use Cases

There are six main use cases that are mostly based on information seeking model from the perspective of a general user as shown in Figure 4. These use cases all provide a way to add, update or delete medical entities, to search for medical entities, to provide feedback for medical entities, to transmit e-prescriptions. Lastly a registration and log in system is necessary when storing user-specific data to differentiate among various roles described in the use case.

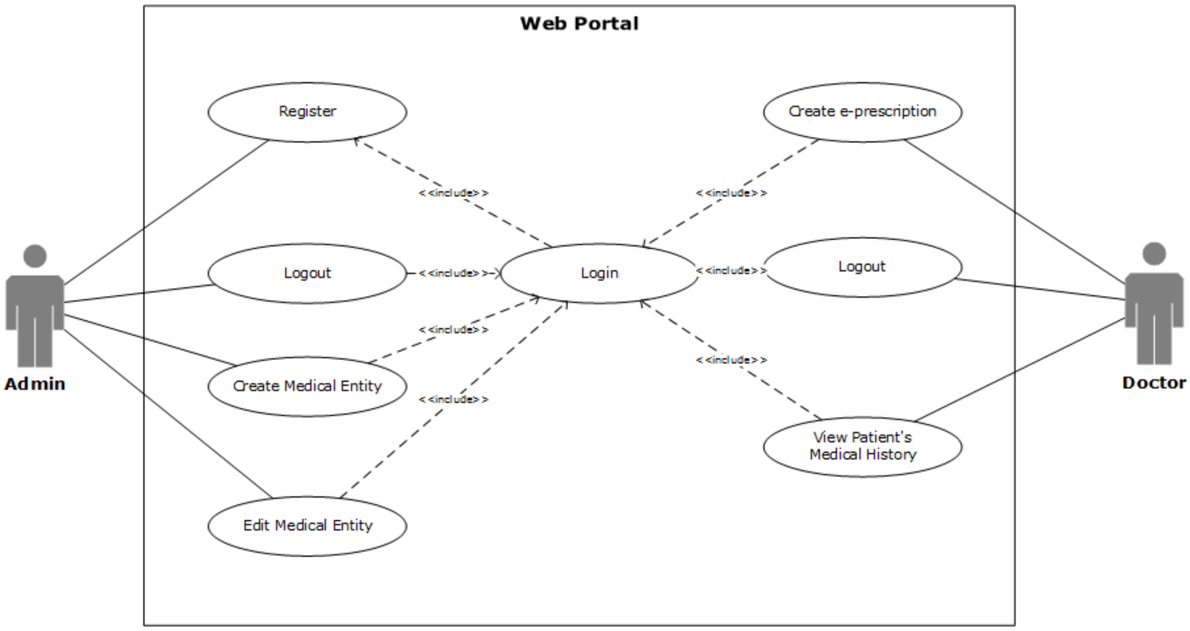


Figure 4.2 Web Portal Use Cases

In above diagram use cases relevant for the Web Portal is shown there are two different actors for the web portal and they are the admin user and the doctors. A admin user has four major use cases whereas the doctor primarily has three major use case.

4.1.3 Registration and Login

For any activities, the user is required to have an account and be logged in. In order to register for an account, the user must enter a valid email address, a preferred username and a password into corresponding fields or use Google accounts to sign in. After having created an account or entered a correct email-password pair, the user will be logged into the application and stay logged in.

4.1.4 Medical Entity Discovery

Users should be able to launch the application, login and immediately start looking for suitable medical services. There should be a search function that allows the user to type in search keywords, searches the database of medical entities and displays results according to fit, distance of medical facilities from users and ratings.

After having found or discovered a medical facility, the user should be able to view all detail information (see section 4.2.2) in order to determine whether the facility fits his/her personal needs.

4.1.5 Medical Entity Creation and Editing

Admin users must have the possibility to create a new medical entity. Each medical entity consists of a title describing the name of the entity, a more detailed description of the entity which includes address, contact details, opening hours, one image of the result and a number of tags for better discovery. Most importantly any medical entity should have the option to mark the entity on map so that user can locate the service point easily.

In addition to the ability to add a medical entity, an admin user must also be able to edit a medical entity details to reflect any changes, also the admin user should have the functionality to remove a medical entity from firebase if necessary. Any newly created medical entity should be uploaded to the back end server immediately to ensure availability of data.

4.1.6 Booking Appointment

Using the mobile application a general user should be able to book appointment with desired medical entity. Once a user complete searching a medical entity he or she can see the detailed view of that particular entity and should be able to book appointment with that exact service. The appointment booking system is a separate system however healthExpress mobile application should have the capability to transmit and receive data to the booking appointment application from the mobile interface.

4.1.7 User Feedback

The system should also offer the user possibilities to give feedback on medical entities. A 5 star rating system should be used that depicts the rating of a specific medical entity as generated by all users or, if available, by the current user in order to ensure public can identify a medical entity that excel in the service it provides.

In order to show appreciation or to mark medical entity for later reviewing purpose, users should be able to save a medical entity to their list of favourites. This list of favourites needs to be available to the user at any point in time when he/she is connected to the Internet.

4.1.8 E-Prescription Transmission

A doctor should be able to create an e-prescription using the web portal connected with the application. Once an e-prescription is created by a doctor it should be transmitted to specific user and the user should be able to view the prescription as a pdf documents on his or her mobile phone. Moreover the user should be able to send the e-prescription via e-mail to desired pharmacy or another person who is listed in the application database.

4.1.9 Non-Functional Requirements

Apart from the functional requirements mentioned above, the system in general must adhere to certain behavioural requirements:

1. Performance: The system shall display an activity indicator when elements take time to be loaded.
2. Availability: The system must be available at all times, except when upgrading or restarting the backend database. In case of no network connection the system shall enable the admin to add, update or delete medical entities to the server once the network connection becomes available again.
3. Usability: Error messages shall only be presented when the user can (try to) perform an action that could potentially fix the erroneous situation. Therefore, error messages shall always include a proposed solution to the problem.

4. Security: The system shall insure that data is protected from unauthorized access. All authorization communication between client and server needs to be encrypted (username and password data).
5. Localization: The app must be available in Bangla and English, depending on the smartphone's language preference settings.
6. Implementation: The app shall be built using Ionic framework with Firebase as Back-end and run on all iPhone and Android devices released after 2013.
7. Capacity and Scalability: The system shall not store any medical entity information on the local client device.

