National College of Ireland

BSc in Computing

2015/2016

Derek Gordon

12110370

X12110370@student.ncirl.ie

Appoint Remind

Technical Report

Table of Contents

Executive Summary

The main objective for this project is to develop a mobile application that will;

a) Allow users organise their medical appointments and have all related information in one location.

b) Allow users manually enter their various upcoming appointments into one location by specific appointment, date and time.

c) Act as a reminder to take medications i.e. to inform users with medical conditions to take their medication at a particular date/time.

Nominated users such as parents, guardians and carers can use this application to ensure appointments are managed effectively. The application can also be utilised by people who take medication on a regular basis. The notification reminder function can act as a prompt to take medication at specific times.

The application will have a calendar that will allow users manually enter their upcoming appointments. Another advantage of this application is the additional functionality of allowing users share their events with other nominated users.

The application will inform the user through SMS, email and push notifications thus reducing the possibility of missed appointments.

What will appeal to the User is, they have the option of customising the type of notification they wish to receive, along with the ability to determine how often they want to receive notifications.

Another appealing factor for users is their information can be accessed by them anywhere anytime; this is invaluable especially where multiple appointments are concerned.

1 Introduction

1.1 Background

The concept for this application came from both personal experience and, talking to other parents about the level of frustration caused by missing their children's hospital appointments. While the Irish health system issues letters regarding upcoming appointments, these letters are sent out months in advance of scheduled appointments.

Some hospitals provide a text messaging service to remind parents/guardians of upcoming appointment's but unfortunately not all. Only a miniscule number of hospitals provide this type of service.

A child suffering from a chronic medical condition requires a vast amount of medical care. This may result in multiple visits to hospitals /doctors in different locations. Parents/Guardians need to track all appointments therefore time management forms a greater part of a child's illness plan for Parent's.

In order to be confident that this type of Application would be utilised I conducted a survey, taking a sample of people from the following groups

- Working parents
- Carer's
- Work Colleagues, I included work colleagues without any knowledge of their personal status to ensure less bias in the survey results.

80% of the people surveyed said they would use this type of App, and the remaining 20% said they would use it sometimes. The survey also looked at frequency, and preferred method of notifications. As the results of the survey were analysed it became evident that a number of choices had to be built into the App. See detailed survey results in Appendix B.

1.2 Aims

The aim of this application is to provide parents/guardians a means by which they can maintain all appointment/medication information in one location. This application acts as a data store, the application will be stored using a cloud service, meaning that the data can be accessed from anywhere at any location providing there is an internet or phone connection. A facility within the application is to send notifications to the parents/guardians of upcoming medical appointments.

The results from research suggested that it would be beneficial to the user if the application could be shared within a domain. This means that two or more users can have access to a shared calendar where they can add, update and delete events as required.

The overall aim of this application is to allow parents/carers manage appointments effectively and efficiently. The demand is there for this type of application as most people's preferences are for online systems where data is easily retrievable.

Technologies

Visual Studio 2013. This is used to provide an IDE (Integrated Development Environment), this provides a suite of tools to help create the application. It is used for UI (User Interface) design, coding, testing, debugging, analysing code and deploying. On deployment; Windows Store will be used to allow users download the application. All coding and design will be completed within Visual Studio.

Microsoft Azure. This will be used when publishing the application. Azure provides a cloud platform with an easy to use functionality to connect to Visual Studio. Azure has built in SQL dB facility and built in notification facility which can be adapted when assigning push notifications to users. Visual Studio and Azure are both Microsoft products which means a seamless connection between the two.

SQLite. This is used as the database for the application.

Visual Paradigm Community Edition. This tool was used for the creation of the UML (Unified Modelling Language) diagrams.

1.3 Structure

Chapter 1: Gives the reader an overview of the application. It explains what the application does, the intended audience, the inspiration for creating the application, the achievement aims and the software tools used during development.

Chapters 2 to 8: Explains in detail the main functions and requirements within the application. This chapter details the application design, the application functions, the tools and patterns used. It also shows the DB structure and how each table links within a relational environment.

Chapters 9 to 12: Describes the requirements.

Chapter 13: Describes the Design and Architecture including the database tables

Chapter 14: Shows how the application was implemented with examples of code.

Chapter 15: Shows all of the testing done on the functionality of the application.

Chapter 16: Provides the reader with interesting screenshots.

Chapter 17: Customer testing reviews from possible end users.

Chapter 18: Describes how the application will be evaluated.

Chapter 19: Conclusions.

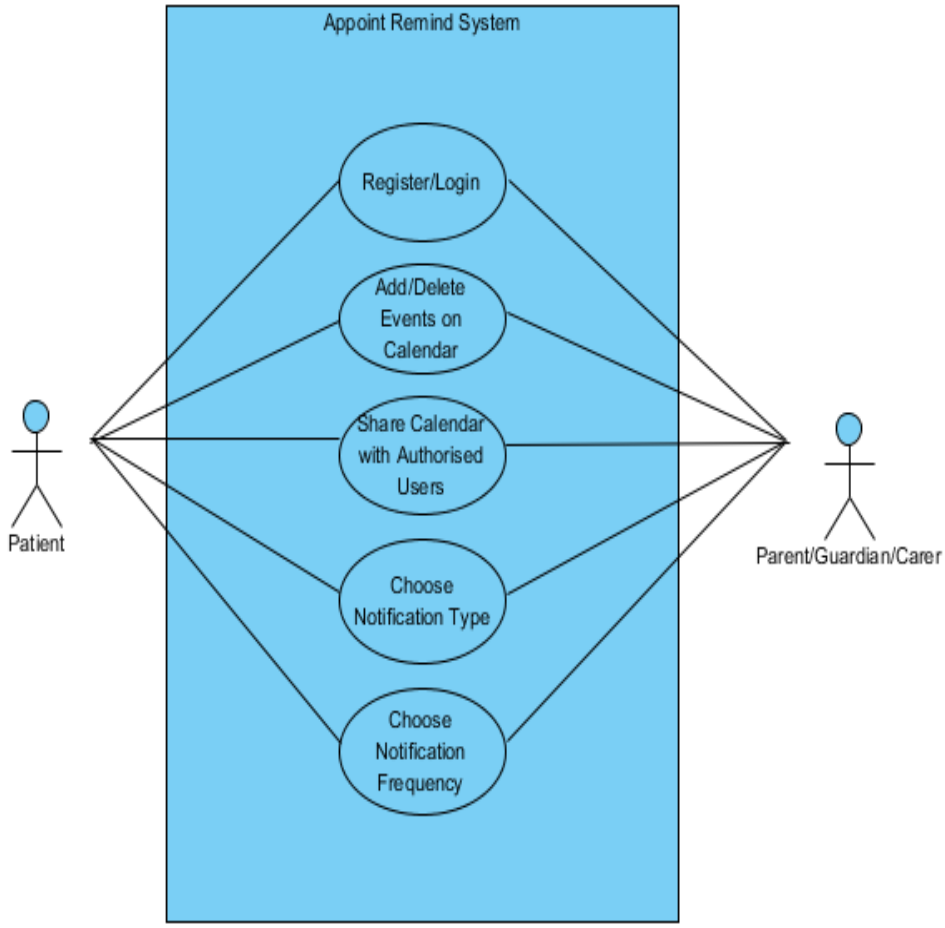
Chapter 20: Further Development.

Chapter 21: References.

Chapter 22: is the appendix. This contains the project proposal, monthly journals and screen shots of the survey completed during requirements gathering.

2 Use Case Requirements

2.1 Use Case Diagram



2.2 List of Actors

Below describes the list of actors shared throughout the processing of the Appoint Remind portal

Actor Name	Description
User	A person who has access to the application and can consult static data. This person has the following privileges: 1. Check status of all Events 2. Check Status of Event Notifications 3. Check Status of all Events
Admin	A person who has access to the application. The Admin holds all the same privileges as the User but can perform specific tasks such as: 1. Users management 2. Errors management 3. Static data management
MPNS	Microsoft Windows Phone Notification Service MPNS is a cloud service that pushes notifications to a device. There are different types of push notifications available within Microsoft. The 'Appoint Remind' applications uses three push notification types: ▪ SMS ▪ Email ▪ Push Notification (Toast)
WNS	Windows Notification Store Windows store allows users to download applications from a cloud server to their mobile device. All the features required with the application are installed on the user's device.

2.3 List of Use Cases

A use case is a discrete, stand-alone activity that an actor can perform to achieve some outcome of a value. A use case is a collection of related usage scenarios, and a scenario is a specific instance of a use case.

UC #	Use Case Name	Use Case Description	Use (Y/N)
UC001	Installation 'Appoint Remind'	Mandatory step: all users must have the application installed on their phone to connect to the application. A user must have a valid Microsoft Account in order to install the application.	Y
UC002	Create Account	Mandatory step: all users must create a valid account to connect to the application.	Y
UC003	Login	Mandatory step: all users must supply a user name and password to connect to the 'Appoint Remind' Application.	Y
UC004	Create Events	Users can Create New Events using this	Y
UC005	Event List	Users can view list of all events created under their account profile. A user can choose to Edit/Remove an event listed within this screen.	
UC006	Notification Frequency	Users can select the frequency of an event. Users can decide the time and date that a Notification is issued to their device.	Y
UC007	Notification Setup	PNS are used to push notifications to a users' device. A user can choose the type of Notifications they wish to receive choosing from three options: Text, Email, Toast	Y
UC008	Grant Privileges to Events	Users can share an Event calendar with other registered users of the application.	Y
UC009	Push Notification Sending	How the push notification will work	Y

3 UC001 – Install Appoint Remind

3.1 Description & Priority

This use case describes the process to install the Appoint Remind application onto a Windows Phone.

