DESIGN AND IMPLEMENTATION OF AN iOS BASED MEDICATION REMINDER APPLICATION

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
SADDAM HUSSAIN RADI OBAIED

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Information Systems

NICOSIA, 2018

SADDAM HUSSAIN RADI OBAIED

DESIGN AND IMPLEMENTATION OF iOS BASED MEDICATION REMINDER APPLICATION

NEU
2018

DESIGN AND IMPLEMENTATION OF AN iOS BASED MEDICATION REMINDER APPLICATION

A THESIS SUBMITTED TO THE GRADUATE
SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY

By
SADDAM HUSSAIN RADI OBAIED

In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Computer Information Systems

NICOSIA, 2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name:

Signature:

Date:

To my family…

ACKNOWLEDGEMENT

I would first like to thank my family all of them .to the soul of my father may god rest him in peace.to my mother the greatest woman I know, my brothers and sisters for their love and support for their wishes and prayers.

All gratitude to my thesis advisor Prof. Dr. Dogan Ibrahim of the computer information system faculty at near East University. he was there when I ever need, for his wise directions and his full support and encouragement. And i would to thank all instructor in CIS department especially Prof. Dr. Nadire Cavus, Assist. Prof. Dr. Seren Başran and Dr. Ali Tüzünkan.

Finally, I must express my very profound gratitude to my friends (Ali, Omar, Thaer, Bashar, khalid, zaid) for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

ABSTRACT

Good health has been a major concern since the inception of mankind whilst for some people attaining good health requires taking prescribed medicines or pills routinely. However, many patients find it very difficult to keep track of taking their medication in the right time and proportion. This happens especially if it involves taking pills or medication on daily basis due to several reasons such as heavy work load, forgetfulness, old age and alterations in day-to-day behavior can also have a significant result on whether patients recall to take their prescribed medications which can be termed as medicine adherence, which is a very serious problem because it may affect the total well-being of the patient, delaying the curing time, raising the total medical cost of the patient and can be a matter of life and death. The aim of the study is to design and develop an automated iOS based mobile application for medicine or pill reminder as prescribed by a doctor to patients using the Xcode integrated development environment for Mac OS. The recent advancement in technology has provided an enabling technique to solve these types of problems by designing and developing an application that can run on smart phones in which patient will find it easy to carry along. The iOS based application was written using the swift programming language while adopting the Xcode IDE for the development. The medication reminder application could impact positively on the life of the patient as it will help patients in keeping track of their daily intake of pills as remembering the intake of these prescribed medications could be a matter of life and death. The novelty of this project is that a Doctor could communicate and also keep track of his patient daily intake of pills from his own backend.

Keywords: Medication reminder; mobile app development; smart phones; swift programming; Xcode iOS

ÖZET

İyi ve sağlıklı olmak insanlığın ilk başlangıcından bu yana çok önem taşımıştır. Bazı insanlar sağlıklı olmak için reçeteli ilaç veya hapları rutin olarak almaları gerekmektedir. Bununla birlikte, birçok hastalar ilaçlarını doğru zamanda ve doğru oranda almayı çok zor bulmaktadırlar. Bu, özellikle iş yükü, unutkanlık, yaşlılık ve günlük davranışlardaki değişiklikler gibi nedenlerden dolayı günlük olarak hap veya ilaç almayı gerektiriyorsa, hastaların reçetelerini almayı hatırlayıp hatırlamadıkları konusunda da önem taşımaktdır. İlaca bağlılık olarak adlandırılabilecek olan bu durum çok ciddi bir problemdir, çünkü hastanın toplam sağlığını etkileyebilir, iyileşme süresini geciktirebilir, hastanın toplam tıbbi maliyetini yükseltebilir ve ayni zamanda bir yaşam ve ölüm meselesi olabilir. Bu çalışmanın amacı, iOS mobil akıllı cep telefonu tabanlı ve Mac OS için Xcode entegre geliştirme ortamını kullanan ve hastalara bir doktor tarafından verilen ilaç veya hapları hatırlatacak olan bir uygulama tasarlayıp geliştirmektir. Teknolojide olan bu son gelişmeler, bu tür sorunların çözülmesinde, hastaların taşıyabileceği kolay bulunabilecek akıllı telefonlarda çalışabilecek bir uygulama tasarlayarak ve geliştirerek, uygun bir teknik sağlamıştır. iOS tabanlı olan bu uygulamayı geliştirmek için Xcode IDE'yi kullanılmış ve böylece hızlı bir uygulama geliştirilmiştir. İlaç hatırlatma uygulaması, hastaların günlük kullanım miktarlarını ve zamanlartını takip etmelerine yardımcı olacağı için hastaların hayatlarında çok olumlu etkiler yapabilir, çünkü bu reçeteli ilaçların alımını hatırlamak, yaşam ve ölüm meselesi olabilir. Bu projenin yeniliği, doktorun kendisinin hastaların günlük hap alımlarını takip edebilmesidir.

Anahtar Kelimeler: İlaç hatırlatmak; mobil uygulama geliştirmek; akıllı telefonlar; swift programlama; Xcode iOS

TABLE OF CONTENTS

CHAPTER 5: SYSTEM IMPLEMENTATION

CHAPTER 6: CONCLUSION AND RECOMMENDATIONS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADF:	Application Development Framework
API:	Application Programming Interface
DS:	Data Synchronization
EMR:	Electronic Medical Record
GUI:	Graphical User Interface
HF:	High Frequency
IDE:	Integrated Development Environment
IOT:	Internet of Things
ISO:	International Standard Organization
IT:	Information Technology
Java ME:	Java Micro Edition
Java SE:	Java Standard Edition
JDK:	Java Development Kit
LED:	Light Emitting Diode
OMA:	Open Mobile Alliance
OS:	Operating Systems
PC:	Personal Computer
PDA:	Personal Digital Assistant
RFID:	Radio Frequency Identification
SDK:	Software Development Kit
UI:	User Interface
XML:	Extensible Markup Language

CHAPTER 1

INTRODUCTION

This chapter discusses the whole study background by introducing the design, development of the iOS based medicine reminder, study problem statement, study aim, study objectives, study significance and application limitation.

1.1 Background

There is a popular saying that health is wealth. Health is one of the most important thing for most individuals simply because not having good health can lead to a very miserable life (Leonard, 2008). In recent times new diseases have emerged which needs to be taken care off by taking medicine or pills routinely (Ohayon, 2002), for many people there are consequences attached to not taking the prescribed medication in proper time and proportion because it can be the difference between life and death. In addition, not taking the prescribed medicine at the proper dosage or time can also result to what is referred to as medicine adherence which can be stated to as the extent at which the medicine is taken at the correct doctor prescribed time and proportion (Hughes, 2004). Medicine adherence is a very serious problem because it may affect the total well-being of the patient, delaying the curing time and also raising the total medical cost of the patient (Grooves et al., 2013). Yet with all the aforementioned consequences people of different ages still forget to take their prescribed medicines or pills in due time and proportion (Banerjee, 2009). However, there are different reasons for been forgetful ranging from busy schedules, old age, cognitive disorders, bad working conditions, Alzheimer disease, loss of memory, dementia, people with emotional problems, stress, anxiety, depression etc.

Nevertheless, the recent advancement in technology has provided an enabling technique to solve these types of problems using different methods, one of the methods used is by buying a device designed purposely to remind patient to take their medicines in the prescribed time and proportion, the aforementioned solution seems to be ineffective and costly (Riehemann et al., 2009). Rather, the use of mobile application seems to be more effective as it does not necessitate a need to procure an additional device and because most people make use of smart phones. The study chose to adopt one of the most widely used smartphone OS which is the iOS because they are the top notch in the smartphones business. However, the iOS seems to be very effective in mobile phones according to the top notch engineers (Nosrati, 2012).

The iOS is an operating system designed and developed by the Apple Inc. mainly for the Apple hardware gadgets such as the iPhone, iPad and iPod Touch, the iOS was developed starting from the earliest stage empower engineers to make convincing versatile applications that take full preferred standpoint of each of the handset. The operating system is mainly created for mobile devices and has been the second most widely adopted after the Android OS. For this reason the proposed mobile application is compatible with smartphones running on one of the most popular mobile based operating system which is the iOS, basically the application helps to remind patients or users to take their medicine in due time and actual proportion using an automated alarm ringing system. In addition, the proposed system helps a physician in keeping track of his patient medicine intake from his own backend, this aforementioned feature is absent in most medicine reminder applications.

Hence, the aim of the study is to design, develop and implement an automated iOS based application for medicine or pill reminder as prescribed by a physician to patients using the swift programming language while adopting the Xcode integrated development environment coupled with some iOS API's. The designed application would help patient to maximize the full benefit of the medicine and abstain from the risk that result as not taking the medicine or pill within the stipulated time prescribed by the specialist.

1.2 Problem Statement

Health is one of the most important thing for most individuals simply because without a good health nothing seems to go well. In recent times new diseases have emerged which needs to be taken care off by taking medicine or pills routinely. In addition, the working conditions of some individuals is bad and hectic and as such had resulted them in forgetting to take their administered medicine or pills in the appropriate timing or proportion and for some people its old age, most elderly people suffer from dementia i.e. forgetfulness. Not taking the prescribed medicine at the proper dosage or sometimes could result to what is referred to as medicine adherence which can be stated to as the extent at which the medicine is taken at the correct doctor prescribed time and proportion. Medicine adherence is a very serious problem because it may affect the total well-being of the patient, delaying the curing time and also raising the total medical cost of the patient. As such the design and development of an iOS based medicine or pill reminder could help in curbing out the aforementioned problems by reminding patients to take their medicines as prescribed by the doctor within the stipulated timing and dosage.

1.3 Aim of the Study

This study aim to design and develop an automated iOS based application for medicine or pill reminder as prescribed by a doctor to patients using the swift programming language and the Xcode integrated development environment.

1.3.1 Study Objectives

- To remind patients about taking their prescribed medicines or pills within the stipulated time as prescribed by a doctor.
- To support the physician monitor his patient in taking their prescribed from his own end.
- To use an alarm ringing system in making the remembrance in order to make patients stay healthy and fit.
- To design a medicine reminder that could support the iOS which is one of the most widely used OS with 11.9% Mobile OS users (Statista, 2018).

1.4 Significance of the Study

- The study demonstrates how to design, develop and implement the iOS based medication or pill reminder mobile application by making an alarm to the patients, using Xcode integrated development environment.
- The application supports the physician in monitoring his patient about their prescribed pill intake.

- The designed application would help patients to maximize the full benefit of the medicine and abstaining them from the risk that result as not taking the medicine or pill within the stipulated time and proportion as prescribed by the specialist.
- The designed iOS based medicine reminder will be of great help for patients suffering from range of problems such as forgetfulness, busy schedules, old age, cognitive disorders, bad working conditions, Alzheimer disease, loss of memory, dementia, people with emotional problems, stress, anxiety, depression, and also individuals with a very hectic work schedules or lifestyle.
- The application help patients staying fit by remembering them to take their medicine in an appropriate time and proper proportion

1.5 Study Limitations

The study is limited to designing, developing and implementing an iOS based mobile application for reminding patients to take their medicine in the prescribed appropriate proportion and timing using the swift programming language using the Xcode integrated development environment for the development along with some iOS libraries. Nevertheless, the designed application comes with some limitations which are stated as follows:

- The designed medicine reminder is completely platform dependent i.e. the application runs only on iOS based powered devices such as the iPhone, iPad, iPod touch.
- The designed medicine reminder has no cloud based backup system, in case the smart phone in which the application was installed got missing or faulty the whole data would be lost.
- The designed medicine reminder makes uses of the smart phone alarming system to notify the patient to take their medicine, if the patient changes the phone mode to silent the patient cannot hear the alarm to remind him/her to take their medicine in the prescribed appropriate time.

- To communicate with the physician, there is need for the application to be connected to an internet. Not connecting the device to the internet were the application is hosted will prevent a user from making any communication.

1.6 Overview of the Thesis

The whole study writes up consists of six chapters, the following list gives a brief overview of each chapter.

- **Chapter one:** This chapter discusses the whole study background by introducing the design, development of the iOS based medicine reminder, study problem statement, study aim, study objectives, study significance and application limitation.
- **Chapter two:** This study reviewed some literatures regarding the design, development and the implementation of medication reminder mobile application for patients from different academic sources.
- **Chapter three:** This chapter describes the system framework and relevant technologies that were used during the design, development and implementation of the iOS based medicine reminder application platform.
- **Chapter four:** This chapter discusses the iOS based medication reminder application architectural design and goals with the application development.
- **Chapter five:** This section of the thesis discuses about how the designed iOS based medical reminder application could implement step by step.
- **Chapter six:** This chapter finalizes the design and implementation of an iOS based application for medication or pill reminder while providing some recommendation on how to improve the mobile based application.