In recent time there has been an emergence of delivering various health care facilities by using mobile information technology platforms. This phenomenon has been tagged as mHealth or mobile health. Although there is no standard or accepted definitions of the items that constitutes the term mHealth until now, but it has been seen as a major factor shaping the healthcare sector of the future. Generally the goals of mHealth are considered to be streamlining the health-care delivery system, ensuring faster and cheaper access to at least some of the health care elements. American Health Information Management Association (AHIMA) defines mHealth (mobile health) as:

"the use of devices such as smartphones or tablets in the practice of medicine, and the downloading of health-related applications or 'apps' … [to] help with the flow of information over a mobile network and … improve communication," specifically between individuals and clinicians. (Source: AHIMA Guide 2013)

In Bangladesh the health care sector has not been able to fully utilize the benefit of advance information technological instrument such as mHealth due to lack of proper infrastructure. In our project, we aim to design a mobile health care system which runs on Android platform to show how application of advance information technology can solve issues within health care sector of Bangladesh. This application allows users to search a doctor, hospital, clinic, diagnostic centre and pharmacy within the boundary of Dhaka metropolitan city. Once a user search and find the desired doctor or facility he or she can select the component and view details of a doctor such as chamber location, contact details etc. Also user will be able to view details of medical facilities such as hospitals, clinic, diagnostic centre and pharmacy. A major goal of the project is to include a high volume of information related to doctors, hospitals and other medical facilities which has not been done previously. The mobile app will be a one stop place for Dhaka city dwellers to find the desired doctor or medical facilities within the limit of Dhaka city.

4.2 Software Development Strategy

The project's software development strategy is based on Agile Software Development disciplines. Against the traditional software development methodologies, the leaders of the software development approaches having similar opinions as Extreme Programming (XP), Scrums, Crystal Methodologies, Feature-Driven Programming formed an alliance and put out the Agile Manifesto in 2001 which is;

"

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*

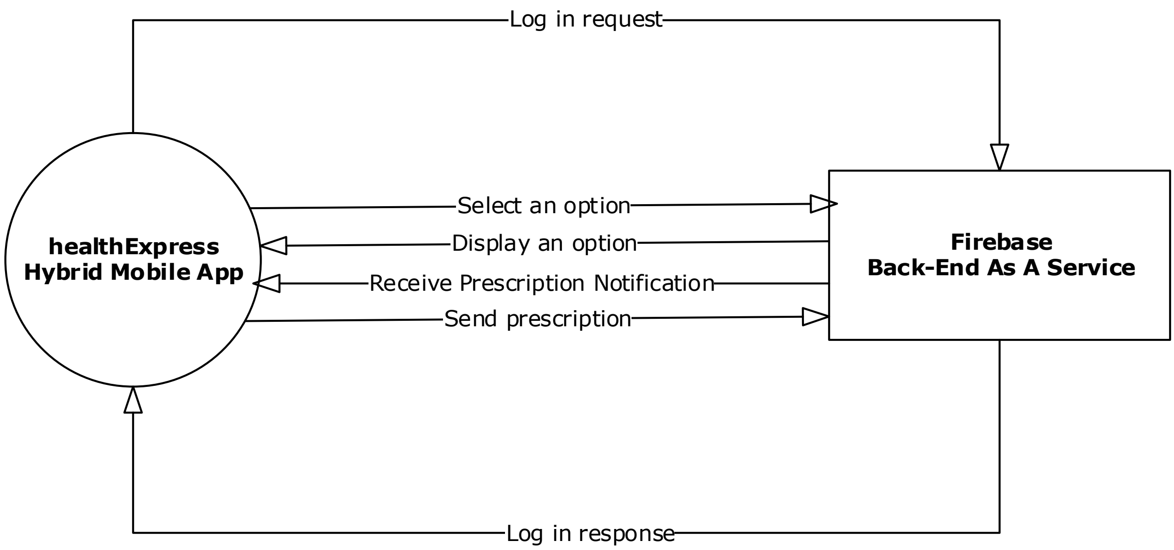
That is, while there is value in the items on the right, we value the items on the left more."

The main goal is bringing speed and effectiveness in modelling and documentation of developing software systems. Teamwork, collaboration, rapid and continuous delivery of high-quality software are the main principles laying behind agility. Agile approach does not oppose the design activities but it is against to design a software at the beginning of a software project. At the beginning of the project, system is pre-designed. The design is flourished with the evolvement of requirements, code and test results. Design is mentioned as output of the process, rather than being input.

According to the agile approach, tasks have to be broken into small increments with minimal planning. Long-term planning is opposed. All development activities have to be done iteratively. Each iteration of 3-4 weeks is composed of a full software development cycle including planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of an iteration, a working software must be delivered to the customer. Stakeholder involvement takes an important place in agile methods. As stated in the principles of the Agile Manifesto, "Business people and developers must work together daily throughout the project." This method is believed to improve the requirements analysis and finding the defects earlier, which increases software quality. To gain agility to the process, tool usage is encouraged. Tools for unit and functional testing and techniques such as continuous integration, pair programming, test driven development, design patterns, domain-driven design, code refactoring are often used to improve quality.

4.2.1 Planning Phase

In this paper, only the first iteration is ensample as a cross cut of the overall system. The project plan is prepared with respect to agile project planning approach and the first iteration starts with the requirement analysis phases. The overall context of the system is analysed and represented as the Data Flow Diagram Level 0, in the following figure-



System Context Diagram

Figure 4.3 HealthExpress Context Diagram

After gaining an understanding of the major functionalities of the system, the requirements are stated as user stories and they are prioritized. Then the prototyping phase initiates. The user interfaces is developed and started to sketch the system. This prototype enabled us to analyse the requirements in a deeper manner. The design of the system is shaped with the development of requirements and the prototype.

4.2.2 Requirement Analysis

After the requirements are gathered, they are prioritized. The first prototype of the system is developed in the first iteration, due to the user stories. The functional requirements are;

1. User can login to hybrid mobile app via his/her mobile phone.
2. User can search a doctor and other medical facilities.
3. User can select an item and view details.
4. User can set an appointment with a doctor.
5. User can view prescription issued by Doctors.

The non-functional requirements are;

1. Performance:

An Android application should be fast and efficient. There is a tendency in the computing world these days to assume that Moore's Law will solve all our problems related with performance. When it comes to embedded applications, Moore's Law is a bit more complicated. Then, embedded devices like phones are increasing in actual, raw speed much more slowly than desktop systems. It’s important to write efficient code and by doing this phones will see the same speed increases as desktops and servers. Writing fast code means keeping memory allocations to a minimum, writing tight code, and avoiding certain language and programming idioms that can cripple performance. In object-oriented terms, most of this work takes place at the method level, on the order of actual lines of code, loops, and so on.