3.2 Actors

Primary Actors: WNS (Microsoft Windows Notification Store)

Secondary Actor: User/Admin

3.3 Precondition

A user has a valid Microsoft account so they can download the application using the Windows Store. Users much have an internet connection so to connect their device to Microsoft Windows Store.

3.4 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when a user chooses to install the 'Appoint Remind' Application using their Windows Store account.	UI001 – Install 'Appoint Remind'
2. User logs into the App List and signs into their Microsoft Account.	
3. System checks if user has a valid Microsoft Account.	AF001 – Unknown credentials
4. Systems send a request to the WNS; passing in the Users Microsoft Account details.	
5. WNS check if account is valid	

6. WNS sends answer with status	
7. Valid Account: System authorizes User to connect to the Windows Store.	AF002 – Valid Microsoft Windows Account
8. Invalid Account: System instructs user to setup a Microsoft Account so to connect to the Windows Store.	MSG01. – Please create Microsoft Account
9. Valid Microsoft Account: User searches for the Appoint Remind' application within Windows store	
10. WNS check if Appoint Remind is available within the Windows Store	
11. WNS sends answer with status	
12. Application found: User selects 'Install' to download the application onto their phone	MSG02. – Application Installed successfully
13. Application not found in listings	AF002 – Application not listed
14. Use case ends.	

3.5 Alternative Flows

3.5.1 UC001 – AF001 – Invalid Microsoft Account

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 8 of basic flow when WNS detects that the user does not have a valid Microsoft Account	
2. WNS does not allow user to use the Windows store to download the application.	MSG03. – Please create Microsoft Windows Account
3. Use case ends.	

3.5.2 UC001 – AF002 – Application not listed

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 9 of basic flow when user searches for the 'Appoint Remind' application using their Windows store account	AF002 – Application not listed
2. WNS check if the 'Appoint Remind' application is available for download	
3. WNS sends response to users phone	
4. WNS returns message to user 'No application found'	MSG04. – No Application found
5. Use case ends.	

3.6 Termination

The system moves from the Microsoft store page to launch the 'Appoint Remind' application.

3.7 Post condition

User has a valid Microsoft Account where they can download Microsoft application using Windows store.

4 UC002 – Create Account

4.1 Description & Priority

This functionality allows users to setup an account within the 'Appoint Remind' application. Once an account is setup; a user has full access to the application.

4.2 Actors

Primary Actors: User

4.3 Precondition

Every account holder must hold a valid account to use the 'Appoint Remind' application.

4.4 Activation

This use case starts when the user selects: 'Register'.

4.5 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case stars when the User activates the application from their phone	
2. User clicks on "Register" menu.	UI002 – Registration screen
3. System displays 'Register' screen so a user can enter their details (Name, Email, Phone, Password)	AF001 – Unknown credentials
4. User enters an Name, Email, Phone, Password	AF002 – System reads entry
5. Using the Email Address entered; the LDAP checks if there is an existing account.	
6. System sends answer with status	
7. No Account Holder: entry created/saved.	MSG05.– New Account Setup
8. Account Holder found: message notification sent to user.	MSG06.– Account already exists
9. End of use case.	

4.6 Alternative Flows

4.6.1 UC002 – AF001 – Account Exists

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 8 of the main flow when an account is found within the system.	
2. User chooses to log-in using the Log-In Screen.	
3. Using the log in details entered; the system checks if there is an existing account.	
4. System sends answer with status (Valid Account)	MSG07.– Log-In Successful
5. System sends answer with status (Invalid Account)	MSG08.– Invalid Account
6. End of use case.	

4.6.2 UC002 – AF002 – Password Retrieval

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 8 of the main flow when an account is found within the system.	
2. User chooses to retrieve log-in details.	
3. User enters email address to get email notification on account details	
4. System checks that the Email is a valid Email Address (characters '@', '.' found within the entry)	
5. Using the Email Address; the system checks if there is an existing account.	
6. System sends answer with status	

7. Registered User: Email sent to user with Account retrieval details	MSG09.– Account retrieval Email sent
8. Non Registered Account: Message appears on screen notifying user that no account exists	MSG010.– Account does not exists; please create an Account
9. End of use case.	

4.6.3 UC002 – AF003 – Mandatory information missing

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at one of followings: – step 4 of Basic Flow when “Log In” button is clicked with mandatory information not filled	
2. System displays an error message detailing missing information	MSG011.– Please complete all fields
3. Use case resumes to one of followings – step 4 of Basic Flow keeping filled information in related fields	

4.7 Termination

The system displays the ‘Register’ or ‘Events’ page when a user has logged in correctly. The Events page displays a list of Events applicable to the users account.

4.8 Post condition

The system goes into await state until the user selects ‘Log-In’ or ‘Register’.

5 UC003 – Log into Application

5.1 Description & Priority

This use case describes the process to log into the application. This has a high priority as the system needs to differentiate between users.

5.2 Actors

Primary Actors: Admin, User

5.3 Precondition

The application is installed on a user's Windows Phone and the phone has a valid internet connection.

5.4 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when login page is displayed.	UI003 - Login screen
2. User enters user name and password into the text fields.	
3. System checks if user details are defined with a role in the application.	AF001 – Unknown credentials
4. Systems send a request to the dB; passing in the User Name and Password.	
5. System checks if name and password is correct	
6. System sends answer with status	
7. System authorises User/Admin to connect and display application welcome page.	AF002 – User Name / Password combination invalid
8. Use case ends.	

5.5 Alternative Flows

5.5.1 UC003 – AF001 – Account Unknown

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 3 of basic flow when the user details entered are not found within the dB.	
2. System does not allow the connection and displays Error message within the 'Log-In' page.	MSG012. – Incorrect Log-In; Please try again
3. Use case ends.	

5.5.2 UC003 - AF002 – Invalid Account

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 7 of basic flow when the system receives a negative response for username/password combination.	
2. System does not allow the connection and displays error message.	MSG013.– Incorrect Log-In; Please try again
3. Use case ends.	

5.6 Termination

The system moves from the login page to the Events page of the Appoint Remind application.

5.7 Post condition

User has access to the screen and can choose to modify/create an event.

6 UC004 – Create Events

6.1 Description & Priority

This functionality allows account holders to add / remove / edit / delete events within their account profile. Once an event is setup; an account holder can setup notifications on each event.

6.2 Actors

Primary Actors: User

6.3 Precondition

Each user must be successfully logged into the application to create/modify events.

6.4 Activation

This use case starts when the user has successfully logged into the 'Appoint Remind' application with a valid account details.

6.5 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when a User is successfully logged into the 'Appoint Remind' application	UI004 – Create Events
2. Users are directed to the 'Event' Page.	
3. System displays an empty text box to manually add an event with Date and Time picker.	
4. User enters an Event; manually enters a Name and selects from a Date and Time pickers	
5. The system checks to see if there is an existing event for the name, date & time input by the user.	

6. System sends answer with status	
7. No records found: Entry created/saved.	MSG014.– New Event created successfully
8. All My Event page opens to screen	UI004 – Event List
9. Records found: message notification sent to user.	AF001 – Duplicate Event Date/Time MSG015.– Event (x) already exists
10. End of use case.	

6.6 Alternative Flows

6.6.1 UC004 – AF001 – Duplicate Event Date/Time

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 9 of the main flow when a user selects an Event Date/Time that already exists.	AF001 – Event Date/Time Exists
2. User amends the Event Date/Time.	
3. The system checks to see if the new Event Date/Time chosen does not exist.	
4. System sends answer with status	
5. Event Date/Time does not exist; entry saved	MSG016.– Event Saved
6. Event Date/Time exists; notification message sent to user	MSG017.– Event Date: (x); Event Time: (y) exists
7. End of use case.	

6.6.2 UC004 – AF002 – Duplicates Event

Steps	Alternative Flow / Data / Message / Business Rule
1. The flow starts at step 9 of the main flow when an event exists for the Date and Time selected	
2. Users chooses to create a duplicate Event for specified Date and Time	
3. System sends answer with status	MSG018.– Duplicated Event Saved
4. End of use case.	

6.6.3 UC004 – AF003 – Mandatory information missing

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at one of followings: – step 4 of AF001 – step 3 of AF002 when “Save” button is clicked with mandatory information not filled	
2. System displays an error message detailing missing information	MSG019.– Missing Event Details: (x)
3. Use case resumes to one of followings – step 2 of AF001 – step 2 of AF002 – step 2 of AF006 keeping filled information in related fields	

6.7 Termination

The Create Events screen allows users to Create Events.

6.8 Post condition

The system goes into a wait state until the user chooses an action from the Events menu.

7 UC005 –Events List

7.1 Description & Priority

This functionality allows users to view all events stored under the users account profile. Once an event is setup; an account holder can view all events associated to the users account.

7.2 Actors

Primary Actors: User

7.3 Precondition

Each user must be successfully logged into the application and have at least one event created.

7.4 Activation

This use case starts when the user has successfully logged into the Appoint Remind application and selects an Event from the Events List page.

7.5 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when a User is successfully logged into the application	UI003 – Log-In
2. System displays a list of all events created under the users profile	UI005 –Events List AF001 – Update an Event AF002 – Delete an Event
3. End of use case.	