CHAPTER 2

LITERATURE REVIEW

This study reviewed some literatures regarding the design, development and the implementation of medication reminder mobile application for patients from different academic sources.

2.1 Design and Development of Mobile based Medication Reminder

Mohammed et al. (2018) designed and developed a smart application for medication reminder that was targeted at the elderly patients with various disabilities to help them in taking their prescribed drugs within the stipulated time. The developed mobile application is cloud based, in supposition the oddity of the created application is the utilization of a Cloud service to give two route correspondence between the more established patients with inabilities and the specialists so the pharmaceutical adherence of the patients can be checked, however the developed applications lacks an interactive medium between the patient and the doctor.

Ameta et al. (2015) designed, developed and implemented a medication reminder application, the program was designed based on the alert ringing framework to remind patient about the intake of their medications. It centers on the interaction between the two parties i.e. patient, doctor. No need for patients to keep a written dosage timing in a paper, they just need to enter these prescribed timings on the apps. The medication reminder could be set for various medical prescriptions with different timings. The app remembers the patients of taking their medicines via text messages. They can look specialist sickness shrewd. The patients will get the contact points of interest of specialists according to their accessibility. Additionally the clients can see diverse articles identified with medicinal fields and human services tips. The framework centers on simple route and a great user interface. Numerous such Medical Reminder Systems have been created where another equipment is required but yet in their work they have made an endeavor to build up a framework which is sparing, efficient and underpins solution adherence. But surprisingly the app adopted user interface was not very friendly to work with, which could affect the application user acceptance and also its usage effectiveness by most patients.

Zao et al. (2010) created a mobile application called Wedjat, it’s an advanced mobile phone application that causes patients to keep away from these oversights. Wedjat can remind its clients to take the right medications on time and keep an in-take record for later survey by medicinal services experts. Wedjat has two recognized highlights: (1) it can caution the patients about potential medication sedate/tranquilize nourishment associations and plan an in-take plan that maintains a strategic distance from this antagonistic cooperation’s; (2) it can amend an in-take plan naturally when a measurements was missed. In the two cases, their application dependably creates the most straightforward timetable with minimum number of in-takes. Wedjat works with the timetable application accessible on most PDAs to issue medication and supper updates. It additionally demonstrates photos of the medication and professional vides concise in-take guidelines. As a tele monitoring gadget, Wedjat can keep up medication in-accept records, synchronizes them with a database on a host machine or transfer them onto an electronic medical records (EMR) framework. Moreover, this application reminds a patient just once, and has its platform dependent.

Park et al. 2012 designed and developed a medication or pill reminder system centered on information or data synchronization. The mobile based application performs its operation by transmitting an open mobile alliance (OMA) and data synchronization (DS) based messages encompassing the medication or pill data of the patients and the device configuration data to a remote manager/medical staff. The mobile application also utilizes the medication server by synchronizing modified or generated data by a personnel. The application lacks an interactive medium between the patient and the doctor.

Slagle et al. (2011) designed and developed a mobile based application called MediHealth. Their application is a pill reminder framework for youngsters. The application keeps running on mobile phones, for example, advanced mobile phones, giving user interface to designing medicine calendars and patient alarms for reminding patients about the time and kind of medication or pills as per the arranged solution plan. This application lacks a cloud based backup system for its data.

However, some studies such as Becker et al. (2009), Ammouri and Bilodeau (2008) and Batz et al. (2005) designed and developed some application systems that adopted the radio frequency identification (RFID) sensors or some sort of detecting high technology to making sure that patients don’t get carried away from actually taking their pills or medicines as prescribed by doctors within the stipulated time. This approach is less effective than developing a mobile based application. And will have less user reach because these days most people use the smartphone.

Prasad (2013) has investigated an approach about designing and developing a pill or medicine reminder application. He designed a free application which underpins up to 15 reminders. Patient could select them in either non-rehashing or rehashing alert examples. In the planned time, application could deliver a warning with a vibration, ring alert, or LED signal. The application lacks an interactive medium between the patient and the doctor. The application got some obvious pop ups that could let others knows that one has to take their medicines.

Ilkko et al. (2009) designed and implemented a medicine dose controller of ubiquitous home environment (2009) called UbiPILL. Their application requires buying a device designed purposely to remind patient to take their medicines in their prescribed time and proportion, the aforementioned solution seems to be ineffective and costly (Riehemann et al., 2009)

Huang et al. (2014) design and developed an intelligent pill box, the usage of the pill box works by helping the elderly patients in remembering to take their prescribed pills or medicine which will give them a good wellbeing. The pill device also will recap the sick about planning by doing medication mishandling can be adequately managed. This technique requires buying a pill box purposely to remind patient to take their medicines in their prescribed time and proportion, the aforementioned solution seems to be ineffective and costly (Riehemann et al., 2009).

Ajmal Sawand et al. (2014) proposed a unique way to deal with accomplishing effective and dependable eHealth checking systems. The innovative converging between IOT, remote body zone system and distributed computing have imperative commitment in eHealth services which enhance the nature of restorative care, fundamentally quiet driven observing assume a part in e human services administrations which include therapeutic information accumulation, collection, information transmission and information examination here whole observing lifecycle and basic administrations segment have plate and additionally configuration challenges in outlining the quality and patient driven checking plan alongside potential arrangement.

Santo et al. (2017) conducted a research specifically on the ways to improve the adherence on medication by coronary heart disease patients. They exploited the available medication reminders applications for both the Android and iOS versions in determining the differences in medicine adherence between advanced and basic medicine reminder application. Surprisingly, the application lacks a cloud based backup system for its data.

2.2 Summary

With regards to the subject matter of this study, there are loop holes associated with the aforementioned related works.

- The systems are mostly designed to be platform dependent i.e. it is either compatible with the Android OS or the iOS. If the system is designed to run on the Android OS this implies that iOS users cannot use the application and vice versa.
- For most designs that were investigated by this study, it was surprisingly found that the adopted user interface was not very friendly to work with, which could affect the application user acceptance and also its usage effectiveness by most patients.
- For some design there is no option to modify the timing and the notifications are mostly compulsory in which some users could find not comfortable to work with.
- For some system design, there is an adoption of built in default alarm tone that boozes and no option for users to modify.
- Finally, some of the systems presented necessitate a special hardware that is needed to be procured while other applications need a lot of hardware processing power to function well.

CHAPTER 3

THEORETICAL FRAMEWORK

This chapter describes the system framework and relevant technologies that were used during the design, development and implementation of the iOS based medicine reminder application platform.

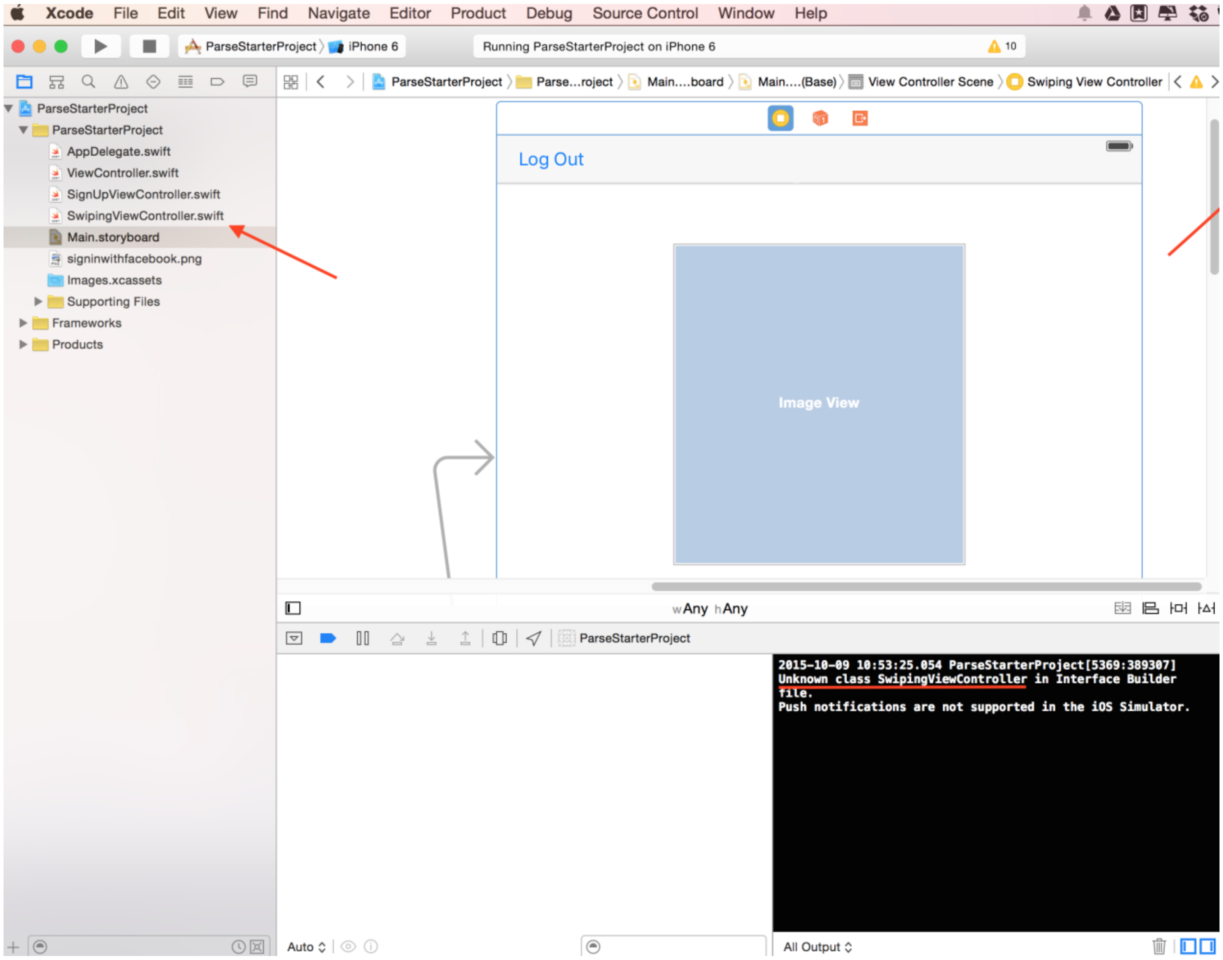
3.1 Xcode

To start off developing iOS applications, there is a requirement of installing the Xcode integrated development environment platform. This is because this is the only IDE tool that the Apple Inc. provides for developing its applications. However, the Xcode IDE comes with all the tools needed to design, develop and implement the iOS based applications. Tools such as the iPhone simulator for making software simulations without using an iOS based device, a rich editor for writing source code, making error debugging, and also provides a very rich graphical user interface. Moreover, the Xcode IDE comes with all the required bundles of iOS SDK libraries.

The Xcode represent the main part of the development tool for the OS X development environment (Rogers, 2009). Xcode is an absolute IDE that enables you to alter, order, investigate, and bundle Mac applications written in various programming languages (Iversen and Eierman, 2014). Regardless of whether you don't mean to utilize it for your essential Java development, it is useful to get comfortable with Xcode.

Xcode has a tool capable of helping in managing and organizing java applications using the Xcode organizer tool. The Organizer demonstrates your undertaking precisely as it is spread out in the record framework. This is as opposed to the primary Xcode venture windows, which enable you to organize records subjectively without adjusting their area on plate. The Organizer's immediate impression of the document framework better serves Java improvement and is like other Java IDEs.

However, the Xcode IDE comes in different version based on the adopted programming language used, but for this project the swift programming language was selected because of its simplicity and efficiency. The following Figure 3.1 depicts the Xcode user interface.



Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
ParseStarterProject 〉 iPhone 6
Running ParseStarterProject on iPhone 6
10
ParseStarterProject 〉 Parse...roject 〉 Main....board 〉 Main....(Base) 〉 View Controller Scene 〉 Swiping View Controller
ParseStarterProject
ParseStarterProject
AppDelegate.swift
ViewController.swift
SignUpViewController.swift
SwipingViewController.swift
Main.storyboard
signinwithfacebook.png
Images.xcassets
Supporting Files
Frameworks
Products
Log Out
Image View
wAny hAny
ParseStarterProject
Auto
All Output

```
2015-10-09 10:53:25.054 ParseStarterProject[5369:389307]
Unknown class SwipingViewController in Interface Builder
file.
Push notifications are not supported in the iOS Simulator.
```

Figure 3.1: Xcode user interface

Downloadable example code and the example code introduced in/Developer/Extras/Java/are both generally given as Xcode ventures. Moreover, there are a few components of documentation seeing that are accessible just through Xcode (Figure3.2).