2. Responsiveness:

The applications that feel slow, hang or freeze for significant periods, or take too long to process input are insufficiently responsive and will frequently cause the system to pop up "Application Not Responding" (ANR) message. Generally, this happens if application cannot respond to user input. For example, if application blocks on some I/O operation (frequently a network access), then the main application thread won't be able to process incoming user input events. After a time the system will conclude that your application has hung, and give the user the option to kill it. The fix is usually to create a child thread, and do most of your work there. This keeps the main thread running, and prevents the system from concluding code has frozen.

3. Seamless:

There are some seamless parameters, we should concern while designing and implementing system. These are:

a. Do not drop data:

If the user was editing data in your application when the other Activity appeared, your application will likely lose that data when your application is killed. A classic example of a good use of this behaviour is a mail application. If the user was composing an email when another Activity started up, the application should save the in process email as a draft.

b. Do Not Interrupt User when he/she is talking:

If the user is running an application (such as the Phone application during a call) it's a pretty safe bet he did it on purpose. That's why you should avoid spawning Activities except in direct response to user input from the current Activity.

c. Got a Lot to do? Take it to a Thread:

If your application needs to perform some expensive or long running computation, you should probably move it to a thread. This will prevent the dreaded "Application Not Responding" dialog from being displayed to the user, with the ultimate result being the fiery demise of your application.

d. Avoid Huge Activities:

Any application worth using will probably have several different screens. When partitioning your UI, be sure to make effective use of Activities.

e. Assume the Network is slow:

You should always code your applications to minimize network accesses and bandwidth. You can't assume the network is fast, so you should always plan for it to be slow.

4.2.3 Design

Android from Google is an operating system for mobile phones that is based on Linux. The architecture of applications for Android introduces concepts that go beyond the classical models.

The data-view model

This is the model used by the library Swing of Java notably. The view is the interface through which the user interacts with the software. Data are stored separately and can be displayed in different views. The view may also change the data according to the context, for example, change the text according to the user's language.

The Android model

Android extends this views/data model, it provides a new model that is suitable for equipment activated at all times. The structure of applications is defined as follows:

The views (Class android.view.View)

The interface of a program for Android is a tree of views. The image at right shows four views inside a screen (an activity).

The file AndroidManifest.xml

It defines the components of the application and their relationships. It gives the permissions to application as to what it can do with users. It can also give permission to components of the application.

The components of the application:

- Activity:
  This is something that the user can do, translated into program. It corresponds to a screen, but can have multiple views.

- Intent:
  Describes an action which must be performed.
- Service:
  Program that operates in background.
- Content Provider:
- It encapsulates data and provides them commonly to several programs.
- Notification:
  Class which informs the user about what is happening.

4.2.4 healthExpress Application Design

Our application design composes of 3 logical layers. These are Notification Layer, Service Layer and Activity Layer, as shown in Figure 4.4. These 3 layers are designed according to Android platform specifications which indicated in development guide of platform web site. As we mentioned in "The Android Model" section, Android application model uses 5 components; Activity, Intent, Service, Content Provider, and Notification. We have used 3 component types in our application design which are activity, service and notification. Also there is an external system which communicates with our mobile application. The external system is a Firebase Back-End as Service. Our client application uses defined interfaces to communicate with this external system. For example; users, doctors and medical services information are stored in here and our mobile client only communicates with Firebase by using defined interfaces.

Activity layer is composed of 5 activities which are responsible from interaction with user. Each activity capable of directly communicate with external system by using defined interface and protocols. "Login Activity" is start point of application.

Basically it is a login screen and appears when application is started. This activity requires 2 parameters from user which are username and password. After validate these parameters, client sends these parameters to external system and waits for authentication confirmation. If authentication is granted, login activity starts service which runs in background. Then, redirects screen to "User Home Activity". In user home activity, user can search and view required medical information. When activity is started it communicates with external system by sending activity identifier and external system return user data which will display in home screen.

Another important activity is "Search activity". User enters some search parameters and activity directly sends these parameters to server and receives results from server. Search activity also display result and by selecting an option from result list, screen is redirected to "Option Detail View". Option Detail View directly communicate with server and send item id and receive option information which contains various details. If user needs to add, modify data he/she can offer by using this activity by sending data directly to server. Our last activity is "Alert Popup Activity" which is used for displaying notification send by server. These could be a new prescription send by the doctor.