7.6 Alternative Flows

7.6.1 UC005 – AF001 – Update an Event

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 2 of Basic Flow	UI005 – List Events
2. User selects an Event from the listings they wish to edit	
3. The 'Change Event' page opens holding the users selection	
4. User makes required Amendments to the Event	
5. System updates the Event and displays list of all active events	UI005 – List Events
6. Use case ends	

7.6.2 UC005 – AF002 – Delete an Event

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at step 2 of the Basic Flow	UI005 – List Events
2. User selects an Event from the listings they wish to remove	
3. The 'Change Event' page opens holding the users selection	
4. User clicks on "Delete" button	MSG020.– Event Deleted: (x)
5. System Deletes the Event and displays list of all active events	AF002 – Event List
6. Use case ends	

7.6.3 UC005 – AF003 – Mandatory information missing

Steps	Alternative Flow / Data / Message / Business Rule
1. This flow starts at one of followings: – step 5 of AF001 when “Save” button is clicked with mandatory information not filled	
2. System displays an error message detailing missing information	MSG021.– Missing Event Details: (x)
3. Use case resumes to one of followings – step 6 of AF001 keeping filled information in related fields	

7.7 Termination

The Events List page is open; users can View/Edit/Delete/Create Events from this screen.

7.8 Post condition

The system goes into a wait state until the user chooses an action from the Events menu.

8 UC006 –Notification Frequency

8.1 Description & Priority

This use case describes the next stage of the Appoint Remind system. It has a high to medium priority as the user can setup a date/time when they wish to receive the notification of upcoming appointments/events. When researching the application; it was noted that users favoured ‘Alert/Toast Notification Types’ as opposed to SMS or Email notifications. The researched showed that different types of potential users wanted different amounts of notifications; these were mainly in the form of Toast Notifications.

This use case describes the function that allows users choose when they receive notifications prior to an appointment and/or upcoming medication consumption dates/times.

8.2 Actors

Primary Actors: User

8.3 Precondition

The system is in initialisation mode after the user selects Notification from the ‘All My Events page’.

8.4 Activation

This use case starts when a notification is required for the user in question.

8.5 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when a user selects the Notification from the 'All my Events' page.	
2. System prompts the user to choose a date then time of notification	
3. The user manually chooses the date and time of their preferred notification.	
4. Use case ends	

8.6 Alternative Flows

8.6.1 UC006 – AF001 – No Notification Date Setup

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when a user selects the Notification from the 'All my Events' page.	
2. System will not allow the user proceed unless they have chosen at least one date and time for their notification	
3. The use case continues at position 1 of the main flow	
4. Use case ends.	

8.7 Termination

The system presents the next option which is to add Notification.

8.8 Post condition

The system goes into a wait state until the user chooses 'Add Notification'.

9 UC007 – Notification Setup

9.1 Description & Priority

This use case describes how the Microsoft Push Notification Service (MPNS) is setup within the ‘Appoint Remind’ application. Push Notifications are used by mobile devices where a message is pushed from a server to a mobile device.

The following are the types of MPNS used within the ‘Appoint Remind’ Application:

- SMS: Short Message Service
- Email: a graphical representation that includes a short text or image
- Push: a pop-up text message appearing at the top of the device

9.2 Actors

Primary Actors: MPNS (Microsoft Push Notification Service)

Secondary Actors: User/Admin

9.3 Precondition

The user must have a valid account to use the ‘Appoint Remind’ application and must have at least one event object stored within the dB. The dB is stored using Azure and the ‘Appoint Remind’ application is connected remotely. The MPNS is setup within the application that is installed on the user’s device.

9.4 Activation

This use case starts when a User selects an Event Object and selects the type of push notification they wish to receive.

9.5 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when the user selects an existing Event and chooses the 'Notification Setup' from the menu.	
2. System displays the details of the selected event and prompts the user to choose a notification type.	
3. The user has a choice of three notifications. • Option 1: SMS • Option 2: Email • Option 3: Push A user can choose one or a mixture or all three options.	
4. The device sends the selection to Azure.	
5. Azure saves the selection type to the PNS	
6. The PNS forwards the notification chosen to the device	
7. Use case ends	

9.6 Alternative Flows

9.6.1 UC007 – AF001 – No Notification Selected

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when the user opens the notification type menu.	
2. System prompts the user to choose their notification type.	

3. User does not select any notification type.	
4. System does not allow the user proceed to the next stage.	MSG022.– Invalid
5. Use case ends.	

9.6.2 UC007 – AF002 –Notification with No Event entry

Steps	Alternative Flow / Data / Message / Business Rule
1. System prompts the user to choose their notification type.	
2. User chooses a Notification without selecting an Event	
3. System informs user that no Event exists	
4. Use case ends.	

9.7 Termination

The system presents the next option which is log-out of the Appoint Remind application.

9.8 Post condition

The system goes into a wait state until the user chooses to log-out.

10 UC008 – Grant Privileges to Event

10.1 Description & Priority

This use case describes how a user can assign privileges to an event object. Each user is the administrator of their own event object. If a user wishes to share an event object with other users they must create privileges for the object.

10.2 Actors

Primary Actors: User

10.3 Precondition

The user must have a valid ‘Appoint Remind’ account and must have at least one event object created within the system.

10.4 Activation

This use case starts when a User assigns privileges to other users to their event object.

10.5 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. User selects ‘Share Calendar’	
2. System displays a pop up box asking the user to enter the invitees ‘Email Address’.	
3. System checks if user is an existing user; i.e. setup on ‘Appoint Remind’.	
4. System sends a request to Server with Email.	
5. Server checks if Email exists.	

6. Server sends answer with status.	
7. Existing User: System authorises user to set privileges and displays applications 'Share Privileges' page.	
8. Non Existing User: System displays a pop up box stating 'No user found'.	
9. System sends email notification to request user to setup an account within the Appoint Remind Application.	
10. Use case ends.	

10.6 Alternative Flows

10.6.1 UC008 – AF001 – No Grant Privileges setup

Steps	Alternative Flow / Data / Message / Business Rule
1. The user chooses not to invite any other users.	
2. The system will proceed to the next stage	
3. Use case ends.	

10.6.2 UC008 – AF002 – Adds un-registered user

Steps	Alternative Flow / Data / Message / Business Rule
1. The user chooses not to invite any other users.	
2. System sends email notification to request user to setup an account within the 'Appoint Remind' Application	MSG023.– Invalid
3. Use case ends.	

10.6.3 Termination

The system presents the next option which is to choose notification type.

10.7 Post condition

The system goes into a wait state until the user chooses notification type.

11. UC009 – Push Notification Sending

11.1 Description & Priority

This use case describes how the push notifications are sent to the user's device. This has a high priority as the Push Notifications is the core of the functionality used within the 'Appoint Remind' application. When researching the application; it was noted that users favoured 'Push Notification Types' as opposed to SMS or Email notifications. The researched showed that different types of potential users wanted different amounts of notifications; these were mainly in the form of Toast Notifications.

This use case describes the function that allows users choose when they would like to receive notifications prior to an appointment and/or upcoming medication consumption dates/times.

11.2 Actors

Primary Actors: Microsoft Push Notification Service (MPNS)

Secondary Actors: Admin

11.3 Precondition

The system is in initialisation mode and the application is running without errors on the device.

11.4 Activation

This use case starts when a notification is issues for an event setup by a user.

11.5 Basic flow

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts when Azure detects that an Event must be sent to a user's device.	UI009
2. Azure checks the type of notification setup by the user	
3. The push notification is Queued to send to the users device	
4. Connects: Azure connects to the user's device.	MSG024.– Connected Successfully; notification sent
5. Successful: Notification received by the device	
6. Un-successful: Azure cannot connect to the device	MSG025.– Cannot connect to device
7. Use case ends	

11.5.1 UC009 – AF001 – Disconnected State

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts at step 7 of the basic flow when Azure cannot connect to the device.	
2. The notification was accepted and queued for delivery. The server waits until the device is connected to the application/server to send the notification.	MSG026.– Awaiting Connected State
3. The device is connected to the application/server	MSG027.– Connected Successfully; notification sent
4. Use case ends.	

11.5.2 UC009 – AF002 – Queue Full

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts at step 7 of the basic flow when Azure cannot send the notification to the user.	
2. The limit of the notifications is reached by the server.	MSG028.– Limit Reached
3. Use case ends.	

11.5.3 UC009 – AF003 – Invalid Subscription

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts at step 7 of the basic flow when Azure cannot find a valid Microsoft subscription.	
2. The cloud service stops sending notifications to the account holder.	MSG029.– Invalid Microsoft Account
3. Use case ends.	

11.5.4 UC009 – AF003 – Service Unavailable

Steps	Alternative Flow / Data / Message / Business Rule
1. This use case starts at step 7 of the basic flow when Azure cannot connect to any mobile devices.	
2. The cloud service stops sending notifications to all account holders.	MSG030.– Service Unavailable
3. Use case ends.	

11.5.5 Termination

The system allows a user to add new Events or log out of the application.

11.6 Post condition

The system goes into a wait state until all notifications are sent.

12 Data requirements

This section outlines how the data for the app is stored.

- **SQLite:** This is an embedded database for local client storage. All of the data entered by the user will be stored on a SQLite database that is saved within the application itself. This is the preferred choice for many mobile applications. This database will hold all of the users data and allow the user to create, read, update and delete their data.

- **Microsoft Azure:** When using the push notifications and publishing the application onto Windows Store, Azure will be used.

13 User requirements

This section describes the essential requirements needed by the user in order to install and run the application.

- **Windows Phone**: The user must have a Windows Phone to use the app. After further development, the app will also be available for use on iPhone and Android.