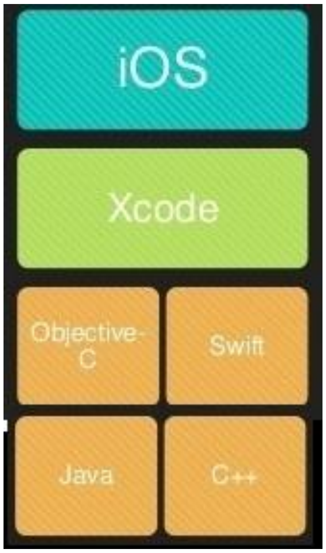


Figure 3.2: Overview of iOS and Xcode framework (Vaidya & Naik, 2013)

3.2 Swift Programming Language

The choice of programming language for this project is swift programming language because of its simplicity of usage and efficiency. For example the Java programming language is too bulky and difficult to learn. Moreover, another reason is that the programming language was developed by the Apple Inc. mainly for developing iOS applications.

3.3 Mobile Application

The development of mobile apps resembles that of the development of web applications but for mobile applications they are platform dependents meaning a developed solely for iOS cannot run on an android environment. Figure3.3 depicts the Mobile Development Framework used for designing, developing and implementing mobile applications. This project also utilized this framework for its design and development.

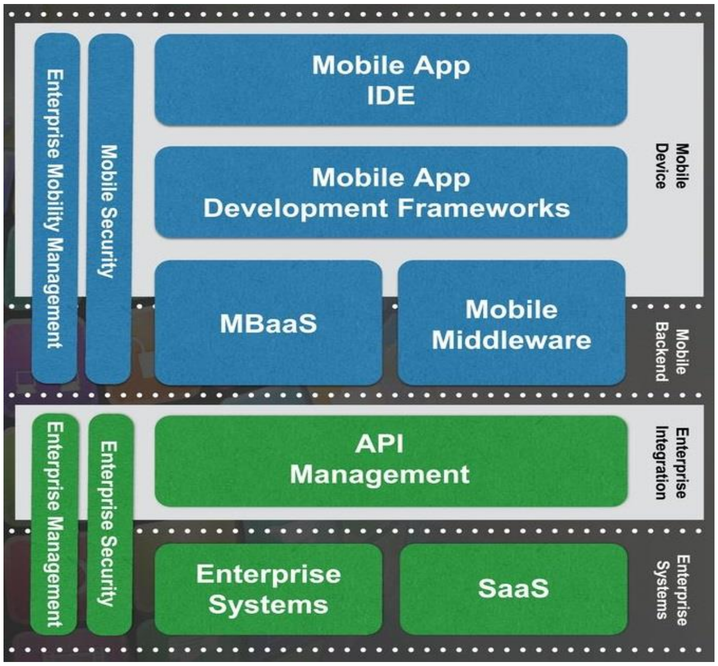


Figure 3.3: Mobile development framework (Kulathumani, 2015)

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

This chapter discusses the iOS based medication reminder application architectural design and goals with the application development.

4.1 Design goal: iOS Medication Reminder

The study design goal basically portrays a real life situation whereby a patient with cognitive disability, heavy workload or old aged is needed to remember when to take in his prescribed medicine because not taking the prescribed medicine in time and correct dosage could deteriorate the patient condition, more importantly it could be a matter of life and death. However, the goals were categorized into different classes namely:

- Program Performance
- Program Dependability
- Program End user Criteria
- Program Maintenance

4.1.1 The Program Performance Criteria

The part of the iOS based medication reminder application to be utilized for patient or doctor record keeping ought to have a quick reaction time (continuous) with greatest throughput. Besides, the iOS based medication reminder application ought not to take up excessively space in memory. The record user has picked quick reaction time over throughput and subsequently the iOS based medication reminder application should attempt to be more intelligent. On account of the timetabling subsystem, the iOS based medication reminder application ought to be more solid keeping in mind the end goal to fulfill the requirements than quick reaction time.

4.1.2 Program Dependability

There is need for the iOS based medicine reminder application to be exceptionally trustworthy as it can be relied upon to be utilized by non-IT experts. The iOS based medicine reminder application ought to be strong and blame tolerant. Moreover, as the system is taking care of sensitive patient data about their administered medicines or pills, high accentuation ought to be given with respect to security.

4.1.3 Program Maintenance

The iOS based medicine reminder application ought to be effortlessly extensible to include new functionalities at a later stage. It ought to likewise be effectively modifiable to roll out improvements to the new improved functionalities and features.

4.1.4 End User Criteria

End users i.e. the patients are associated with the program usability, usability is the degree to which a program can be utilized by indicated users to accomplish determined objectives with viability, proficiency and fulfillment in a predefined setting of utilization. From the end users' point of view the program ought to be planned such that it is anything but difficult to learn and utilize, effective and having couple of mistakes assuming any.

4.2 System Architecture

The newly designed and developed iOS based medicine reminder application is to automate the reminding of patient to take their prescribed medicine or pills within the proper stipulated time and proportion, the automation was carried out using the swift programming language alongside some swift standard libraries by utilizing the Xcode integrated development environment. The following Figure 4.1 depicts the iOS based application architecture.

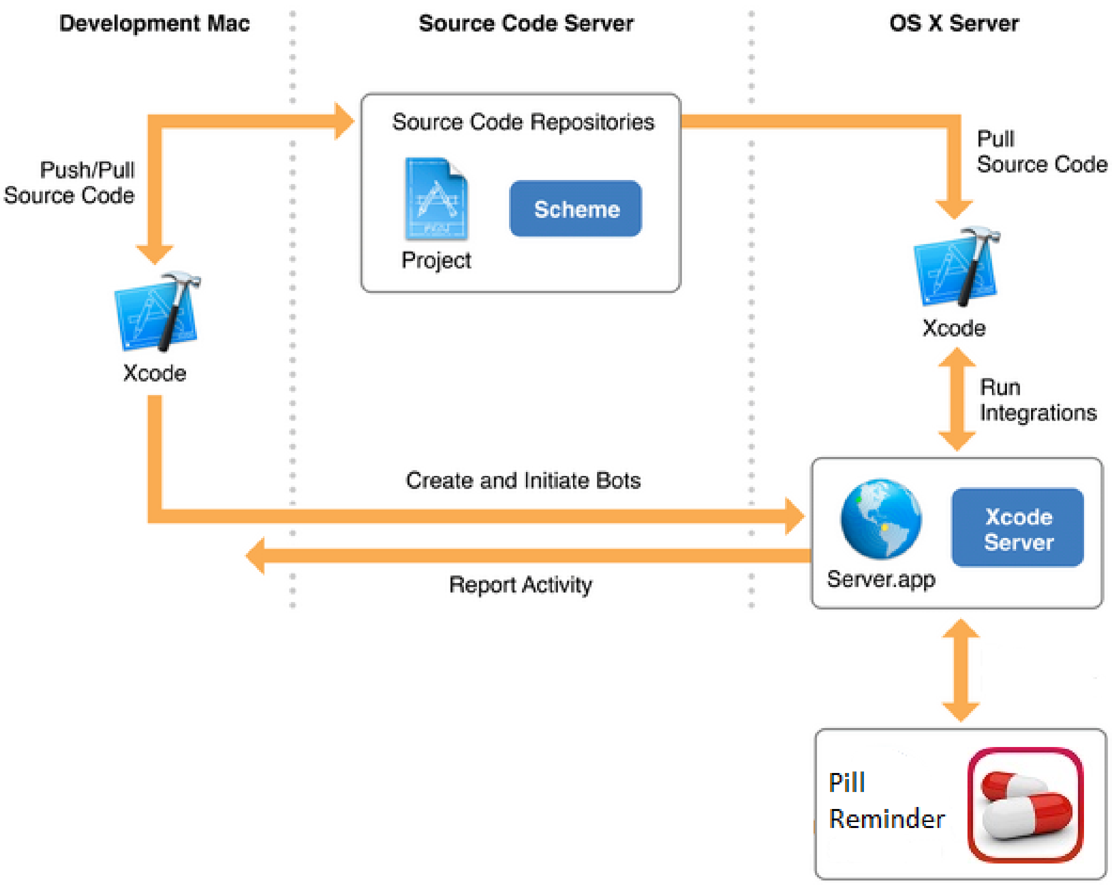


Figure 4.1: Xcode system architecture (Xinran et al., 2016)

4.3 Proposed Application Development

The proposed application is compatible only with smartphones running on the popular iOS based operating system, basically the application helps to remind patients or users to take their medicine in proper due time and proportion using an automated alarm ring system with also a physician backend.

The iOS is an operating system designed and developed by the Apple Inc. mainly for the Apple hardware gadgets such as iPhone, iPad and iPod Touch. The operating system is mainly created for mobile and has been the second most widely adopted after the Android OS. Hence, this means that the developed mobile application is compatible with smartphones running on the iOS platform, basically the application helps to remind patients or users to take their medicine in due time and actual proportion using an automated alarm ringing system.

The framework is based on the iOS working framework simply because the piece of the overall industry of iOS is high. iOS additionally accompanies an application development framework (ADF), which gives an API for application development and incorporates administrations for building GUI applications, information gets to, and other segment writes. The framework is intended to rearrange the reuse and reconciliation of parts. iOS applications are manufactured utilizing a required XML show record. The show record esteems are bound to the application at gather time. This record gives basic data to an iOS stage for dealing with the life cycle of the application. Cases of the sorts of data incorporated into a show record are portrayals of the application's parts among other structural and setup properties. Parts can be one of the accompanying sorts: Activities, Services, Broadcast Receivers, and Content Providers. The following Figure 4.2 shows the proposed system overview

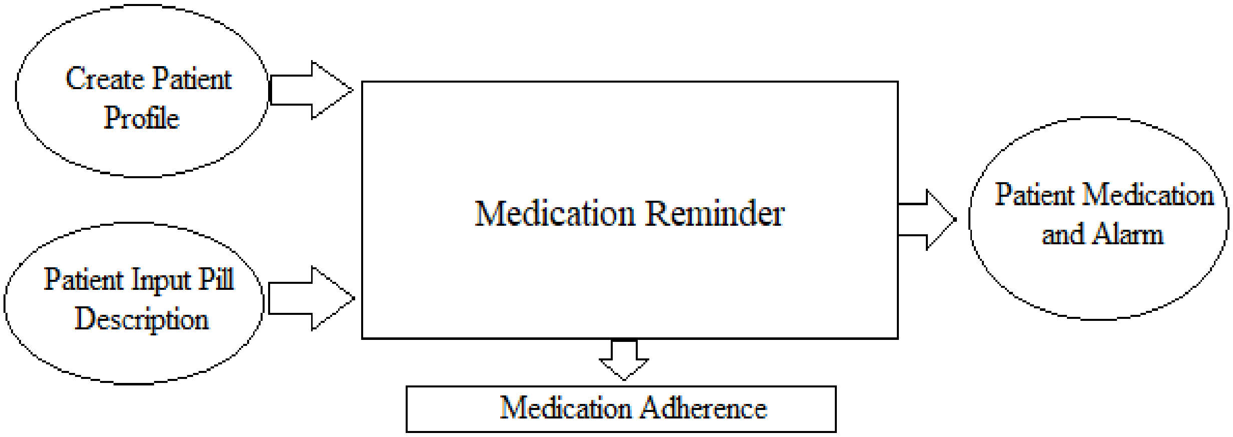


Figure 4.2: System review

The above Figure 4.2 shows the proposed system overview, the patient needs to create a profile in the first place, then entering the doctor medicine prescription which includes medicine name, date, time etc. The system output put focus on the medication adherence. Medication adherence can refer to as state of whether patients take their administered medicine in proper time and proportion. Recently, there is a growing number of medication adherence amongst patients in the world (Ho et al., 2009). The system has two major modules which are the alarm module and notification module.

- **Alarm module**: This module will assist in making the patient aware to take their prescribed medicine. Patient can include some details such as the administered dose. There is a field to include the starting and ending dates within which the medicine can be taking. There is a time field that shows when the alarm gets rung. Patient should be able to add medicine or pill description. The app stores the record in a system file which makes the record available whenever is called upon. Lastly, there is an option to change the alarm ringtone to the more preferred ringtone.
- **Notification module**: This module will assist the patient in getting a notification after the alarm in rung. However, the patient or user can deactivate or activate the notification in accordance. Basically, if the patient don't want the notification then he/she can modify the option vice versa.

4.4 Proposed Application Requirements

The following lists the make mention of the hardware and software's used for the development of the iOS0 based medicine reminder.

- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for And iOS SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- requires Mac OS X El Capitan 10.11 or higher.
- Xamarin.iOS requires Apple's Xcode IDE and iOS SDK.
- Xamarin.Forms can target iOS platforms, given the requirements listed above.
- macOS Sierra 10.12: Community, Professional, and Enterprise.
- Mac OS X El Capitan 10.11: Community, Professional, and Enterprise.

4.5 Application Flow Chart

The flow chart showing how the application work from the start then when the user get schedule of medicine then the system enter waiting if isn’t a correct time the system back to waiting, if the correct time coming then the system will notify and make sure the medication is taken .