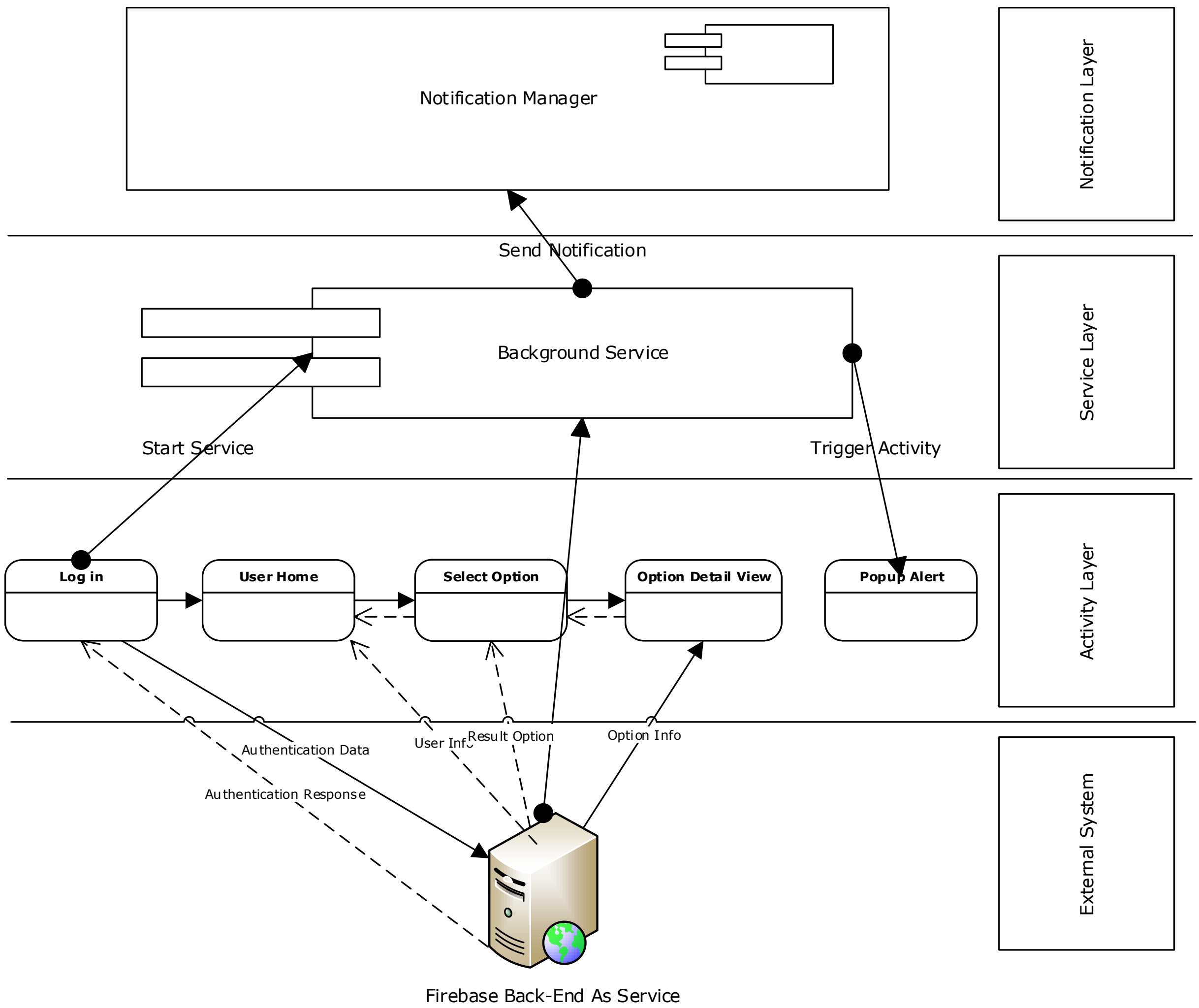


Figure 4.4 Mobile Application Component Design

Server Layer contains only one component which is Background Service. This service is started when user successfully logged in application. This service runs until user log out. This component can always receive data from Firebase and this is the most important functionality. But there are some situations which application should pay attention. If user is speaking using the phone, service does not use alert popup. Instead of alert popup, it uses Notification manager to notify user.

Notification Layer has one component which is Notification Manager. This component manages notifications like new SMS is received or missed calls. These notifications appears one of the corner of the mobile phone screen which similar to mail icon in a standard mobile phone. Our notification Manager is triggered by background service in case of user is talking with mobile phone and small icon appears right corner of the screen. When user clicks this icon, our Alert Popup Activity is displayed by showing notification which is received from Firebase.

To serve the health care need of our massive population we need to find innovative solutions and using IT enabled services to achieve these goals is both cost effective and less time consuming. Mobile health applications are gaining popularity so mobile operating systems present new design principles to their users. We started this project by identifying the requirements both functional and non-functional ones. Afterwards we analysed an applicable design to validate these requirements. We developed an architecture using the 3 logical layers supported by Android, these are Notification Layer, Service Layer and Activity Layer. A proper software architecture documentation will further add value to this project.

Chapter 5: Implementation

5.1 Data Model

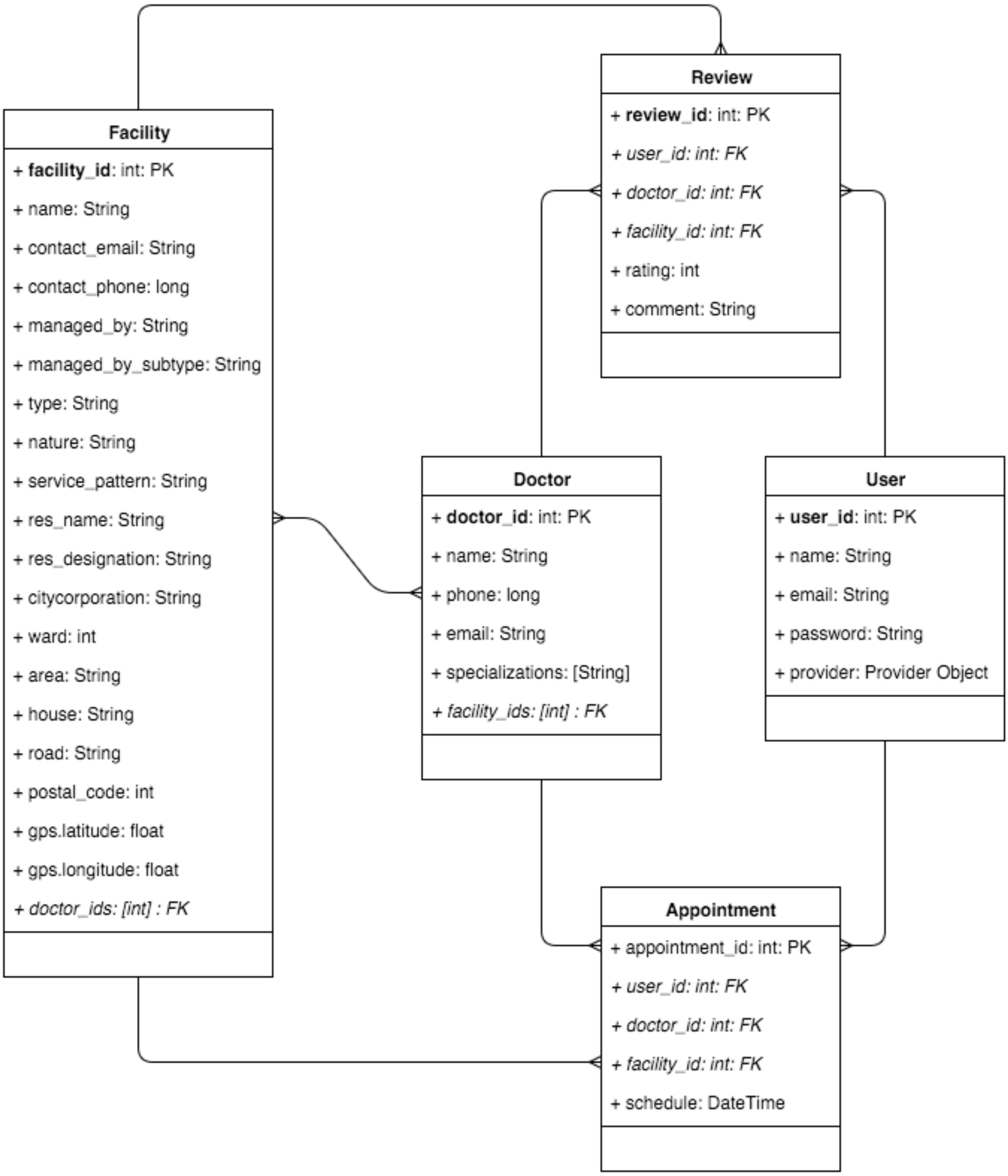


Figure 5.1 The basic data model of the prototypical application

The basic data model that the system is based upon can be seen in above figure. The central element of the data model is the Facility. It contains all facility data, such as the facility name, description, facility type and contact details. Each facility also includes GPS information to locate the facility on map. Besides Facility there are Doctor, User, Appointment and Review entity. Each Doctor entity includes doctor name, contact details, and a facility ID where the doctor is practicing. The User entity includes data for general public who uses the application on their smart phone. This entity stores user name, user email address and password for the user to access the application.