- **OS 8.1 or higher**: The app was developed using Windows 8.1 so the minimum operating system for the phone must be 8.1, if using older versions such as 7 or 8, there cannot be a guarantee that the app will run as expected.

- **Windows Store**: The user must have this installed on their phone in order to install the app.

- **Internet Access**: In order to install the app from the Widows store, the user must have internet access. Once installed, the app can run without internet access but it will have to be connected at regular intervals in order for the push notifications to be registered with the Push Notification Service (PNS).

14 Environmental requirements

This section describes the tools that were used to develop the application.

- **A HP laptop:** This was used to run the tools for development. The model that I own did not have Hyper V capabilities which are used to run the emulator for development and testing.

- **Windows Phone 8.1:** Instead of using the emulator for development and testing, I used an actual device which gave me a clear picture of how the application would look on a real device.

- **Visual Studio 2013:** Was used to develop and test the application. This is the preferred choice for developing all Windows applications.

- **Internet Access:** Vital in the development process. In order to research and learn how to develop the application, access to the WWW was paramount. Internet access is also needed to install the tools required for development. Visual Studio requires you to install SQLite, Microsoft phone SDK, Microsoft Visual C# Runtime and Windows Azure Mobile packages. These do not come as standard with Visual Studio.

- **Microsoft Azure:** This has to be used for publishing the application onto the Widows Store and to provide the push notification service.

15 Usability requirements

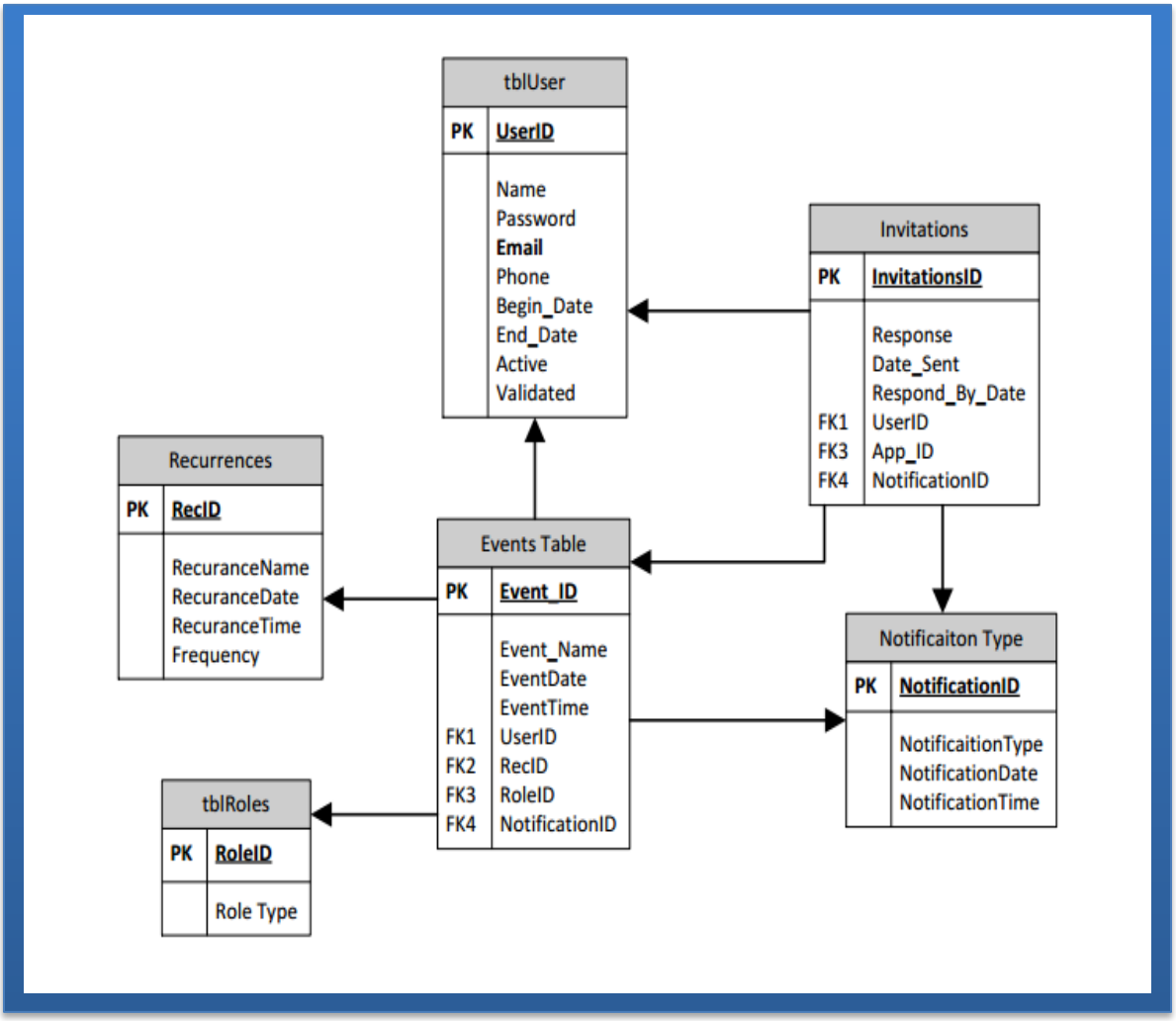
The main usability requirement for this application is that it is user friendly. This means that guidelines laid out for all mobile applications are adhered to. These rules are there to ensure that the user has a positive experience whilst using the application. Without considering these guidelines, the user could have a negative experience thus decreasing the chances of wanting to use the application again.

With this in mind, during development the different types of potential end user were considered. By using, easy to read and large buttons, the user should not get confused as to what each button does and/or find their way around the application quite easily.

For mobile applications it is advised not to have too many pages to navigate through, as the user will become uninterested very quickly. This was considered throughout development, so the application was kept to a minimum of pages needed to function correctly.

The date and time pickers were preferred over a calendar, as they are easier to view.

16 Design and Architecture



The above diagram shows the tables required for the database. Once a user creates a new account; their email address is validated and tested by sending the new user an email to the email address provided on registration. A user's account is not fully validated until the user responds by clicking the link issued. The Validated field on tblEvent; is defaulted to 0 when an account is created. On validation of the link issued, the 'Validated' field is set to 1.

A user roles table stores the roles that are available for each user within an Event table. The roles consist of Read, Write or Delete. Each user can have multiple events, and each role can be assigned to each event. A user can assign a role to other users provided they are the administrator of the Event in question.

There are different types of Notifications issued to a user. These notifications are selected by a user on completion of setting up an event. The Notification Type consists of Email, text message or push notification. When an invitation is issued to a user from a current user who wishes to share their event; the invitations table holds the type of notification issued.

The invitation table is used when a user decides to share their event object with another user. When an existing user creates and event they have the facility to share the event with another user. User A creates an event and inputs User B's email address. Using the email address the system checks if User B is a current account holder. If they are; the system sends them a link to the Event Object. If User B is not an existing customer; the system sends an email inviting them to register onto the application.

The re-occurrence table is used to input an event that occurs more than once. For example, a user may have an appointment re-occurring every Monday at 9.30. As opposed to re-entering this event each week, inputting a reoccurrence and stating the frequency will mean that the user will only need to enter this once. The system assigns and blocks off the calendar for every Monday at the time selected by the user. The system also records the type of notifications the user has selected for this type of event.

The MVVM (Model – View – View Model) pattern is used in the development of this application. This is similar to the MVC (Model – View – Controller) pattern. It is a pattern used by Microsoft Windows app development. It provides separation and means that the application is not tightly coupled. This means that there is a clean separation between the application logic and the UI (User Interface) which is beneficial as it makes the application easier to test, maintain and evolve.

This pattern is mainly used in conjunction with XAML (eXtensible Application Markup Language) platforms which will also be used in this application.

17 Implementation

This section describes the technologies used in the implementation of the Windows Phone application. It also covers the methodologies used in the implementation.

17.1 Technology Overview

The approach used to build the Widows Phone application was that of creating a native application. Native apps for Windows Phone are installed through the Windows Store. They allow for taking advantage of the devices' features. The application uses Windows preferred language for developing native apps - C#, XAML is used is used for initializing structured values and objects and SQLite is used as the database.

17.1.2 Technologies

C# (sharp): is an object orientated programming language. This was the language used to develop the 'Appoint Remind' application.

XAML (eXtensible Application Markup Language): This is used to create the views (GUI).

SQLite: Is used to store the applications data within the phone itself.

MVVM (Model View – View – Model): This pattern separates the responsibility for the appearance and layout of the UI from the responsibility for the presentation logic

17.1.3 Code Examples

This is an example of how XAML is used to create the Registration page. Extensible Application Markup Language (XAML) is a language used to create a design of each page within Visual Studio.

```
<Page
    x:Class="APPOINTREMIND.Views.Register"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:APPOINTREMIND.Views"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title-->
        <StackPanel Grid.Row="0" Margin="0,17,0,28">
            <TextBlock FontSize="35" Text="Register" Margin="9,-7,0,0" />
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
            <StackPanel >
                <TextBlock FontSize="25" Text="Name"/>
                <TextBox Name="NametxtBx"/>

                <TextBlock FontSize="25" Text="Email"/>
                <TextBox Name="EMailtxtBx"/>

                <TextBlock FontSize="25" Text="PhoneNumber"/>
                <TextBox InputScope="Number" Name="PhonetxtBx" />

                <TextBlock FontSize="25" Text="Password"/>
                <PasswordBox Name="txtPassword" VerticalAlignment="Top"
Width="376" />

                <Button x:Name="RegisterOK" Content="Ok" Command="{Binding
                Path=OkCommandRegister}"
                CommandParameter="{Binding ElementName=txtPassword}"
                Margin="0,0,-0.167,0" Width="376" Click="Register_Click"/>
            </StackPanel>
        </Grid>
    </Grid>
</Page>
```

XAML is similar to HTML (HyperText MarkUp Language); it is the graphical user interface (GUI) of each page. XAML is the page that a user interacts with at run time. This language can be coded by hand or using a design tool within Visual Studio.Net.