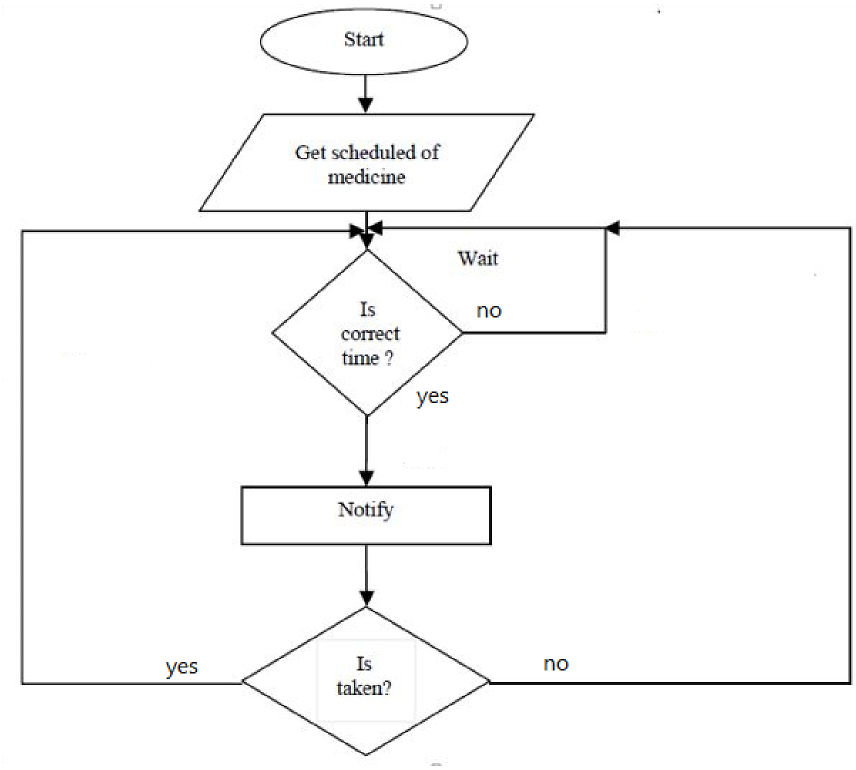


Figure 4.3: Application flow chart

4.6 Project Schedule

The time taken for the completion of the design, development and implementation of the iOS based medicine or pill reminder application, it took only 52 weeks as shown in the table below. The completion of the study took 52 weeks as depicted in the (Table 4.1). Figure 4.4 shows the Gantt chart for the study.

Table 4.1: Project schedule

Work done	**Duration**
Project Feasibility Studies	16 Weeks
Design and Development	22 Weeks
Program Testing	1 Week
Implementation	1 Week
Project Write up	8 Weeks
Write up corrections	1 Week
Total	**52 weeks**

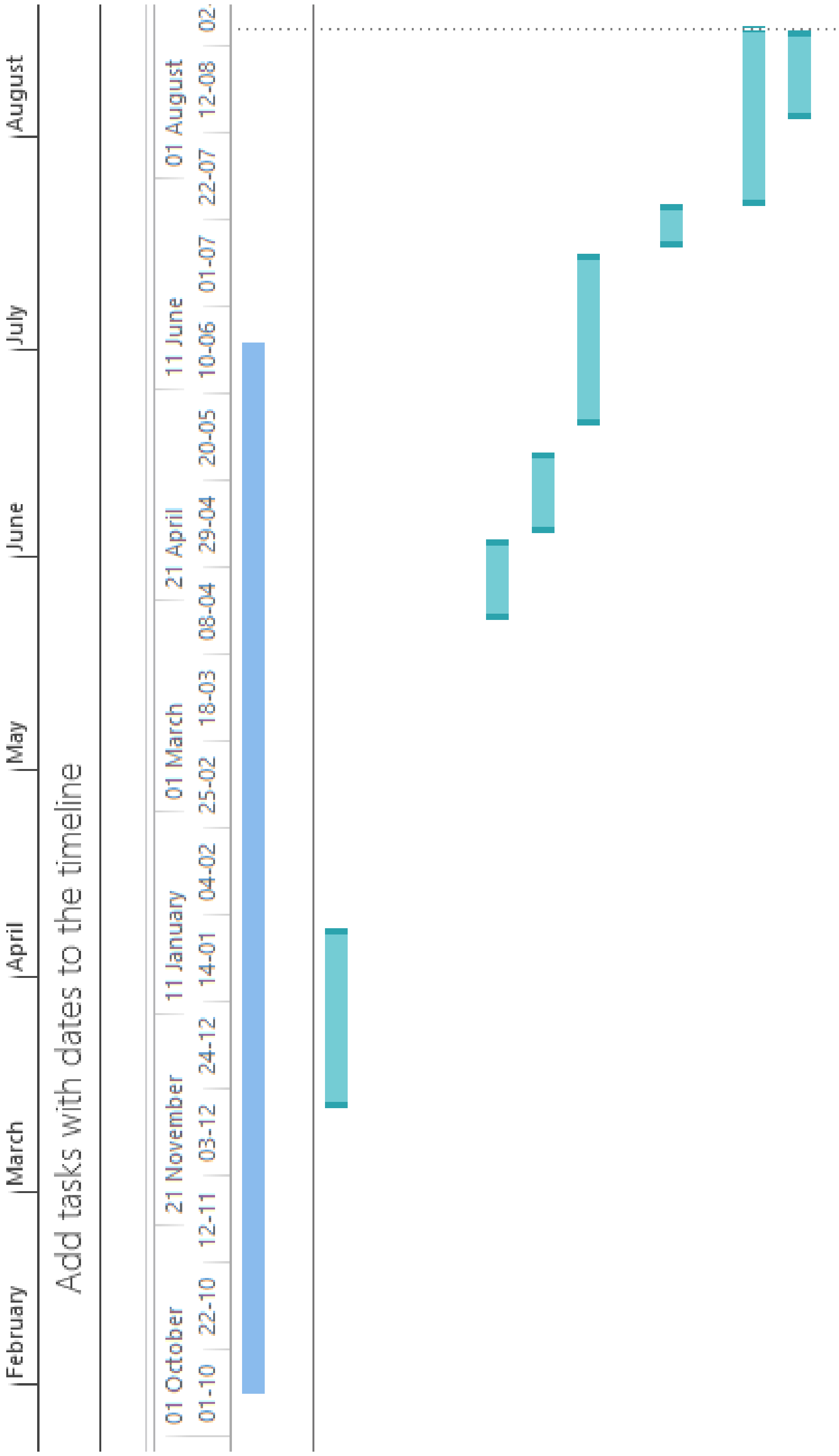


Figure 4.4: Gantt chart of the study

CHAPTER 5

SYSTEM IMPLEMENTATION

This section of the thesis discuses about how the designed iOS based medical reminder application could implement step by step.

5.1 Application Description

The mobile application comes in one of the most widely used mobile platforms namely the iOS. This application is basically a medication or pill reminder, the application makes a reminder to a patient about his routinely administered drug, the application works by making a booze or an alarm on the smartphone till the patient makes an acknowledgement. The application helps to keep medication intake on track and on time with an excellent medication intake reminder. In addition, the application allows a patient to communicate with his Doctor anytime, anywhere thus allowing the Doctor to monitor the patient whether he is taking the prescribed drugs on time or not, the system sends an automated message to the Doctor if the patient refuses to take his medicine on time.

5.2 Application Installation

The installation starts by clicking on the generated iOS executable file which is has an extension of. ipa which consist of a series of instructions. After following through the installation process of the iOS framework, the installed medication reminder application creates a shortcut with an icon in the menu for an easy access. Clicking on the newly created icon executes the iOS medication reminder application for usage

5.3 Gaining Access

After a successful installation of the iOS based medication reminder application, gaining authorization to the application is the next step.

A user has two options to gain access, it’s either to login or create an account. An existing user could just enter his/her login credentials then pressing the login button which voila gives an access with the condition that the entered login credential is correct otherwise access is denied.

The second option of gaining access to the iOS based medication reminder application is by making login via the Facebook account which in turns helps a user to quickly gain access without the need to create a new account for this application. However, this option was added to the application so as to help users easily gain access at their disposal. Nevertheless, creating an account is quite easy. The process involves a user entering their email/phone-no as username and registering the password. The entered credentials could be used to gain access to the iOS based medication reminder application (Figure 5.1).

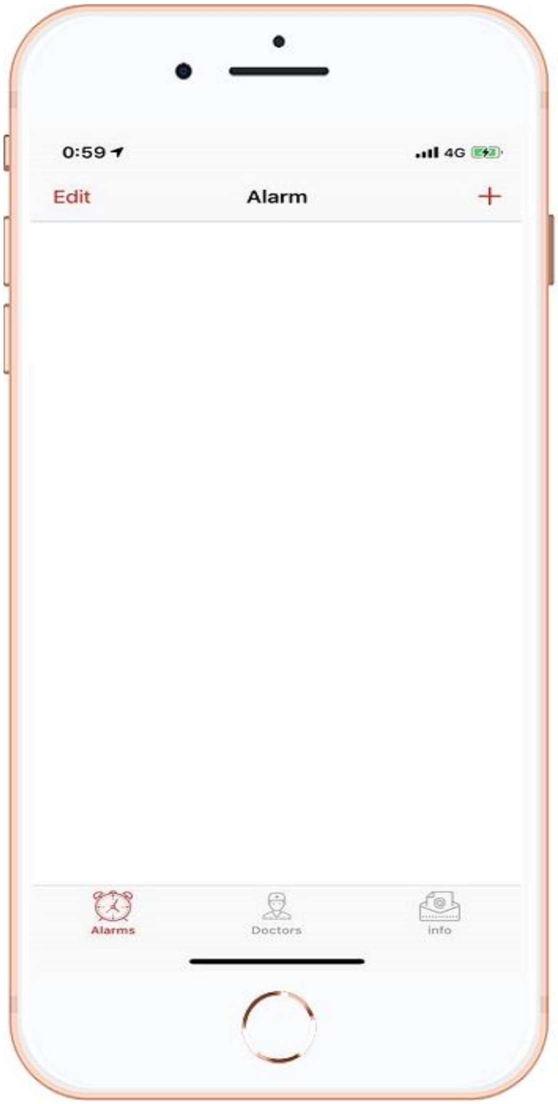
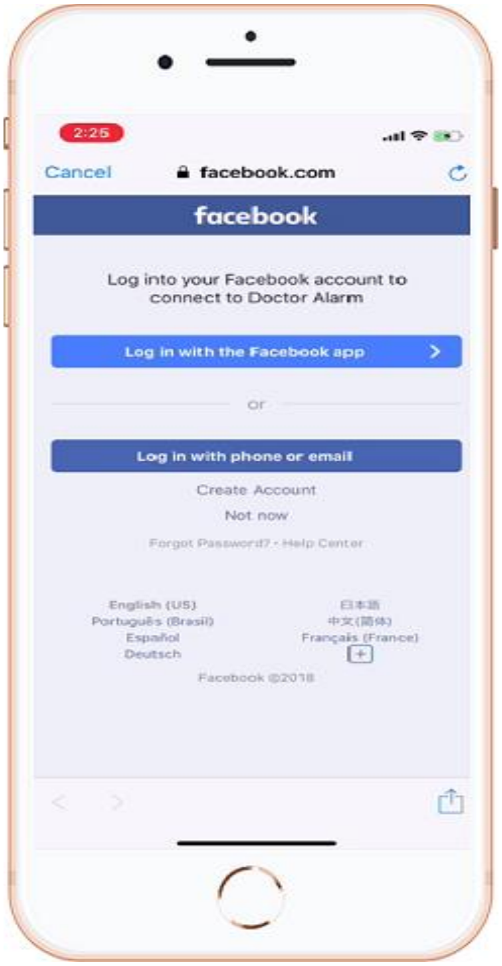


Figure 5.1: iOS medication reminder main menu (Beleid, 2016)



Log into your Facebook account to connect to Doctor Alarm

Figure 5.2: iOS Medication reminder login menu

The Figure 5.2 above shows the iOS based medication reminder application interface for making login via Facebook, login with phone or email, creating a new account to begin with and recovering password incase a user forgets his/her password.