The application receives most of its model content from a database server with the root node located at https://health-express.firebaseio.com/. More information on the system architecture can be found in following section.

Figure 5.2 Data model as a simplified JSON file on Firebase

On the server, all data is stored as JSON objects, thereby increasing the extensibility of the prototype. A flat hierarchy is implemented so that loading one node does not also invoke loading its potentially vast number of children. This is especially useful when searching for keywords in the title or tags of all facilities –

the location and doctor elements of the facility do not automatically have to be loaded as well.

5.2 System Architecture

The system is designed as a simple model-view-controller application. The views are constructed in the Ionic framework and stored in a files templates folder in the project www directory. Each view is connected to a controller that loads and sets the dynamic view content from the server and carries out button actions. Any server communication is done using a singleton instance of the class FirebaseService. This class contains all calls that create, update, remove and read content on the server as well as methods for the parsing of JSON objects from the server and converting the local model to JSON objects. In order to communicate with the server, the FirebaseService class makes use of the Firebase SDK that is part of the free Hacker Plan offered by the service Firebase.

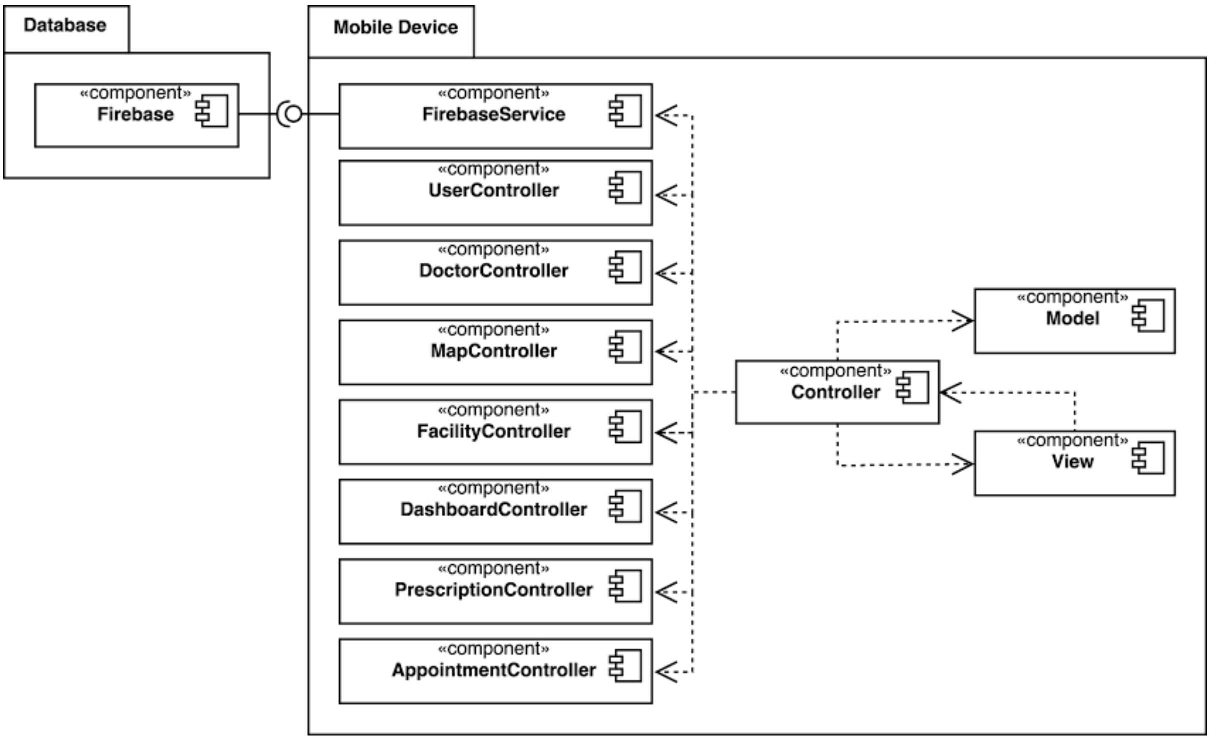


Figure 5.3 Architecture of the mobile application

In case the user of the application is not authenticated when accessing the application, the class UserController offers methods to add this user in firebase. The DoctorController encompasses basic methods for reading, adding, editing data objects while the MapController mainly assist to display map, placing markers using google maps API. FacilityController provides methods for working with facilities, e.g. listing, rating facilities into JSON-conform byte64 strings. These seven classes – FirebaseService, UserController, DoctorController, MapController, FacilityController, DashboardController, PrescriptionController and AppointmentController – were created to reduce the code redundancy in the view controller classes and to consolidate similar methods in a respective class. The target operating system for the application is both latest version of Android and IOS for mobile devices and Ionic framework has been used as development tool.

5.3 User Interface

Figure 5.4 Log in and Dashboard screen mockup

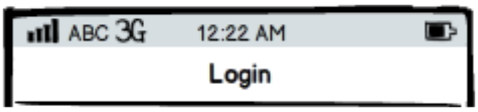
For generating a feeling for the application without having to write any code, the first version of the user interface was simply mocked-up using (Balsamiq Mockups). Trying out the application as a simple PDF file with clickable buttons resulted in noticing and resolving a number of shortcomings. Since the application targets novice users with little to no experience to search medical entities using a mobile application, its user interface needed to be as intuitive and easy-to-use as possible. Therefore, a few different alternatives were reviewed that structured the content of the application differently. A few examples of this user-friendly restructuring can be seen in the screens below.

The 'Login' screen is the first functional screen the user is presented with when starting the application. The image on the left in above figure depicts the 'Login' screen in which links to accessing the application using email and password or users existing Google sign in credentials.

The next mockup is the 'Dashboard' screen that appears on users screen when a user log in successfully. All functions and the user's personal content were displayed on one page. However, this screen appeared to be too overwhelming as a user's first look into and impression of the application and, as a result, a tab bar at the bottom of the screen has been introduced. It divides the application into four main areas – Dashboard, Doctors, Facilities and Account – that can be easily navigated to from any screen.

Figure 5.5 Doctor and Facility screen mockup

Each tab bar item is connected to a navigation controller that stacks view controllers (objects managing the content of the screen) on top of each other in the order in which they were activated. It allows for navigating to new screens via segues and back to old screens with chevrons in a navigation bar at the top of the screen like next pair of images.