This is the code behind the XAML page. From here the values that are entered into each textbox are passed into the database helper class.

```
namespace APPOINTREMIND.Views
{

    public sealed partial class Register : Page
    {
        public Register()
        {
            this.InitializeComponent();
        }

        private async void Register_Click(object sender, RoutedEventArgs e)
        {
            {
              DatabaseHelperClass Db_Helper = new DatabaseHelperClass();//Creating
              object for DatabaseHelperClass.cs from Helpers/DatabaseHelperClass.cs

              if (NametxtBx.Text != "" & EMailtxtBx.Text != "" & PhonetxtBx.Text !=
              ""    & txtPassword.Password != "")
                {
                     Db_Helper.Insert(new Users(NametxtBx.Text, EMailtxtBx.Text,
                     PhonetxtBx.Text, txtPassword.Password));
                     Frame.Navigate(typeof(AddEvent));//after add user redirect to
                     add event page
                }
                else
                {
                     MessageDialog messageDialog = new MessageDialog("Please
                     complete  all fields");//Text should not be empty
                    await messageDialog.ShowAsync();
                }
            }
          }
        }
    }
```

When a XAML file is created a code behind class event is created automatically. The class page is used to pass in values from the XAML page.

Register Click

The Register_Click was defined within the Register.XAML page. A command button 'OK" button was added, which is used as a method of selection. In this scenario; when the OK button is clicked by a user; an event is triggered.

Within the Register.xaml page; a user enters: Name, Email Address, Phone Number and Password. These details are passed in from the Register.XAML page to the class page so the system can do some processing with the entries. In this scenario the data passed in is saved to the dB.ReadAllUsers.class

The ReadAllUsers class is defined within the Database Helper class. This is a function is used to connect to the dB, read the users input and return the details of the dB back.

```
// Retrieve the specific user from the database.
public Users ReadAllUsers(int userid)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
      var existingUser = dbConn.Query<Users>
      ("select * from Users where Id  =" + userid).FirstOrDefault();
        return existingUser;
    }
}
```

The procedure: ReadAllUsers: passes in userID as the parameter input

The system first connects to the dB using the SQLiteConnection

Once a valid connection is established; the system checks if the user exists within the dB

```
var existingUser = dbConn.Query<Users>("select * from Users where Id =" +
userid).FirstOrDefault();
                return existingUser;
```

If the user is found within the dB; the system returns the user details using the UserID where details of the user are output screen

This code is from the Database Helper class which is used to update, delete and create an event.

```
//Update existing event
public void UpdateEvent(Events events)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
      var existingevent = dbConn.Query<Events>("select * from Events where
      EventId =" + events.EventId).FirstOrDefault();
        if (existingevent != null)
        {
            existingevent.EventName = events.EventName;
            existingevent.CreationDate = events.CreationDate;
            dbConn.RunInTransaction(() =>
            {
                dbConn.Update(existingevent);
            });
        }
    }
}

//Delete specific event
public void DeleteEvent(int EventId)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
      var existingevent = dbConn.Query<Users>("select * from Events where
      EventId =" + EventId).FirstOrDefault();
        if (existingevent != null)
        {
            dbConn.RunInTransaction(() =>
            {
                dbConn.Delete(existingevent);
            });
        }
    }
}

//Insert the new event in the Event table.
public void Insert(Events newevent)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
        dbConn.RunInTransaction(() =>
        {
            dbConn.Insert(newevent);
        });
    }
}
```

UpdateEvent

The UpdateEvent is used to update the Events created by a user.
The system first connects to the dB using the SQLiteConnection.

Once a valid connection is established; the system passes in the EventID from the users selection.

```
var existingevent = dbConn.Query<Events>("select * from Events where EventId =" +
events.EventId).FirstOrDefault();
                if (existingevent != null)
                {
                    existingevent.EventName = events.EventName;
                    existingevent.CreationDate = events.CreationDate;
                    dbConn.RunInTransaction(() =>
                    {
```

System retrieves the Event within the system.

System updates the current record with the new details.

```
                    {
                        dbConn.Update(existingevent);
                    });
```

DeleteEvent

The DeleteEvent is used to delete the Events created by a user.

The system first connects to the dB using the SQLiteConnection.

Once a valid connection is established; the system passes in the EventID from the users selection.

```
var existingevent = dbConn.Query<Users>("select * from Events where EventId =" +
EventId).FirstOrDefault();
                if (existingevent != null)
```

System retrieves the Event within the system.

System deletes the current record with the new details.

```
                {
                    dbConn.Delete(existingevent);
                });
```

This is the code behind the XAML page. From here the values that are entered into each textbox are passed into the database helper class.

```
namespace APPOINTREMIND.Views
{

    public sealed partial class Register : Page
    {
        public Register()
        {
            this.InitializeComponent();
        }

        private async void Register_Click(object sender, RoutedEventArgs e)
        {
            {
              DatabaseHelperClass Db_Helper = new DatabaseHelperClass();//Creating
              object for DatabaseHelperClass.cs from Helpers/DatabaseHelperClass.cs

              if (NametxtBx.Text != "" & EMailtxtBx.Text != "" & PhonetxtBx.Text != ""
              & txtPassword.Password != "")
                {
                     Db_Helper.Insert(new Users(NametxtBx.Text, EMailtxtBx.Text,
                     PhonetxtBx.Text, txtPassword.Password));
                     Frame.Navigate(typeof(AddEvent));//after add user redirect to add
                     event page
                }
                else
                {
                     MessageDialog messageDialog = new MessageDialog("Please complete
                     all fields");//Text should not be empty
                    await messageDialog.ShowAsync();
                }
            }
          }
        }
    }
```

This is the code in the Database Helper class that is used to select a particular user and insert a new user to the database.

```
// Retrieve the specific user from the database.
public Users ReadAllUsers(int userid)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
       var existingUser = dbConn.Query<Users>("select * from Users where Id
      =" + userid).FirstOrDefault();
        return existingUser;
    }
}

// Insert the new User in the Users table.
public void Insert(Users newuser)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
        dbConn.RunInTransaction(() =>
        {
            dbConn.Insert(newuser);
        });
    }
}
```

This code is from the Database Helper class which shows how to update, delete and create an event.

```
//Update existing event
public void UpdateEvent(Events events)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
      var existingevent = dbConn.Query<Events>("select * from Events where
      EventId =" + events.EventId).FirstOrDefault();
        if (existingevent != null)
        {
            existingevent.EventName = events.EventName;
            existingevent.CreationDate = events.CreationDate;
            dbConn.RunInTransaction(() =>
            {
                dbConn.Update(existingevent);
            });
        }
    }
}

//Delete specific event
public void DeleteEvent(int EventId)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
      var existingevent = dbConn.Query<Users>("select * from Events where
      EventId =" + EventId).FirstOrDefault();
        if (existingevent != null)
        {
            dbConn.RunInTransaction(() =>
            {
                dbConn.Delete(existingevent);
            });
        }
    }
}

//Insert the new event in the Event table.
public void Insert(Events newevent)
{
    using (var dbConn = new SQLiteConnection(App.DB_PATH))
    {
        dbConn.RunInTransaction(() =>
        {
            dbConn.Insert(newevent);
        });
    }
}
```

18 Testing

#	Functionality	Scenario	Description	Notes	Tested by	Automated	Pass/Fail
1.	Start up	App Loads	After tapping on icon, app loads within 2 seconds		DG	Yes	Pass
2.	Login	As an existing user can I log in?	Login with existing account.	App does not recognise existing user if its newly installed	DG	Yes	Fail
3.	Login	As an existing user can I log in with incorrect details?	Login with existing account but enter incorrect details.	App displays popup message stating "Incorrect Login; please try again".	DG	Yes	Pass
4.	Login	Register button works.	New user taps register button.	User is brought to registration page	DG	Yes	Pass

5.	Login	Existing user tries to login.	Existing user enters details and taps login button.	User is brought to the Event page.	DG	Yes	Pass
#	**Functionality**	**Scenario**	**Description**	**Notes**	**Tested By**	**Automated**	**Pass/Fail**
6.	Register	User enters details	Textbox saves new users information		DG	Yes	Pass
7.	Register	Are all fields filled in correctly?	User must fill in all four fields	Ok button was tapped to move to next page but message box popped up stating "Please complete all fields". User cannot move to next section without filling all fields	DG	Yes	Pass
8.	Register	System checks if user details are saved to the system	Message Box appears on screen: New Account Setup	No message returned on screen	DG	No	Fail
9.	Register	System checks if Account is found within the dB	Message Box appears on screen: Account Exists	No message returned on screen	DG	No	Fail

#	Functionality	Scenario	Description	Notes	Tested by	Automated	Pass/Fail
10.	Event Page	Add a new Event.	Can the user add a new event?		DG	Yes	Pass
11.	Event Page	Date & Time Picker.	When the date & time pickers are tapped, do they work?	When date & time pickers are tapped, the pickers appear on screen allowing the user select their preferred date and time	DG	Yes	Pass
12.	Event Page	No event added.	What happens if there is no event added to the textbox?	Message pops up on screen asking the user to "Please fill field". User cannot proceed.	DG	Yes	Pass
13.	Event Page	No date or time chosen.	What happens if the user does not add a date or time of chosen event.	When the "add event" button is pressed, the app allows the user to move onto next stage.	DG	Yes	Fail