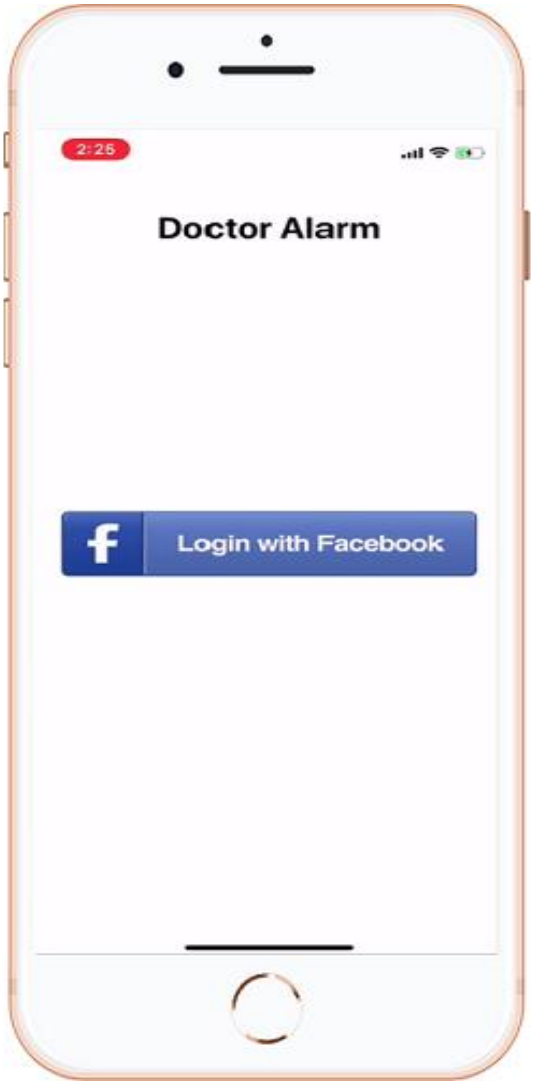
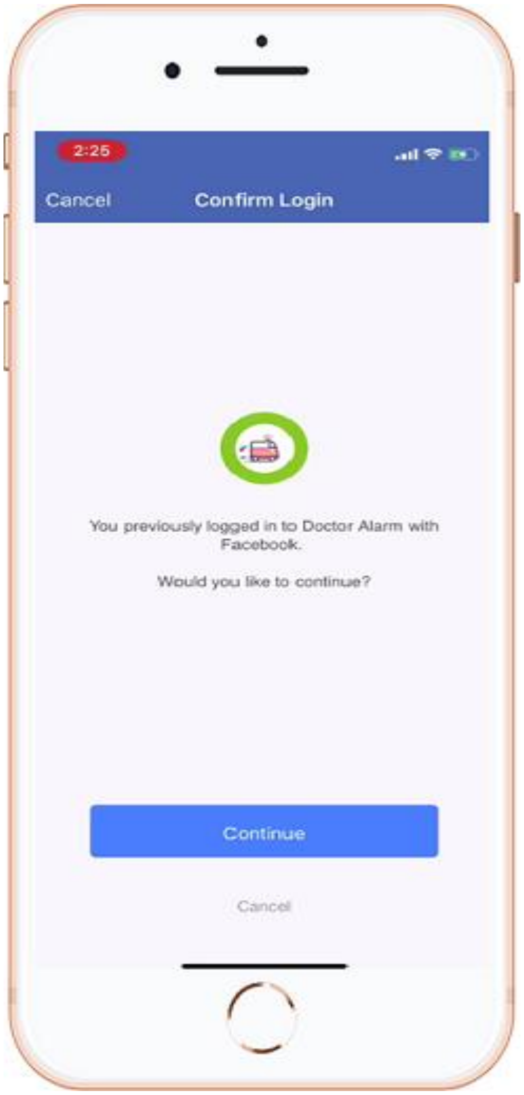


Figure 5.3: Login with facebook

The above Figure 5.3 shows the Facebook login prompt page displayed by the medication reminder app, then followed by a confirmation button to continue with the login process. The login is done by clicking on the ‘Login with Facebook’ button which will then proceed to login into the application by getting a confirmation to do so. After the confirmation, the application login using an existing Facebook account login credentials or in session. The application added this feature to make it easier and quicker for a patient to adopt this application anywhere, anytime.



2:25

Cancel Confirm Login

You previously logged in to Doctor Alarm with Facebook.

Would you like to continue?

Figure 5.4: Facebook confirmation

Clicking on the login with Facebook button requires a confirmation to make login using ones Facebook account for security reasons. Access will be granted if continue button is pressed otherwise the cancel button denies accessibility (Figure5.4).

5.4 Main Menu

After a successful login either with Facebook Login or Email Login, the medication application displays its main menu. The medication reminder application consists of three main tabs to work with at the bottom most of the application namely, ‘Alarm’, ‘Doctors’ and ‘info’. While the top most tabs consist of an ‘Edit’, ‘Alarm’, ‘+’ with features to edit, view and add a medicine (Figure 5.5).

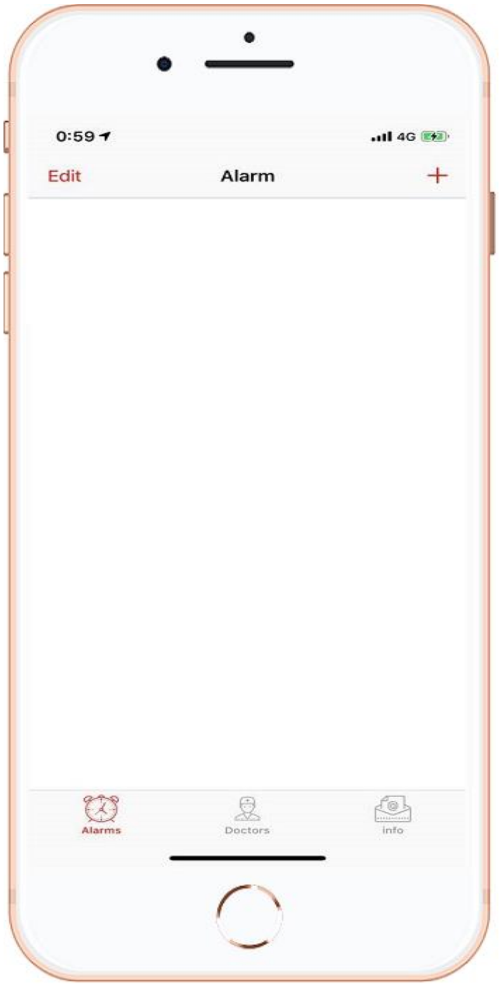
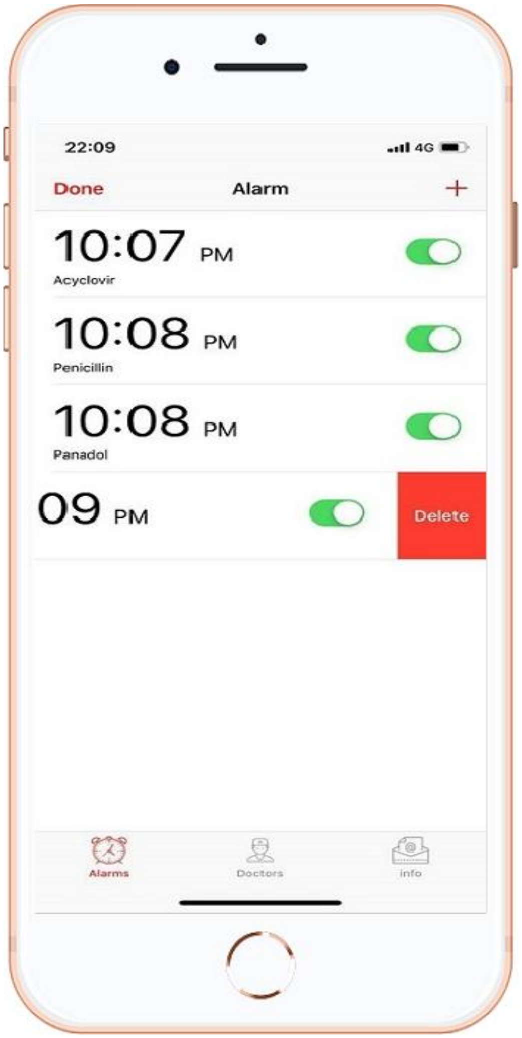


Figure 5.5: Main menu



10:07 PM
Acyclovir

10:08 PM
Penicillin

10:08 PM
Panadol

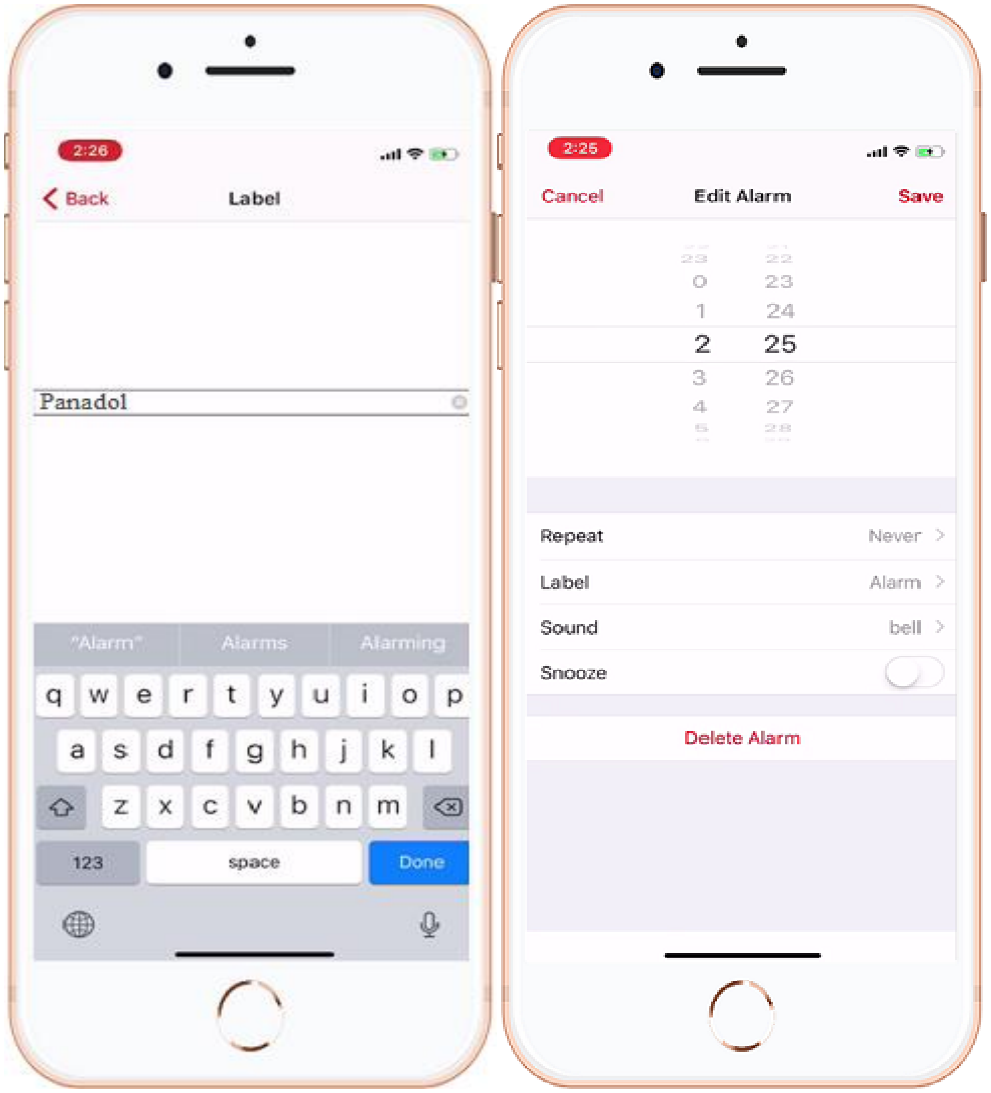
09 PM

Figure 5.6: View of medicine alarm

The Figure 5.6 above shows how medication reminder application interface is when a user has entered a medicine. Panadol, Penicillin and Acyclovir medicines were entered. And the program shows the medicines in stack, one after the other with their timing and options to delete, edit and deactivate the alarm.

5.5 Create Medication Alarm

Creating an alarm for medication (e.g. Panadol) intake is quite simple with this application, to create an alarm just press the ‘+’ button at the right top most of the tabs. Then a page opens and prompt one to enter some attributes of the Alarm, such as ‘Label’, ‘Time’, ‘Repeat’, ‘Sound’, ‘Snooze’ which are all required. After entering this information, the Alarm will automatically be created (Figure5.7).



Label: Panadol

Repeat	Never
Label	Alarm
Sound	bell
Snooze	○

Figure 5.7: Create alarm

5.6 Delete Medication Alarm

The application allows it users to be able to discard a medication alarm when is no longer needed. To delete a medication alarm, there is need to click to the specific list of medication alarm that is needed to be deleted, then a red button appears which if clicked will delete the specified medication alarm from the list. The following Figure 5.8 depicts the deletion scenario.