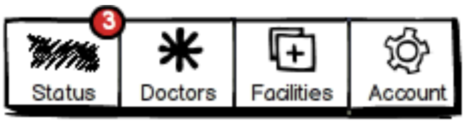


Figure 5.6 Header and Footer mockup

A better visualization of the hierarchy of the tab bar controller, the navigation controllers and the view controllers can be seen in below figure.

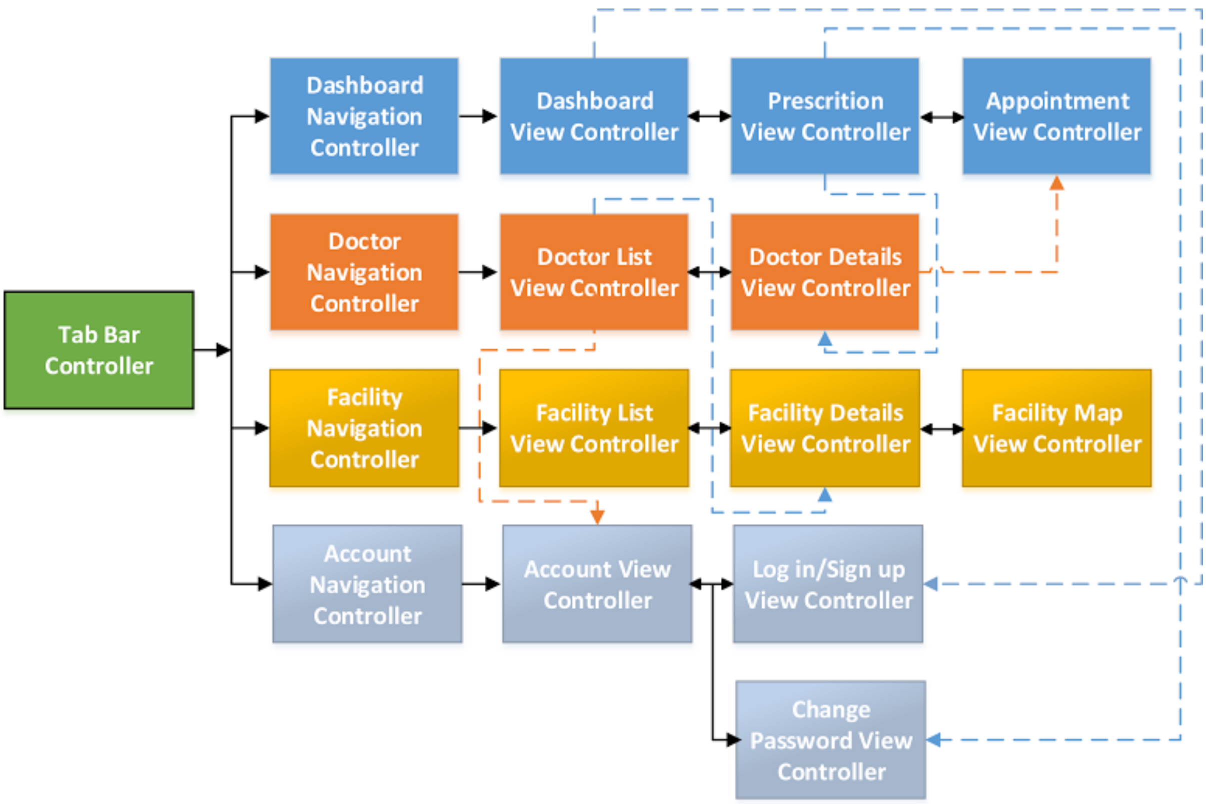


Figure 5.7 The app controller hierarchy

The arrows with solid lines represent the natural segues available via the navigation controllers, while the arrows with a dotted line point to a more complex changing of views. There, the tab bar item and the corresponding navigation controller have to be programmatically modified to switch between views.

After having settled on a basic version of the user interface using mockups, this version was implemented and more changes and modifications were gradually added as easier alternatives were noticed and too complex structures were reduced. The following sections further detail the final structure, look and content of all tab bar items. The colour palette that was used was based on the Ionic frameworks built in CSS libraries. The prototypical application uses icons from the sets (Ionicons) and (Octicons).

5.3.1 Log in screen and Dashboard

When a user downloads the application from Google Play store or App Store (IOS) for Apple devices and install the app on their mobile device the first screen a user

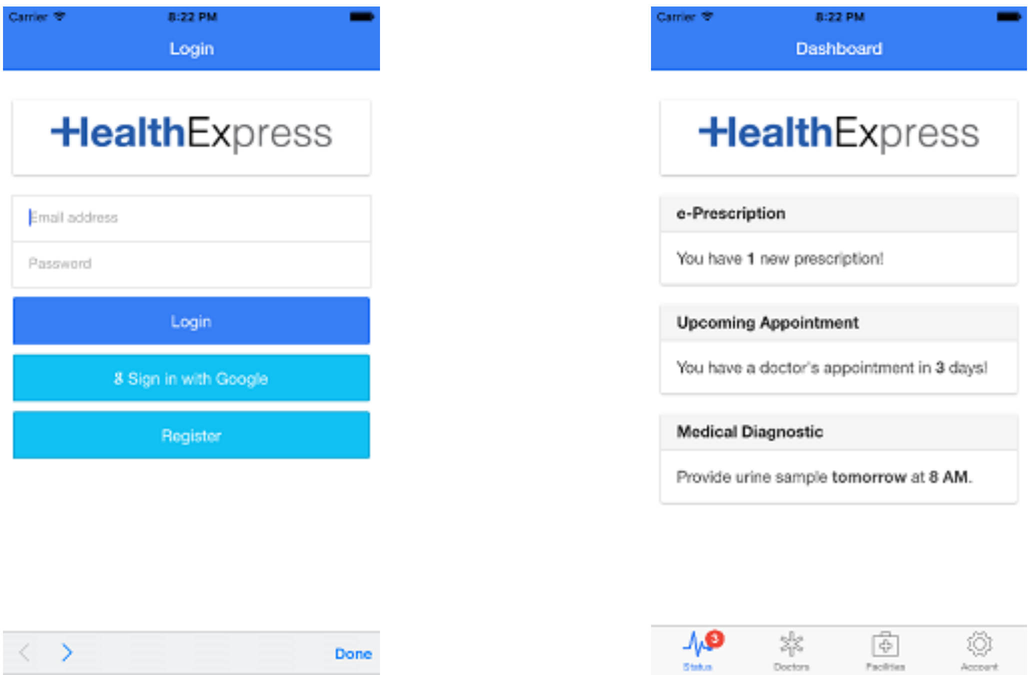


Figure 5.8 Log in and Dashboard screen

see is the Login screen of the app. A user is can register using email and desired password or users can also sign in using their existing google credentials.