#	Functionality	Scenario	Description	Notes	Tested by	Automated	Pass/Fail
14.	Show Events	Are events displaying?	Does the “All my events” page show users event/s?	User’s events are shown in descending order with the most recent on top.	DG	Yes	Pass
15.	Show Events	Are events displaying correct information?	Are the events displayed showing the correct data as well as the date and time chosen by user?	All events displayed are showing correct data and date/time chosen using the pickers.	DG	Yes	Pass
16.	Show Events	Add event	Does the “add event” button bring the user back to the add event page?	By tapping “add event” button, user is brought back to “add event” page.	DG	Yes	Pass
17.	Show Events	Notification	Does the “notification button bring the user to the next page?	By tapping the “Notification” button, user is brought to the notification page.	DG	Yes	Pass

18.	Show Events	Event	When an event is tapped, does it bring the user to the “Change Event page”	When an event is tapped, the user is brought to the “Change Event” page	DG	Yes	Pass
#	**Functionality**	**Scenario**	**Description**	**Notes**	**Tested By**	**Automated**	**Pass/Fail**
19.	Change Event	Delete Event	When “Delete” button is tapped, is the event deleted?	Event is not deleted when “Delete” button is tapped. User is brought back to “Show Events” page.	DG	Yes	Fail
20.	Change Event	Update Event	When event is updated, do the changes happen?	When “Update” button is tapped, the event is updated. User is brought back to “Show Events” page with event updates made correctly.	DG	Yes	Pass

#	Functionality	Scenario	Description	Notes	Tested By	Automated	Pass/Fail
21.	Choose Reminder Date & Time	Choose Date & Time	Do the date & time pickers work as expected?	When date & time pickers are chosen, the date & time pickers appear on the screen, allowing the user to manually select date & time.	DG	Yes	Pass
22.	Choose Reminder Date & Time	Choose Date & Time	Are the date & time chosen by the user saved?	Date & time chosen by the user are not saved.	DG	Yes	Fail
23.	Choose Reminder Date & Time	Choose Date & Time	Does the “Add Notification” button work as expected?	When the “Add Notification” button is tapped, users are directed to the next page.	DG	Yes	Pass

24.	Choose Reminder Date & Time	Choose Date & Time	Does the “Back” button work as expected?	When the “Back” button is tapped, users are directed to the previous page.	DG	Yes	Pass

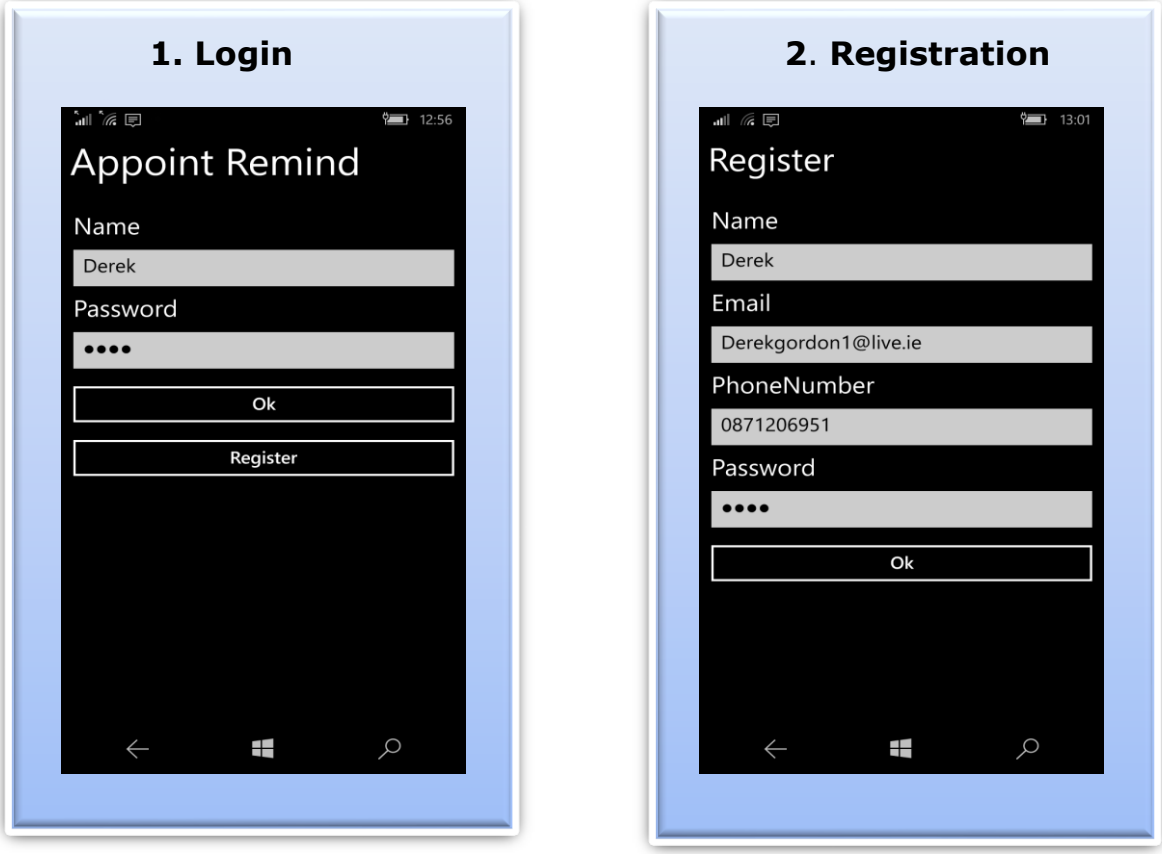
#	Functionality	Scenario	Description	Notes	Tested By	Automated	Pass/Fail
25.	Choose Notification	Choose Notification Type	When a SMS notification is chosen by the user, do they receive the notification at their chosen date & time?	When SMS notification is chosen, the SMS is delivered immediately.	DG	Yes	Fail
26.	Choose Notification	Choose Notification Type	When a SMS notification is chosen by the user, do they receive the notification at their chosen date & time?	When SMS notification is chosen, the SMS is delivered immediately.	DG	Yes	Fail
27.	Choose Notification	Choose Notification Type	When a Push notification is chosen by the user, do they	When Push notification is chosen, nothing happens.	DG	NO	Fail

			receive the notification at their chosen date & time?				

#	Functionality	Scenario	Description	Notes	Tested By	Automated	Pass/Fail
28.	Grant Privileges	Allow a user to share an Event	A user can assign privileges to an event object	Was unable to share out Events to other account users	DG	No	Fail

#	Functionality	Scenario	Description	Notes	Tested By	Automated	Pass/Fail
29.	Push Notification Sending	User chooses push notification	Push Notification appears on screen	No connection from Azure to application	DG	No	Fail

19 Graphical User Interface (GUI) Layout

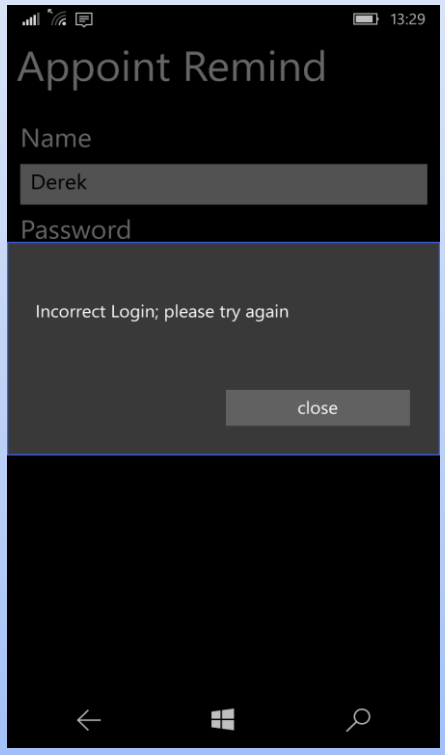


After successful installation of the application, the user will be directed to the (1) login screen.

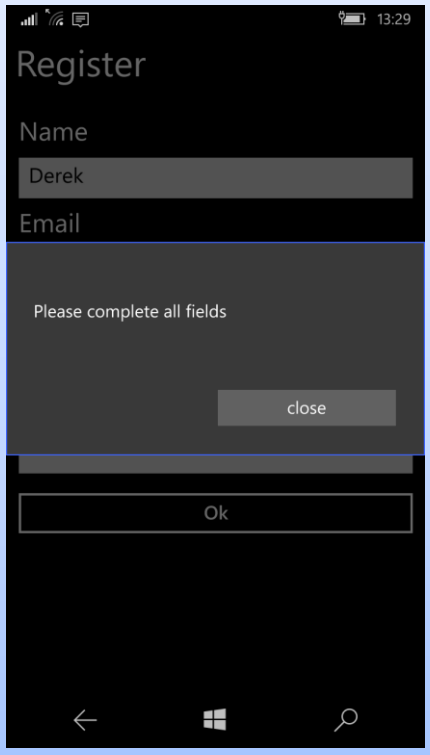
As this is the first time that they are using the app, they will have to register (2).

If a user tries to login using unrecognised credentials, they will be redirected to the Registration page. Here they will enter their preferred user name, email address, phone number and password.