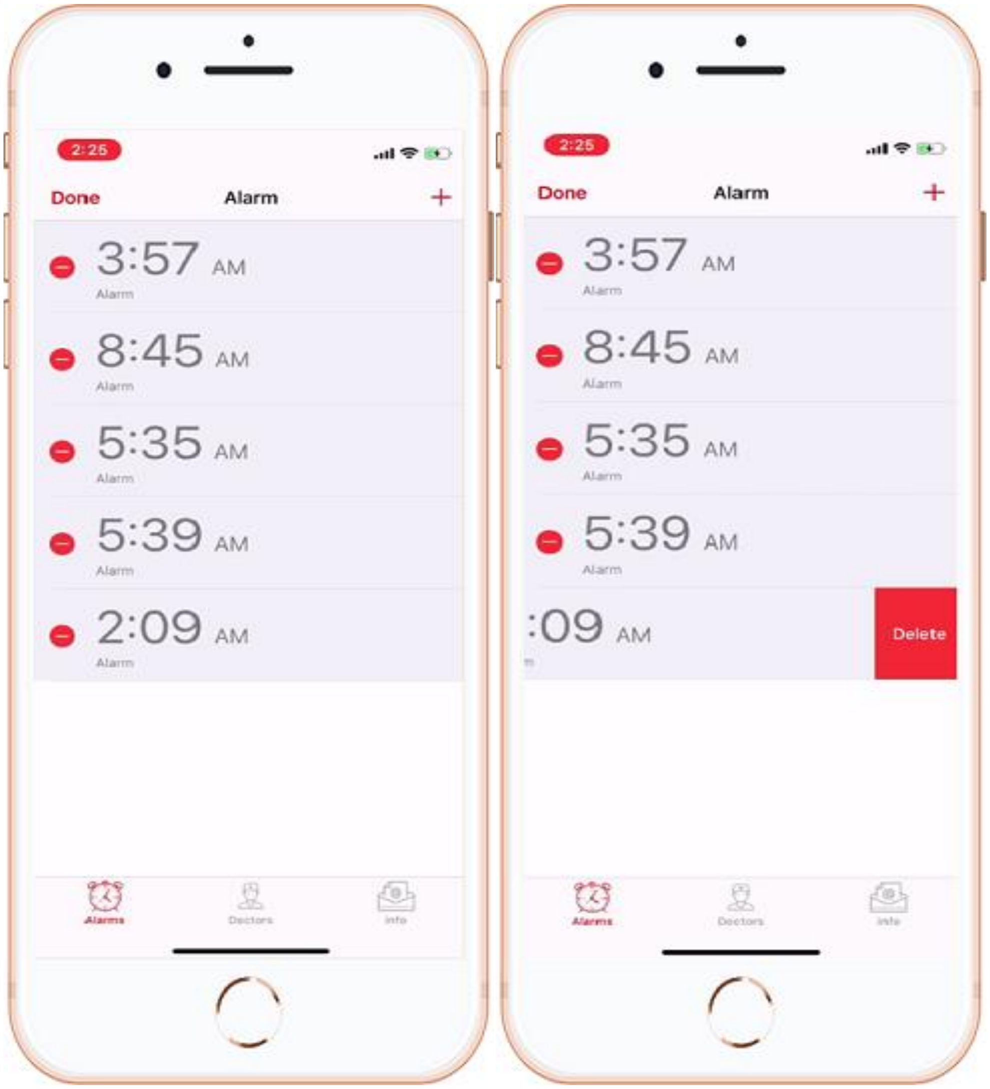
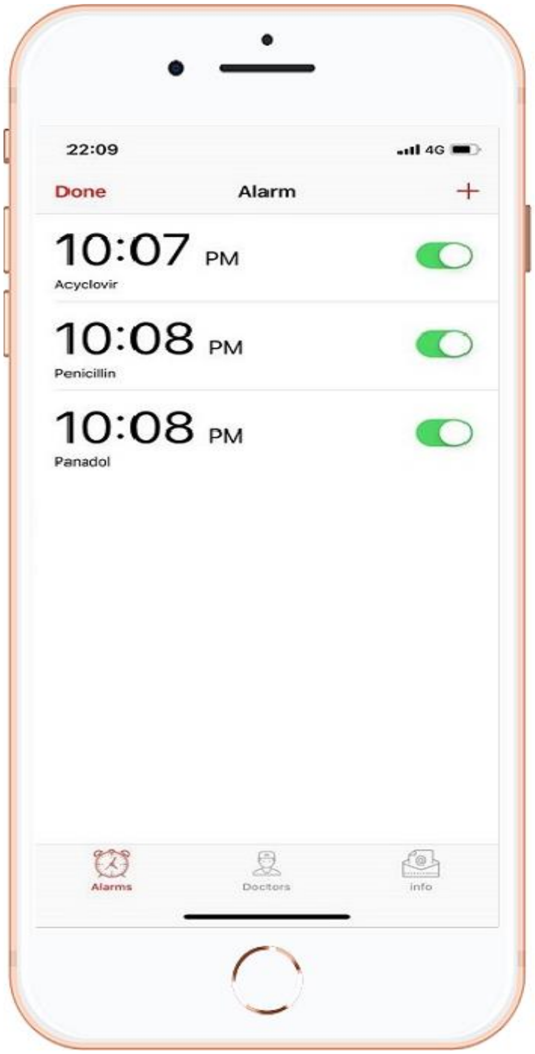


Figure 5.8: Delete alarm

5.7 The Alarm Reminder

The medication reminder application works by reminding patient to take their pills at the prescribed time by making an alarming sound. However, a patient can choose to mute this alarm, or change the type of sound to play to remind him of taking his medicine in due time. The Figure 5.9 depicts how the application reminds patient to take their drugs in prescribed dosages and due time.



10:07 PM
Acyclovir

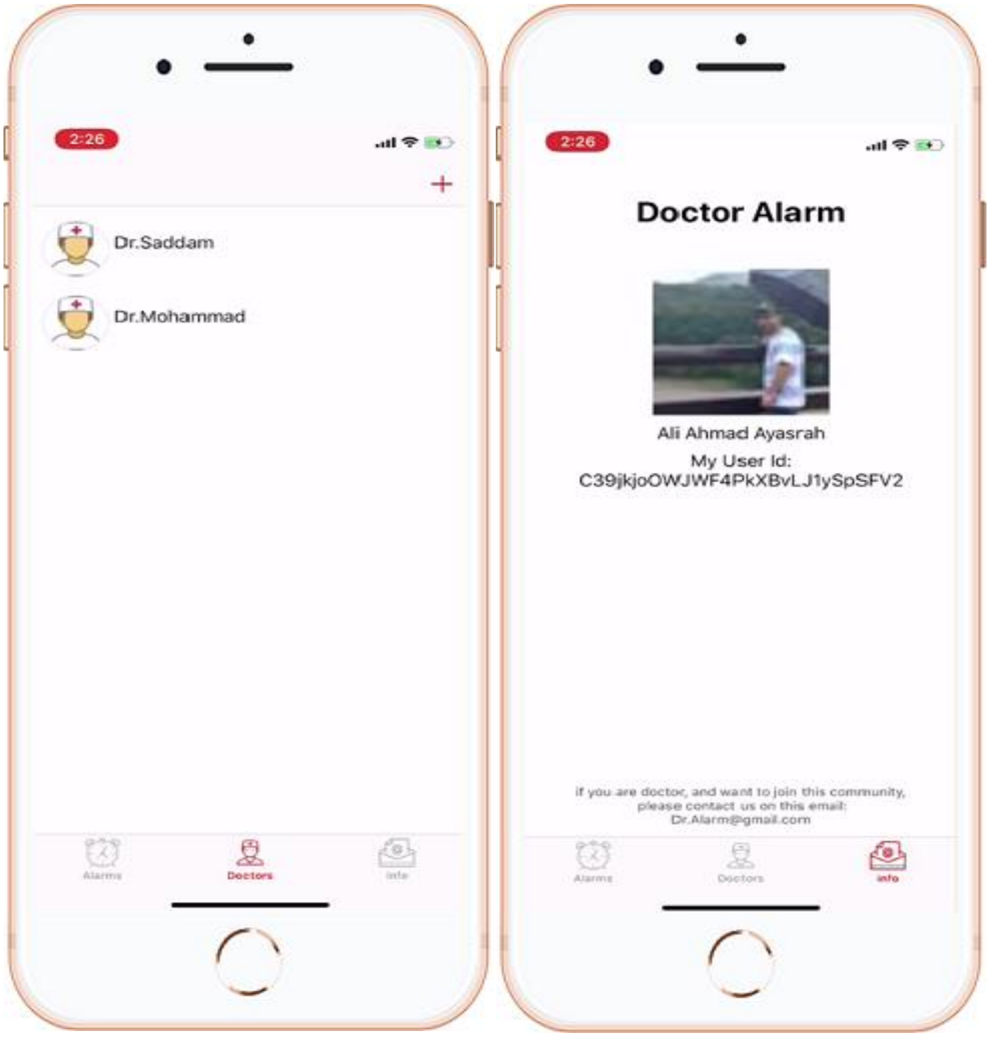
10:08 PM
Penicillin

10:08 PM
Panadol

Figure 5.9: Snoozing

5.8 Doctor page

One of the novelty of this application is the inclusion of a feature that lets the Doctor to interact and monitor the patient progress virtually on the application, this will bolster the interaction between a patient and the Doctor. And also could serve as a way Doctor or patient could quickly communicate with each other (Figure 5.10).



Dr.Saddam

Dr.Mohammad

Doctor Alarm

Ali Ahmad Ayasrah

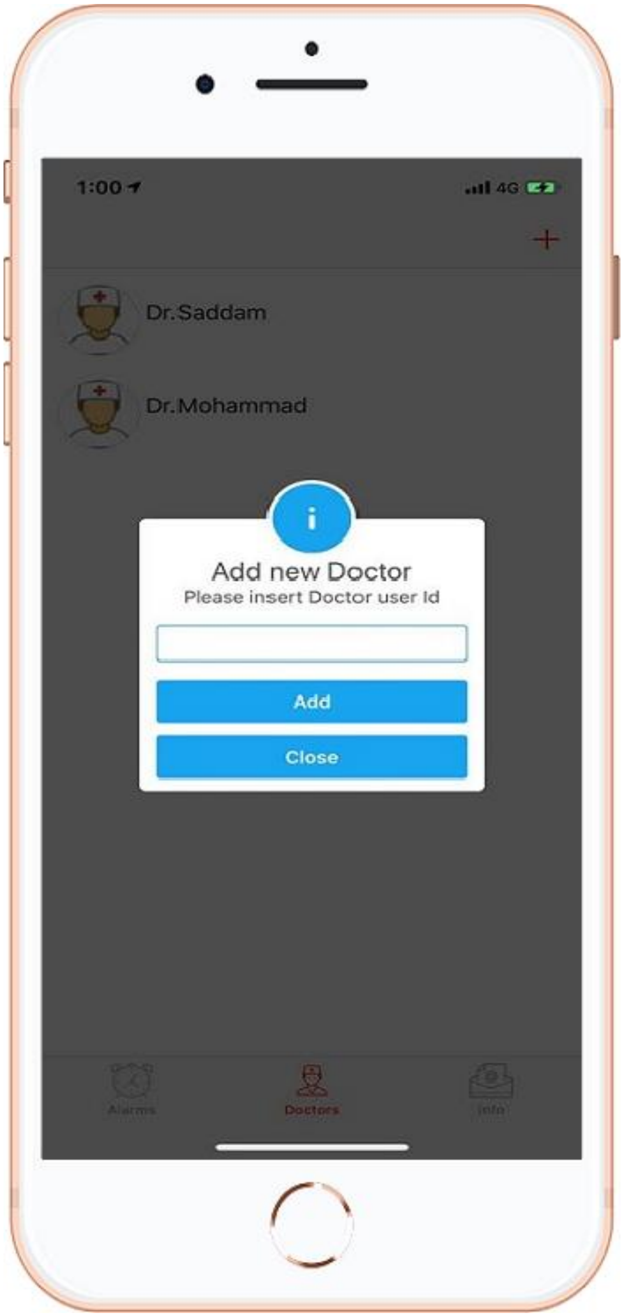
My User Id:
C39jkjoOWJWF4PkXBvLJ1ySpSFV2

if you are doctor, and want to join this community,
please contact us on this email:
Dr.Alarm@gmail.com

Figure 5.10: Creating a doctor page

5.9 Create doctor account

The Doctor page is created by clicking on the Doctor tab at the bottom most of the application user interface. Then a page pops up for the Doctor to input the required parameters. The Figure 5.11 below depicts the process.