Once a user successfully logs in she or she will view the Dashboard screen where a new user will see no details however a returning user can view their medical details if he or she has used the app to book appointment with a doctor and scheduled to provide medical samples to any diagnostic centre. User also gets social media like notification on the footer from the Status icon.

5.3.2 Doctor View

Users can navigate to the Doctor's tab from the footer of the Dashboard screen and view the Doctor search screen if they wish to search a doctor. Once users view the doctor's listing after performing a search they can also tap a particular doctor to view details of that doctor.

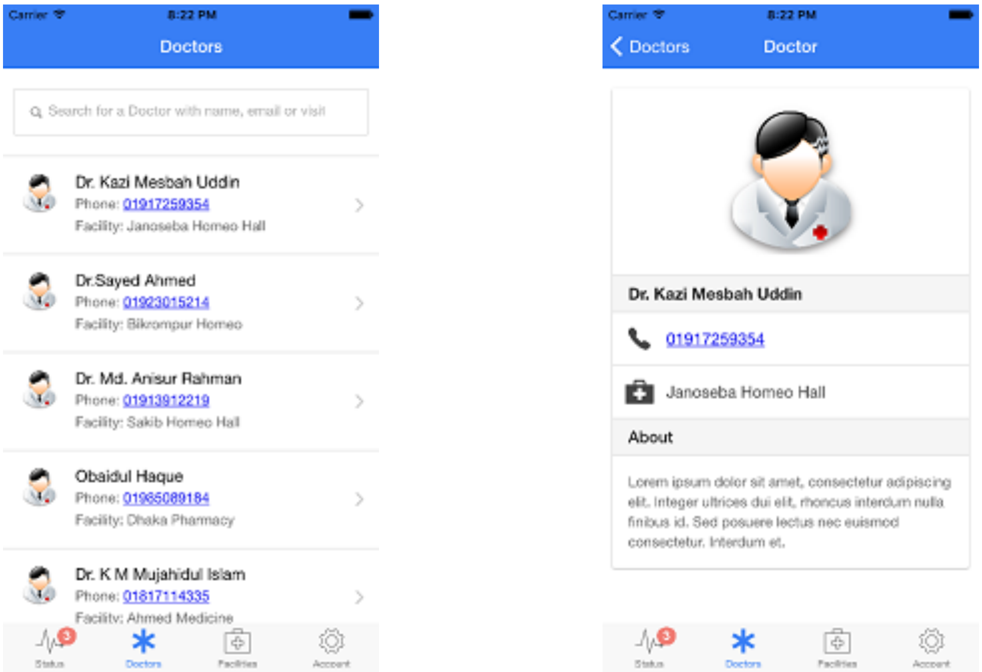


Figure 5.9 Doctor listing and Doctor details view screen

5.3.3 Facility View

User can also navigate to Facility view from Dashboard or from another view by simply tapping the Facilities icon on the footer of any views. Once a user enters the facility view he or she can see the listing of facilities based on different types and can choose a type desired.

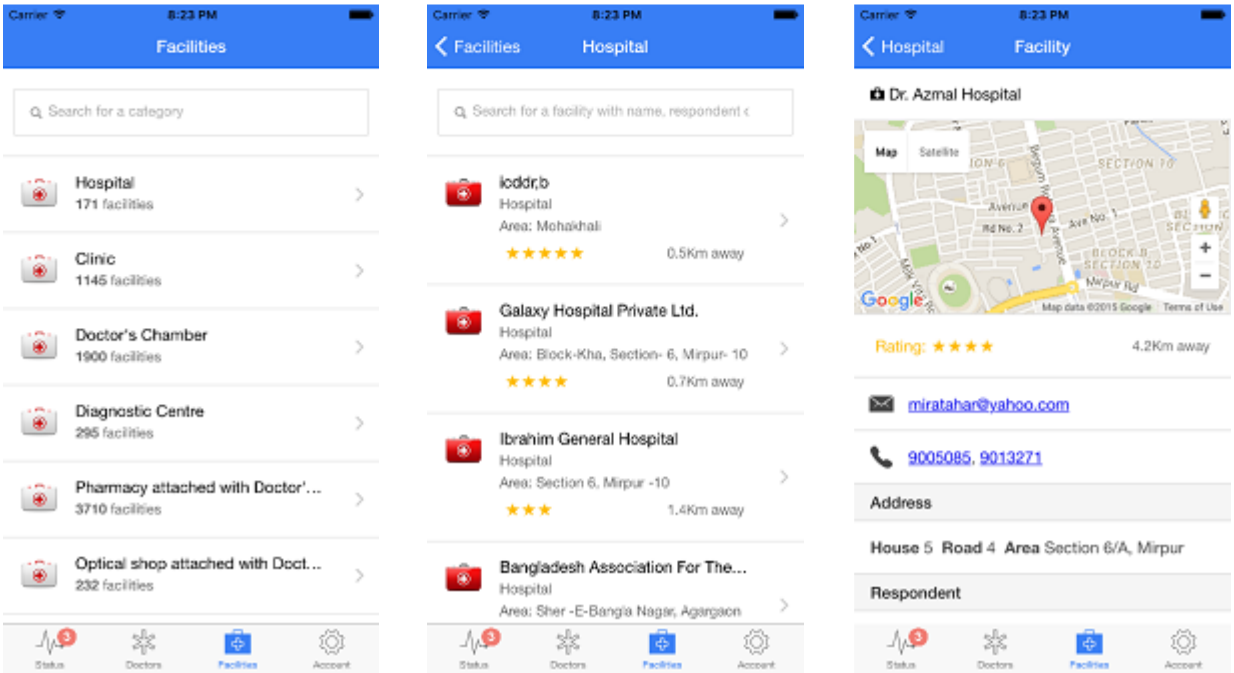


Figure 5.10 Facility listing and Facility details view screen

For example if an user taps and selects Hospital he or she will enter the Hospital search view where the user can perform a hospital search based on required parameter. Afterwards the user can select a Hospital from the list to enter the detailed view of the Hospital with location on Google Map and other relevant details of the Hospital.

5.3.4 User Account

A user can view the account settings using the Account icon on the footer of any view. In the Account view users can see if he or she has any other connections using the app should the user choose to use the app using their social media credentials.