1a. Invalid Login

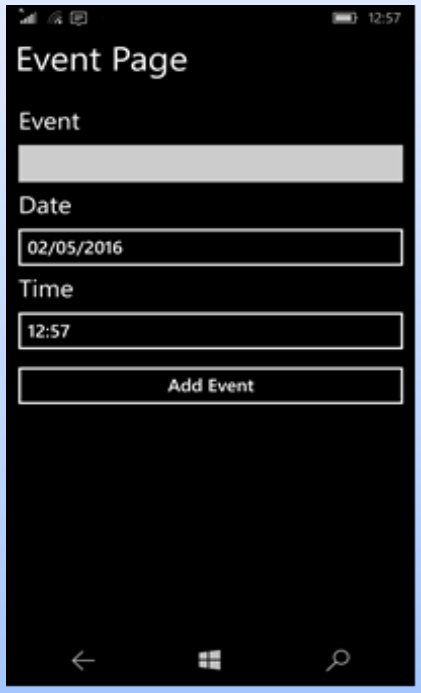


2a. Incomplete form

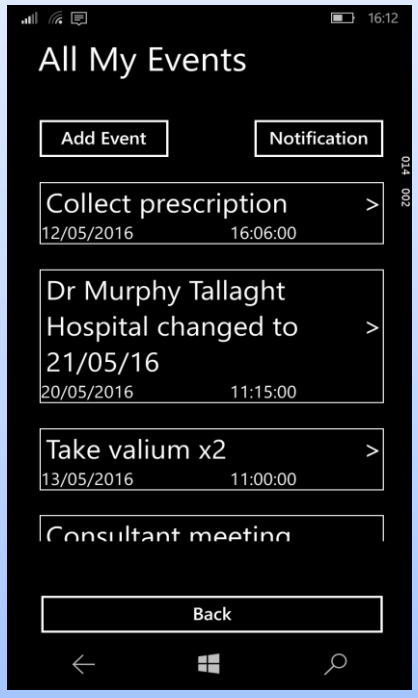


If a user tries to log in with incorrect credentials, the above message is displayed as seen in image 1a. If a user does not complete all of the text fields on the registration, the message on 2a will appear.

3. Add Event



4. Events List

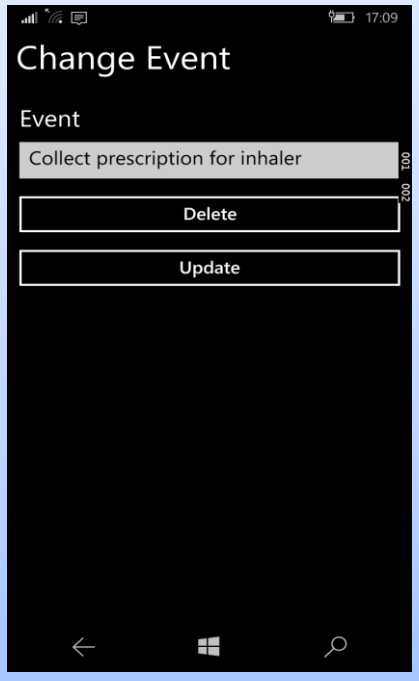


After successful login/registration the user is brought to Add Event Page (3)

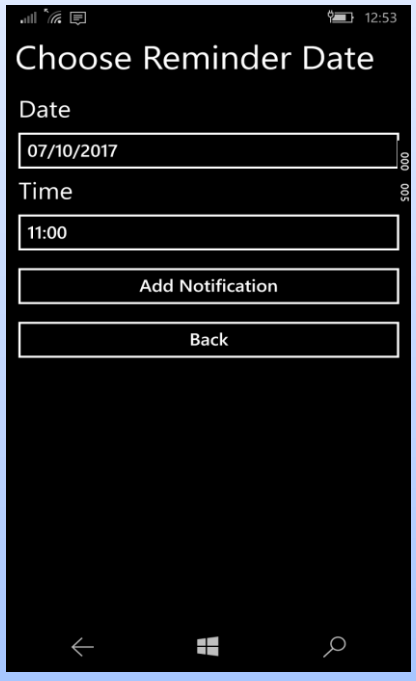
Here they can manually enter the event or appointment, then add in the date and time using the date/time pickers.

The “Add Event” button then brings the user to (4) Events list page. This page displays the event/s that the user has created. All previous events that have not been deleted are displayed on this page too.

5. Update/Delete Event



6. Reminder Date/Time



When the user taps on a certain event they will be brought to Change Event page (5). The event can then be deleted or updated to add more information in.

When the user picks which type of update they want, they are redirected back to the Events List page (4).

The results of their changes will be shown here. They then tap the “Notification” button which brings them to the Reminder Date/Time page (6). Here the user manually chooses their preferred date and time to receive their notification by using the date and time pickers

20 Customer testing

To gain feedback from potential end users, the application was shown to users from all backgrounds and walks of life.

- **Mom of three young children:** Mary: When I spoke to her first about the idea for the app she thought it would be a great idea as with three young children; she has a lot of organising to do regarding appointments and medication to be taken. She has an iPhone and never used a Windows phone before so she found the style and layout very different to that of the iPhone. I didn't tell her exactly how the app worked, just a general overview of what I intended it to do. I wanted her to play around with it and see if it was as user friendly as I'd hoped it was. The results were that she found the app easy to navigate and understood what each function did. At the time of testing the app was not functioning correctly (the notifications were being sent immediately) Mary said she would like to give it a trial when it is functioning properly to see if it suits her needs. She would prefer to have an iPhone version of the app as that is what she uses.

- **Retiree:** James: Has been retired for six years, he had to finish work earlier than he had hoped in his career due to ill health. His illness requires him to take medication at set times every other day. When I told him my idea for the app back in September, he thought that it would be a very useful tool. He does not have much experience of using a smart phone so I had to guide him through it the first time he used it. He thought that it was easy to use as the buttons were large, there was not a lot of text to read and there were not too many pages to navigate through. He said he liked the idea of the app having recurring notifications as he would not want to have to add a new notification every day. At the time of James testing the app, this functionality was not yet working correctly but he was in favour of that function.

- **Plumber:** Mick: Has his own business as a plumber. When I first told him about the app idea in September he asked if it could be used for other notifications other than medical appointments. When I was initially designing the app I just had medical appointments in mind but then after talking to him I realised it could be used for any type of upcoming events. He uses an Android phone at the moment so was a bit put off by the square look of the Windows phone apps.
  Aesthetics aside he found it fast and responsive. When completed and fully functioning he said he would like to try it out to see if it will help his organisation of business appointments and jobs.

21 Evaluation

In order to publish the application, Windows Store performs its own standard tests to ensure it is up to the required standard to be published. Merely as a test I tried to publish the application; however it was rejected as I did not have the correct size logo that appears on the tiles. All other preliminary checks were passed.

When the application is fully functioning, I will get users with different levels of computer literacy to test the system, from installation right through to adding the notifications. They will not be told how to use the system, or will not be informed on how the system works or what it intends to do. The results will show up any defects or bugs that were not caught during my own testing.

22 Conclusions

The advantage of this Application is that it works in a quick and easy manner it's like a personal helper, in that it provides reminders of future appointments/events. It even takes the worry out of remembering to take medication.

It removes the worry/inconvenience of managing paper schedules and lets the user concentrate on actual events. However, the disadvantage is that further research is required (as described in section 23) in order to enhance the product.

For the purpose of this project though the initial aim has been achieved in that the application will acts as a data store, users can access their information, receive notifications from anywhere at any location.

23 Future development and research

There are many features that could be included in the future but given the time constraints they would not be possible for this project. During research I looked at similar applications on the market such as "medicine reminder", users stated that they would like to have an option to take a picture of the medication or have an image of their medication required.

24 References

C# Best Practices: Improving on the Basics. [ONLINE] Available at: https://app.pluralsight.com/player?course=csharp-best-practices-improving-basics&author=deborah-kurata&name=csharp-best-practices-improving-basics-m2&clip=3&mode=live. [Accessed 19 January 2016].

Database design and development, Co Cork, Ireland. [ONLINE] Available at: http://w3.ie/database-design. [Accessed 09 October 2015].

Date and Time Picker Control in Windows Phone 8.1. [ONLINE] Available at: http://www.c-sharpcorner.com/UploadFile/2d2d83/date-and-time-picker-control-in-windows-phone-8-1/. [Accessed 10 March 2016].

How to create a local database app with MVVM for Windows Phone 8. [ONLINE] Available at: https://msdn.microsoft.com/en-us/library/windows/apps/hh286405%28v=vs.105%29.aspx?f=255&MSPPError=-2147217396. [Accessed 13 January 2016].

Interactions and usability with Windows Phone. [ONLINE] Available at: https://msdn.microsoft.com/en-us/library/windows/apps/hh202889(v=vs.105).aspx. [Accessed 09 February 2016].

Notification Hubs Overview. [ONLINE] Available at: https://msdn.microsoft.com/en-us/library/azure/jj927170.aspx. [Accessed 20 October 2015].

Navigating Between Pages | Windows Phone 8 Development for Absolute Beginners | Channel 9. [ONLINE] Available at: https://channel9.msdn.com/Series/Windows-Phone-8-Development-for-Absolute-Beginners/Part-18-Navigating-Between-Pages. [Accessed 4 January 2016]

Retrieving, Storing and Displaying Data in Windows Phone Apps -- Visual Studio Magazine. [ONLINE] Available at: https://visualstudiomagazine.com/articles/2013/11/01/retrieving-storing-and-displaying-data.aspx. [Accessed 10 April 2016].

Roadmap for Windows Runtime apps using C# or Visual Basic - Windows app development. [ONLINE] Available at: https://msdn.microsoft.com/en-us/library/windows/apps/xaml/br229583.aspx. [Accessed 19 January 2016].