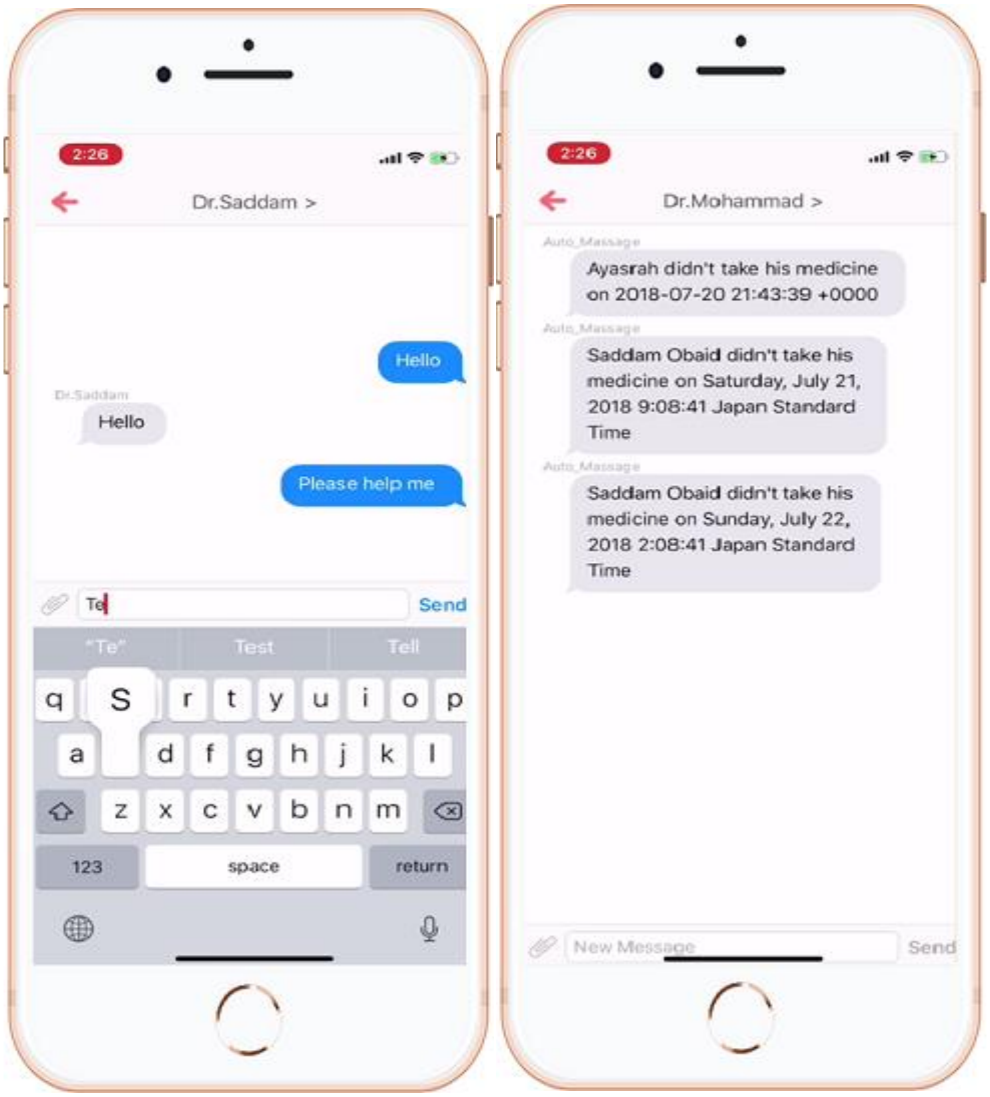
Add new Doctor

Please insert Doctor user Id

Figure 5.11: Doctor account

5.10 Chat and Status Feedback

The application includes a feature that lets communication between the two parties possible via text chatting and also the Doctor or physician could quickly get a feedback details of his patient status co, these allows the Doctor to monitor the patient virtually. Status such as whether a patient has taken the prescribed drugs or not, the timing of the drug intake etc. The application does this virtually and quickly reports to the Doctor of the current situation via a message reporting (Figure 5.12).



Dr.Saddam >

Hello

Dr.Saddam

Hello

Please help me

Te

Dr.Mohammad >

Auto_Massage

Ayasrah didn't take his medicine on 2018-07-20 21:43:39 +0000

Auto_Massage

Saddam Obaid didn't take his medicine on Saturday, July 21, 2018 9:08:41 Japan Standard Time

Auto_Massage

Saddam Obaid didn't take his medicine on Sunday, July 22, 2018 2:08:41 Japan Standard Time

Figure 5.12: Chatting and status

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

This chapter finalizes the design and implementation of an iOS based application for medication or pill reminder while providing some recommendation on how to improve the mobile based application.

6.1 Conclusion

Patient finds it very difficult to keep track of their medication especially if it involves intake of pills on daily basis due to several reasons such as heavy work load, forgetfulness and alterations in day-to-day behavior can also have a significant result on whether patients remember to take their prescribed medications, for this reasons the project aimed at designing and developing a mobile based application which runs on the second most popular mobile platform which is the iOS that will assist in keeping track of patients prescribed medication or pills. The iOS based medical reminder application allows it users to create a medication profile and also assign time to the profile, the created medication profile reminds the patient on their medication intake. The program allows for the creation of a lots of medication profiles with different timeframe attached to each profile. However, the medication profile can be edited or deleted as required by the patient. Furthermore, the patient current status could be tracked by the Doctor that prescribed the medicine and the both parties could also chat with one another. The iOS based medical reminder application was designed using the x-code as the IDE and the iOS SDK while adopting the SWIFT as the programming language. This implies that the application is platform dependent meaning the application can only run on the iOS based devices.

The medication reminder application can impact positively on the life of the patient as it will help the patient in keeping track of their daily intake of pills as remembering the intake of the prescribed medication can be a matter of life and death. Furthermore, the application could be a medium of communication between the physician and the patient and also help the physician monitors the intake his patient prescribed pills.

Finally, the novelty of this project is that the application has a Doctor interface whereby he or she could be able to virtually interact with their patient through chat and also prescribe medicines to patients from their own ends. The system however will assist the Doctor's in keeping track of patients prescribed medication. While the app provides an array of options to customizing the booze ringing tone, turn off the notifications and could see missed dose which is absent in some application designs.

6.2 Recommendations

On the design and development of the medication reminder, the application was designed and developed specifically to run on only the iOS based devices. This shows that the application is still platform dependent, therefore there is need to make the application run across various other smartphones operating system platforms such as the Windows mobile, Symbian OS, and Android OS etc.

Secondly, there is need to improve the application interface so that it can be more user friendly and also there is need for the application to have a cloud based backup incase a patient or physician device got missing or faulty, so as to enable them recover all their data.

Additionally, the application quality should be enhanced and be in conformance with the ISO 9126 quality model. Finally, more functionalities should be introduced such as which a Doctor can prescribe medicines to patients from their own end.

REFERENCES

Ameta, D., Mudaliar, K., & Patel, P. (2015). Medication reminder and healthcare-an android application. *International Journal of Managing Public Sector Information and Communication Technologies, 6*(2), 254-267.

Ammouri, S., & Bilodeau, G. A. (2008). Face and hands detection and tracking applied to the monitoring of medication intake. *In the Proceeding of the Computer and Robot Vision, 2008. Canadian Conference on, 2*(11), 147-154.

Attri, J. P., Khetarpal, R., Chatrath, V., & Kaur, J. (2016). Concerns about usage of smartphones in operating room and critical care scenario. *Saudi Journal of Anaesthesia, 10*(1), 87-98.

Banerjee, S. (2009). The use of antipsychotic medication for people with dementia: time for action. *A report for the Minister of State for Care Services: United Kingdom Department of Health, 12*(13), 448-454.

Batz, D., Batz, M., da Vitoria Lobo, N., & Shah, M. (2005). A computer vision system for monitoring medication intake. *In the Proceeding of the Computer and Robot Vision, 2nd Canadian Conference on IEEE, 2*(9), 362-369.

Becker, E., Metsis, V., Arora, R., Vinjumur, J.K., Xu, Y., & Makedon, F. (2009). Smart drawer: RFID-based smart medicine drawer for assistive environments. *Proceeding of Pervasive Technologies Related to Assistive Environments, 2*(9), 1-8.

Beleid, M. H. (2016). A mobile phone based medication reminder system using cloud computing. *Master Thesis, Near East University, Department of Computer Information System,* Lefkoşa, Cyprus.

Chen, B. X. (2011). Always on how the iPhone unlocked the anything-anytime-anywhere future. *Journal of Computing Sciences in Colleges, 2*(9), 17-29.

Darcey, L., & Conder, S. (2012). Android essentials on interface development. *Android Wireless Application Development Center, 2*(4), 77-89.

Grønli, T. M. (2012). Cloud computing and context-awareness: A study of the adapted user experience. *Doctoral dissertation, Brunel University, School of Information Systems, Computing and Mathematics,* Darussalam, Brunei.

Groves, P., Kayyali, B., Knott, D., & Van Kuiken, S. (2013). The big data revolution in healthcare. *McKinsey Quarterly*, 2(3), 112-121.

Ho, P. M., Bryson, C. L., & Rumsfeld, J. S. (2009). Medication adherence: its importance in cardiovascular outcomes. *Medical Circulation, 119*(23), 3028-3035.

Huang, S., Chang, H., Jhu, Y., & Chen, G. (2014). Design and implementation of intelligent pill box. *Journal of Computer Science 2,* (1), 235-236.

Hughes, C. M. (2004). Medication non-adherence in the elderly. *Drugs & Aging, 21*(12), 793-811.

Ilkko, L., & Karppinen, J. (2009). UbiPILL a medicine dose controller of ubiquitous home environment. *Third International Conference Mobile Ubiquitous Computer System Service Technology, 1*(2), 329-333.

Iversen, J., & Eierman, M. (2014). Learning mobile app development: a hands-on guide to building apps with iOS and android. *Journal of Computer Science, 11*(3), 2024-2035.

Jacob, B., & David, T. (2018). Android studio platform architecture. Retrieved May 26, 2018 from https://developer.android.com/guide/platform/

Kulathumani, H. A. (2015). MSDN mobile applications development: a case study of interface design and development. Retrieved February 22, 2018, from http://mobiledevelopmentsrilanka.com/MobileAppDevelopment/msdnmobile-applications-development

Lam, C. Y. (2012). A case study on the use of developmental evaluation for innovating: Navigating uncertainty and unpacking complexity. *Master Thesis, School of Computing, Queen's University,* Ontario, Canada.

Leonard, T. C., Richard, H., Thaler, C. R., & Sunstein, N. (2008). Improving decisions about health, wealth, and happiness. *Proceeding of Pervasive Technologies Related to Physical Health, 1*(3), 11-18.

Levis, P., & Culler, D. (2002). Maté: A tiny virtual machine for sensor networks. *In ACM Sigplan Notices*, *37*(10), 85-95.

Majchrzak, T. A., Dageförde, J. C., Ernsting, J., Rieger, C., & Reischmann, T. (2017). How cross-platform technology can facilitate easier creation of business apps. *In Apps Management and E-Commerce Transactions in Real-Time, 3*(12), 104-140.

Mamdouhi, H., Khatun, S., & Zarrin, J. (2009). Bluetooth wireless monitoring, managing and control for inter vehicle in vehicular ad-hoc networks. *Journal of Computer Science, 5*(12), 922-929.

Medvidovic, N., & Taylor, R. N. (2000). A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering*, *26*(1), 70-93.

Mohammed, H. B., Ibrahim, D., & Cavus, N. (2018). Mobile device based smart medication reminder for older people with disabilities. *Quality & Quantity, 2*(1) 1-14.

Nosrati, M., Karimi, R., & Hasanvand, H. A. (2012). Mobile computing: principles, devices and operating systems. *World Applied Programming*, *2*(7), 399-408.

Ohayon, M. M. (2002). Epidemiology of insomnia: what we know and what we still need to learn. *Sleep Medicine Reviews, 6*(2), 97-111.

Onarlioglu, K., Bilge, L., Lanzi, A., Balzarotti, D., & Kirda, E. (2010). G-Free: defeating return-oriented programming through gadget-less binaries. *In Proceedings of the 26th Annual Computer Security Applications Conference, 2*(11), 49-58.

Parida, M., Yang, H. C., Jheng, S. W., & Kuo, C. J. (2012). Application of RFID technology for in house drug management system. *15th International Conference Network Based on Information Systems, 11*(2), 577-581.

Park, K. H., & Lim, S. H. (2012). Construction of a medication reminder synchronization system based on data synchronization. *International Journal of Bio-Science and Bio-Technology, 4,* (4), 1-10.

Prasad, B. (2013). Social media, health care, and social networking. *Gastrointestinal Endocrines, 77*(21), 492-495.

Quindlen, R. (2000). Confessions of a venture capitalist: inside the high-stakes world of start-up financing. *Journal for Grand Central Publishing, 7*(25), 585-591.

Ravel, B., & Newville, M. (2005). Athena, artemis, hephaestus: data analysis for x-ray absorption spectroscopy using IFEFFIT. *Journal of Synchrotron Radiation, 12*(4), 537-541.