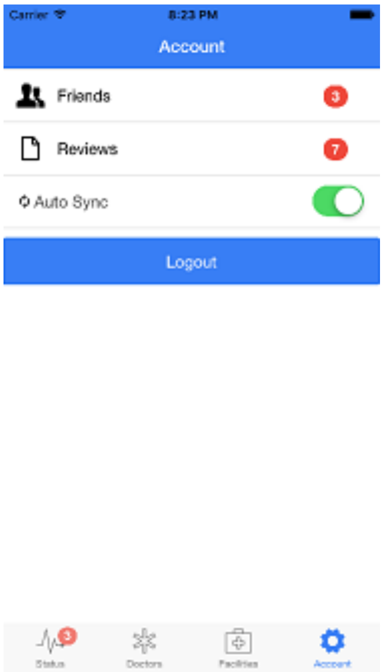


Figure 5.11 Account setting view screen

The user can also view how many reviews he or she has made while using the app and lastly the user can turn on auto sync so that whenever the apps become online it collects necessary updates relevant to the user for example- Doctor's prescriptions, appointment notifications etcetera.

Chapter 6: Conclusion

6.1 Conclusion

In conclusion, the work and contribution of this paper is summarized below. A literature review was conducted that revealed a need for applying modern technologies to eliminate the practice of medical consultation by unregistered doctors. Mobile technologies such as application were examined to determine how this a mobile application could be used to solve this problem. Requirements for a mobile application that helps to resolve this issue were elicited and a prototypical application was implemented. Furthermore a literature review was also conducted to identify what are the major software engineering challenges to develop and mobile application to solve healthcare problems. Four major issues that were identified are Creating Universal User Interfaces, Enabling Software Reuse across Mobile Platforms, Designing Context-Aware Mobile Applications, Balancing Agility and Uncertainty in Requirements.

6.2 Future Work

While the potential of the application and its supportive structure for novice users are apparent, there is a need for a review of the user interface of the application in order to decrease its complexity and introduce an interface with Bangla font for users. Apart from a user interface re-design, there are also many capabilities and functionalities that could potentially increase the usefulness of the prototypical application. New elements such as possibility to view medical diagnostic report on smart phone and transmit such reports to desired doctor or clinic/hospital could be included to further distinguish the application from available mobile health care applications. Another useful addition would be sharing possibilities and login via social platforms which could increase the usage rate of the prototype. Due to a lack of time, the features mentioned above could not be implemented however in future these areas of improvement should be explored. Also it was not possible to conduct an evaluation of the application by distributing it among a selected group of users. It is very much necessary that in future an evaluation is conducted to find out potential shortcomings of the application both from software and hardware perspectives.

References

1. The World Factbook. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2119rank.html#bg. [Online; accessed 16 November 2015].

2. BBC. http://www.bbc.com/news/world-asia-33337787. [Online; accessed 16 November 2015].

3. The World Factbook. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2151rank.html#bg. [Online; accessed 16 November 2015].

4. Levy, D. (2012). Emerging mhealth: Paths for growth. www.pwc.com/mhealth. Accessed 12 April 2013.

5. The Daily Star. http://archive.thedailystar.net/beta2/news/8-fake-doctors-jailed-in-ctg/. [Online; accessed 08 September 2015].

6. News Age. http://newagebd.net/33490/fake-doctor-jailed-in-city/. [Online; accessed 08 September 2015].

7. Dhaka Tribune. http://www.dhakatribune.com/crime/2014/sep/05/five-fake-dentists-jailed-nine-months-capital. [Online; accessed 08 September 2015].

8. Blaya JA, Fraser HS, Holt B: E-health technologies show promise in developing countries. Health Aff (Millwood) 2010, 29:244-251.

9. Lindquist AM, Johansson PE, Petersson GI, Saveman B-I, Nilsson GC: The use of the Personal Digital Assistant (PDA) among personnel and students in health care: a review. Journal of Medical Internet Research 2008, 10:e31.

10. ITU: Mobile overtakes fixed: Implications for policy and regulation. ITU International Telecommunications Union 2003.

11. Orbicom-ITU: From the digital divide to digital opportunities: measuring infostates for development. Canada 2005.

12. Healy, J. C. (2007). The WHO eHealth resolution - eHealth for all by 2015? Methods of Information in Medicine, 46, 2–5.

13. Van Bemmel, J. H. (1984). The structure of medical informatics.Medical Informatics,9, 175–180. doi:10.3109/14639238409015187

14. Valenčík, R. (2003). Theory of a productive consumption (Teorie produktivní spotřeby). Praha, Czech Republic: Vysoká škola finanční a správní o.p.s.

15. Friedman, M. (1967). A Theory of Consumption Function. Princeton, NJ: Princeton University Press.

16. Stiglitz, J. (2000).Economics of the Public Sector. New York: W. W. Norton & Co Inc.

17. B. Shneiderman, "Designing the user interface," 1987.

18. A. I. Wasserman, "Software engineering issues for mobile application development," in Proceedings of the FSE/SDP workshop on Future of software engineering research - FoSER '10, 2010, pp. 397-400.

19. G. C. Roman, G. P. Picco, and A. L. Murphy, "Software engineering for mobility: a roadmap," in Proc. of the Conf. on the Future of Software Engineering, 2000, pp. 241–258.

20. J. Gong and P. Tarasewich, "Guidelines for handheld mobile device interface design," in Proceedings of DSI 2004 Annual Meeting, 2004, pp. 3751–3756.

21. T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach," in 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the, 2003.

22. AHIMA Guide. (22 April 2013). Mobile apps 101: A primer for consumers. http://myphr.com/Stories/NewsStory.aspx?Id=596. Accessed 01 October 2015.

23. Estrin, D., & Sim, I. (2010). Open mHealth architecture: An engine for health care innovation. Science, 330, 759–760.

24. Malvey, D. and Slovensky, D. 2014. mHealth. 1st ed. New York: Springer.

25. Agile Manifesto. (27 October 2001). Manifesto for Agile Software Development. http:// http://www.agilemanifesto.org/principles.html. Accessed 01 October 2015.

26. Android (2015), Android Application Fundamentals. Avaliable in: http://developer.android.com/guide/components/fundamentals.html. Accessed 05 October 2015.

27. Structuring Data – Firebase (2015). JavaScript Web Guide Structuring Data. https://www.firebase.com/docs/web/guide/structuring-data.html. Accessed 02 October 2015.

28. Balsamiq Mockups. Available online at http://balsamiq.com/products/mockups/, Accessed 13 October 2015.

29. Ionicons. Available online at https://www.iconfinder.com/iconsets/ionicons, Accessed 13 October 2015.

30. Octicons. Available online at https://www.iconfinder.com/iconsets/octicons, Accessed 13 October 2015.