Riehemann, K., Schneider, S. W., Luger, T. A., Godin, B., Ferrari, M., & Fuchs, H. (2009). Nanomedicine-challenge and perspectives. *Angewandte Chemie International Edition, 48*(5), 872-897.

Riesenfeld, R. F., Haimes, R., & Cohen, E. (2015). Initiating a CAD renaissance: Multidisciplinary analysis driven design: Framework for a new generation of advanced computational design, engineering and manufacturing environments. *Computer Methods in Applied Mechanics and Engineering, 2*(84), 1054-1072.

Rogers, M. (2009). It's for you: an iPhone development primer for the busy college professor. *Journal of Computing Sciences in Colleges, 25*(1), 94-101.

Ryan, W. (2015). Extreme Tech: Google plans to remove Oracle's Java APIs from Android. Retrieved May 26, 2018, from https://www.extremetech.com/mobile/220136-google-plans-to-removeoracles-java-apis-from-android-n

Santo, K., Chow, C. K., Thiagalingam, A., Rogers, K., Chalmers, J., & Redfern, J. (2017). Medication reminder apps to improve medication adherence in coronary heart disease study: a randomized controlled trial protocol. *BMJ Open, 7*(10), 175-190.

Sawand, A., Djahel, S., Zhang, Z., & Naït-Abdesselam, F. (2014). Multidisciplinary approaches to achieving efficient and trustworthy eHealth monitoring systems. *In the Proceeding of the Communications in China, IEEE/CIC International Conference on, 2*(5), 187-192.

Slagle, J.M., Gordon, J.S., Harris, C.E., Davison, C.L., Culpepper, D.K., Scott P. & Johnson, K.B. (2011). Designing a next generation system for child-centered medication management. *Journal of Biomedical Informatics, 43*(5), 27-31.

Smith, G. S. (2013). Straight to the top: CIO leadership in a mobile, social, and cloud-based world. *Journal of Computer Science 3*, (21), 2235-2236.

Statista, (2018). Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018. Retrieved February 22, 2018, from https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

Taylor, I., Shields, M., & Wang, I. (2003). Distributed p2p computing within triana: A galaxy visualization test case. *In Parallel and Distributed Processing Symposium, 19*(4), 8-19.

Techotopia (2016). An Overview of the Android Architecture (Android Studio). Retrieved May26,2018,from,https://www.techotopia.com/index.php/an_overview_of_the_android_architecture_android_studio_java_apps.

Vaidya, A. H., & Naik, S. (2013). Comprehensive study and technical overview of application development in I0OS, android and window phone. *8th International Journal of Computer Applications*, *64*(19), 121-143.

Vijayan, V. (2012). Android forensic capability and evaluation of extraction tools. *Master's Thesis, Edinburgh Napier University*, Edinburgh, Scotland.

Xinran, L., Baisong, L., Anqi, C., Hui, L., & Zhihong, T. (2016). Current cybersecurity situation and emergency response of cybersecurity. *Strategic Study of Chinese Academy of Engineering, 18*(6), 83-88.

Zanjal, S. V., & Talmale, G. R. (2016). Medicine reminder and monitoring system for secure health using IOT. *Procedia Computer Science*, *78*(5), 471-476.

Zao, J. K., Wang, M. Y., Tsai, P., & Liu, J. W. (2010). Smart phone based medicine in-take scheduler, reminder and monitor. *In the Proceeding of the e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, *23*(8), 162-168.

APPENDIX
SOURCE CODE

```
import UIKit
import Foundation
import MediaPlayer

class AlarmAddEditViewController: UIViewController, UITableViewDelegate,
UITableViewDataSource{

    @IBOutlet weak var datePicker: UIDatePicker!
    @IBOutlet weak var tableView: UITableView!

    var alarmScheduler: AlarmSchedulerDelegate = Scheduler()
    var alarmModel: Alarms = Alarms()
    var segueInfo: SegueInfo!
    var snoozeEnabled: Bool = false
    var enabled: Bool!

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    override func viewWillAppear(_ animated: Bool) {
        alarmModel=Alarms()
        tableView.reloadData()
        snoozeEnabled = segueInfo.snoozeEnabled
        super.viewWillAppear(animated)
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }

    @IBAction func saveEditAlarm(_ sender: AnyObject) {
        let date = Scheduler.correctSecondComponent(date: datePicker.date)
        let index = segueInfo.curCellIndex
        var tempAlarm = Alarm()
        tempAlarm.date = date
        tempAlarm.label = segueInfo.label
        tempAlarm.enabled = true
        tempAlarm.mediaLabel = segueInfo.mediaLabel
        tempAlarm.mediaID = segueInfo.mediaID
        tempAlarm.snoozeEnabled = snoozeEnabled
        tempAlarm.repeatWeekdays = segueInfo.repeatWeekdays
        tempAlarm.uuid = UUID().uuidString
        tempAlarm.onSnooze = false
        if segueInfo.isEditMode {
            alarmModel.alarms[index] = tempAlarm
        }
        else {
            alarmModel.alarms.append(tempAlarm)
        }
        self.performSegue(withIdentifier: Id.saveSegueIdentifier, sender: self)
    }



    func numberOfSections(in tableView: UITableView) -> Int {
        // Return the number of sections.
        if segueInfo.isEditMode {
            return 2
        }
        else {
            return 1
        }
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        if section == 0 {
            return 4
        }
        else {
            return 1
        }
    }
```

```
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {

        var cell = tableView.dequeueReusableCell(withIdentifier: Id.settingIdentifier)
        if(cell == nil) {
        cell = UITableViewCell(style: UITableViewCellStyle.value1, reuseIdentifier: Id.settingIdentifier)
        }
        if indexPath.section == 0 {

            if indexPath.row == 0 {

                cell!.textLabel!.text = "Repeat"
                cell!.detailTextLabel!.text = WeekdaysViewController.repeatText(weekdays:
segueInfo.repeatWeekdays)
                cell!.accessoryType = UITableViewCellAccessoryType.disclosureIndicator
            }
            else if indexPath.row == 1 {
                cell!.textLabel!.text = "Label"
                cell!.detailTextLabel!.text = segueInfo.label
                cell!.accessoryType = UITableViewCellAccessoryType.disclosureIndicator
            }
            else if indexPath.row == 2 {
                cell!.textLabel!.text = "Sound"
                cell!.detailTextLabel!.text = segueInfo.mediaLabel
                cell!.accessoryType = UITableViewCellAccessoryType.disclosureIndicator
            }
            else if indexPath.row == 3 {
```

```
        cell!.textLabel!.text = "Snooze"
        let sw = UISwitch(frame: CGRect())
        sw.addTarget(self, action:
#selector(AlarmAddEditViewController.snoozeSwitchTapped(_:)), for:
UIControlEvents.touchUpInside)

        if snoozeEnabled {
          sw.setOn(true, animated: false)
        }

        cell!.accessoryView = sw
      }
    }
    else if indexPath.section == 1 {
      cell = UITableViewCell(
        style: UITableViewCellStyle.default, reuseIdentifier: Id.settingIdentifier)
      cell!.textLabel!.text = "Delete Alarm"
      cell!.textLabel!.textAlignment = .center
      cell!.textLabel!.textColor = UIColor.red
    }

    return cell!
  }


  func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    let cell = tableView.cellForRow(at: indexPath)
    if indexPath.section == 0 {
        switch indexPath.row{
        case 0:
            performSegue(withIdentifier: Id.weekdaysSegueIdentifier, sender: self)
            cell?.setSelected(true, animated: false)
            cell?.setSelected(false, animated: false)
        case 1:
            performSegue(withIdentifier: Id.labelSegueIdentifier, sender: self)
            cell?.setSelected(true, animated: false)
            cell?.setSelected(false, animated: false)
        case 2:
            performSegue(withIdentifier: Id.soundSegueIdentifier, sender: self)
            cell?.setSelected(true, animated: false)
            cell?.setSelected(false, animated: false)
        default:
            break
        }
    }
    else if indexPath.section == 1 {
        //delete alarm
        alarmModel.alarms.remove(at: segueInfo.curCellIndex)
        performSegue(withIdentifier: Id.saveSegueIdentifier, sender: self)
    }

}
```

```
@IBAction func snoozeSwitchTapped (_ sender: UISwitch) {
    snoozeEnabled = sender.isOn
}


// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    // Get the new view controller using segue.destinationViewController.
    // Pass the selected object to the new view controller.
    if segue.identifier == Id.saveSegueIdentifier {
        let dist = segue.destination as! MainAlarmViewController
        let cells = dist.tableView.visibleCells
        for cell in cells {
            let sw = cell.accessoryView as! UISwitch
            if sw.tag > segueInfo.curCellIndex
            {
                sw.tag -= 1
            }
        }
        alarmScheduler.reSchedule()
    }
    else if segue.identifier == Id.soundSegueIdentifier {
        //TODO
        let dist = segue.destination as! MediaViewController
        dist.mediaID = segueInfo.mediaID
        dist.mediaLabel = segueInfo.mediaLabel
    }
    else if segue.identifier == Id.labelSegueIdentifier {
        let dist = segue.destination as! LabelEditViewController
        dist.label = segueInfo.label
    }
    else if segue.identifier == Id.weekdaysSegueIdentifier {
        let dist = segue.destination as! WeekdaysViewController
        dist.weekdays = segueInfo.repeatWeekdays
    }
}

@IBAction func unwindFromLabelEditView(_ segue: UIStoryboardSegue) {
    let src = segue.source as! LabelEditViewController
    segueInfo.label = src.label
}

@IBAction func unwindFromWeekdaysView(_ segue: UIStoryboardSegue) {
    let src = segue.source as! WeekdaysViewController
    segueInfo.repeatWeekdays = src.weekdays
}

@IBAction func unwindFromMediaView(_ segue: UIStoryboardSegue) {
    let src = segue.source as! MediaViewController
```

```
        segueInfo.mediaLabel = src.mediaLabel

        segueInfo.mediaID = src.media

}


}




import Foundation

import MediaPlayer


struct Alarm: PropertyReflectable {

    var date: Date = Date()

    var enabled: Bool = false

    var snoozeEnabled: Bool = false

    var repeatWeekdays: [Int] = []

    var uuid: String = ""

    var mediaID: String = ""

    var mediaLabel: String = "bell"

    var label: String = "Alarm"

    var onSnooze: Bool = false


    init(){}


    init(date:Date, enabled:Bool, snoozeEnabled:Bool, repeatWeekdays:[Int], uuid:String,
mediaID:String, mediaLabel:String, label:String, onSnooze: Bool){
        self.date = date
        self.enabled = enabled
        self.snoozeEnabled = snoozeEnabled
        self.repeatWeekdays = repeatWeekdays
        self.uuid = uuid
        self.mediaID = mediaID
        self.mediaLabel = mediaLabel
        self.label = label
        self.onSnooze = onSnooze
    }

    init(_ dict: PropertyReflectable.RepresentationType){
        date = dict["date"] as! Date
        enabled = dict["enabled"] as! Bool
        snoozeEnabled = dict["snoozeEnabled"] as! Bool
        repeatWeekdays = dict["repeatWeekdays"] as! [Int]
        uuid = dict["uuid"] as! String
        mediaID = dict["mediaID"] as! String
        mediaLabel = dict["mediaLabel"] as! String
        label = dict["label"] as! String
        onSnooze = dict["onSnooze"] as! Bool
    }

    static var propertyCount: Int = 9
}
```

```
extension Alarm {
    var formattedTime: String {
        let dateFormatter = DateFormatter()
        dateFormatter.dateFormat = "h:mm a"
        return dateFormatter.string(from: self.date)
    }
}

//This can be considered as a viewModel
class Alarms: Persistable {
    let ud: UserDefaults = UserDefaults.standard
    let persistKey: String = "myAlarmKey"
    var alarms: [Alarm] = [] {
        //observer, sync with UserDefaults
        didSet{
            persist()
        }
    }

    private func getAlarmsDictRepresentation()->[PropertyReflectable.RepresentationType] {
        return alarms.map {$0.propertyDictRepresentation}
    }

    init() {
        alarms = getAlarms()
    }
```

```
func persist() {
    ud.set(getAlarmsDictRepresentation(), forKey: persistKey)
    ud.synchronize()
}

func unpersist() {
    for key in ud.dictionaryRepresentation().keys {
        UserDefaults.standard.removeObject(forKey: key.description)
    }
}

var count: Int {
    return alarms.count
}

//helper, get all alarms from Userdefaults
private func getAlarms() -> [Alarm] {
    let array = UserDefaults.standard.array(forKey: persistKey)
    guard let alarmArray = array else{
        return [Alarm]()
    }
    if let dicts = alarmArray as? [PropertyReflectable.RepresentationType]{
        if dicts.first?.count == Alarm.propertyCount {
            return dicts.map{Alarm($0)}
        }
        }
        unpersist()
        return [Alarm]()
    }
}
```