Setting up an authenticated web service to send push notifications for Windows Phone 8. [ONLINE] Available at: https://msdn.microsoft.com/en-us/library/windows/apps/ff941099(v=vs.105).aspx. [Accessed 10 April 2016].

Twilio Docs: C# Helper Library . [ONLINE] Available at: https://www.twilio.com/docs/csharp/install. [Accessed 20 April 2016].

Sqlite Sample for Windows Phone 8.1 (C#-Xaml) in C# for Visual Studio 2013. [ONLINE] Available at: https://code.msdn.microsoft.com/windowsapps/Sqlite-Sample-for-Windows-83d88433. [Accessed 3 January 2016].

Sending push notifications with Azure Notification Hubs on Windows Phone | Microsoft Azure. [ONLINE] Available at: https://azure.microsoft.com/en-us/documentation/articles/notification-hubs-windows-phone-get-started/. [Accessed 10 April 2016].

The MVVM Pattern. [ONLINE] Available at: https://msdn.microsoft.com/en-ie/library/hh848246.aspx?f=255&MSPPError=-2147217396. [Accessed 20 October 2015].

What is XAML? [ONLINE] Available at: https://msdn.microsoft.com/en-us/library/cc295302.aspx?f=255&MSPPError=-2147217396. [Accessed 21 October 2015].

Quickstart: Sending an email message Windows Phone Store app using C++, C#, or Visual Basic (Windows). [ONLINE] Available at: https://msdn.microsoft.com/en-us/library/windows/apps/xaml/dn632427.aspx. [Accessed 03 March 2016].

Appendix A

1.0 Project Proposal

Project Proposal

Appoint Remind

Derek Gordon

X12110370

X12110370@student.ncirl.ie

BSc (Hons) in Computing

Specialisation: Networking and Mobile Technologies

Date: 29/09/2015

2.0 Objectives

The main objective for this project is to develop a mobile application that will

a) Allow users to organise their medical appointments and to have all related information in the one location.

The application will be a calendar in which users can manually enter their various upcoming appointments

b) Act as a reminder to take medications i.e. to inform users with certain medical conditions that they have to take their medication at a particular date/time.

The user will simply input the prescription details including dates and times, notifications will be sent to the user to remind them of when they need to take their medication.

The application will inform the user through text, email notifications and push notifications thus reducing the possibility of missed medication.

Nominated users such as parents, guardians and carers can use this application to ensure appointments are managed, and/or where applicable medications are taken at the prescribed times.

3.0 Background

The reason I decided to create this type of application is that after talking to friends and family who have children with special needs, I have found out that they sometimes struggle to keep up with scheduled appointments. The parents receive letters from hospitals detailing the appointments but they could receive these letters up to eight or nine months in advance of the appointment. Unless they are extremely organised, there is a possibility that the letter might get misplaced therefore leading to a missed appointment. Occasionally two different appointments are made for the same day/time.

As part of research conducted, it was found that only certain hospitals and institutions will send an appointment reminder. Invariably parents do not receive notifications from the majority of public hospitals, therefore there is a possibility that appointments could be missed.

The feedback received from parents suggests that they would like to have an application that they could use to keep all of their information in the one place that can be accessed anytime and would also send notifications to the parents of upcoming appointments.

In relation to using the application for managing medication, there are a number of campaigns running at the moment for people with Alzheimer's. The key message is to have a system/process in place at an early stage to remind people of what they need to do in regards to remembering things. One area that people tend to forget is when they took their medication, the application can be used as a reminder by the person themselves or their carer. This is just one example of how the application can be used for managing medication use.

It has been suggested that it would be beneficial if the application could be shared within a domain, this means the two or more users can use the same calendar and add, update and delete the same information.

4.0 Technical Approach

In order to complete this project I will have to conduct research in regards to gathering information from potential users as to how they would like this application to work i.e. what information should be stored on it?, the look and feel?, consider the users level of experience of using an application, how often and by what means they would receive the notifications?.

Once the requirements gathering at a high level is completed then I will have to research how to implement this idea.

I will have to further my knowledge of C# (sharp) programming language, SQLite databases, XAML, MVVM, Visual Studio 2013. I will have to learn how notifications can be sent through email, SMS text messages and push notifications.

I will use online tutorials on windows phone app development, asp.net tutorials to familiarise myself with the MVVM (Model – View – View Model) pattern, C# and the use of Visual Studio 2013.

Microsoft virtual Academy and Pluralsight will also be used for online tutorials.

5.0 Special resources required

In order to complete this project I will use a laptop with Windows 10 to develop the application and a windows phone to test and run the application on. Visual Studio 2013, an emulator and Microsoft Azure will be the software tools used.

Project Plan	2015				2016				
Appoint Remind	**Sept**	**Oct**	**Nov**	**Dec**	**Jan**	**Feb**	**Mar**	**Apr**	**May**
Concept creation									
Proposal									
Requirements Specification									
Analysis & Design									
Development									
Mid-Point Presentation									
Final Upload/Presentation									
Testing									
Technical Report									

7.0 Technical Details

I will use C# programming language, Visual Studio 2013, SQLite for the database and Azure to publish the application.

8.0 Evaluation

It is my intention to use SQLite to store the data and then test the Database connectivity. When the database and the application are connected, I can add data to the database to see if it is displayed correctly. Ongoing evaluation will be key to this project as it will point out any flaws or issues that I had not previously thought of arise.

The Agile Scrum process will be used for continuous deployment of sections of the application, this will allow for feedback from potential end users.

9.0 Monthly Journals

9.1 Month: September

When we started back in college on the 15/09/2015 I realised I should have been thinking about the final year project throughout the summer. I had to do a project throughout the summer as I could not do the work placement, I was concentrating on that and never even considered the final year project.

First night back was an eye opener, felt totally overwhelmed walking out that night. Panic started to set in a bit as I only had a few weeks before the project proposal had to be uploaded.

After a few weeks of driving myself and everyone around me mad, I finally came up with an idea that I thought would be suitable. I spent a few weeks asking around for ideas and looking for inspiration. Any idea I came up with was either too simple or too complicated. I struggled to find an idea that would be tick all of the boxes required. I needed to come up with something that I would actually be

interested in doing rather than just doing something just to get through the final year.

Then just five days before I had to upload my proposal I was talking to a friend who was just saying something in passing that I thought would be a great idea. The penny had finally dropped.

The idea for the project is to create an application stores all medical appointments in the system and that can be accessed by any person who is nominated by the user. The system will send notifications of upcoming appointments. The notification function of the application can also be used to remind people with medical conditions that they have to take their medicines at a particular date/time.

9.2 My Achievements

This month, I was able to achieve probably one of the hardest parts of the project….come up with a viable idea. Once the idea is finalised, I can just totally focus on that. Up until a few days before the proposal had to be uploaded, I still didn't know what I was going to do. Once I had the idea and spoke to Eamon about it, I was very happy as panic starting to creep in.

Once the project idea was finalised I started to do a bit of requirements gathering. I spoke to a number of friends and family to see how the application could benefit them or even if they would use such a system. I received a lot of positive feedback so far.

9.3 My Reflection

I should have been thinking about the final year project as early as possible. Big mistake not being prepared.

9.4 Intended Changes

Next month I will have to start requirements gathering ASAP, once I know what's going to be in the application I can focus on that. I need to keep on top of this or I will struggle to keep up with the deadlines. There is a lot of learning to do just to

see how this project will work. I need to brush up on my programming skills as it's something I have struggled with throughout college so far.

There's so much work to do its daunting but I feel that it will be a great learning curve. There's no getting away from the fact that this will be tough and I will be under a lot of pressure but I will be delighted with myself if I get can through this.

10.0 Supervisor Meetings

Date of Meeting: 29/09/2015

Items discussed: Discussed the project idea with Eamon, I was delighted when he said it was a good idea. After the previous few weeks of worry this chat put my mind at ease.

Action Items: Get started ASAP!

11.0 Month: October

11.1 My Achievements

This month, I was had my first meeting with my supervisor (Manuel). This was a great benefit to me as in the first few weeks of the month I didn't get much done on my project as I started a new job. After having the meeting with Manuel I was happy as he gave me some direction on how to start the project. He advised me on the Agile scrum concept and to use that along with online project management tools. This gave me the focus that I needed. Before the meeting I was really worried as I was struggling to see where and how I would even start the project.

My contributions to the projects included requirements gathering. I conducted a survey using survey monkey and distributed it to family and friends, I was pleased with the results as I gained an insight on how people would use my application, what they wanted to see in it and what they would probably not use. I wanted to information from people of all ages and walks of life as this would give me a better understanding of what I need to have on my application.

11.2 My Reflection

I felt, it worked well to have the meeting with my supervisor as he gave me direction on how to start the project and how to keep on top of things as it proceeds.

However, I was not successful in starting to do any coding or choosing which technologies I will use to create my application. This month all I done was the requirements gathering and filling out the requirements specification. I understand now how important that document is as before I started to fill it out I only had a vague idea of what I wanted in my application and how it should look. As I was filling out the document, everything began to become clearer. The requirements gathering was a great tool as I received input from a lot of different type of potential users which gave me ideas of what should be in the application and what is not needed.

11.3 Intended Changes

Next month, I will try to research which technologies will best suit the type of application that I am trying to create. As soon as I figure out which would be the best I can concentrate on learning these and starting to get some coding done. As per the Agile scrum concept I have to have something to show my supervisor within the next couple of weeks.

I realised that I need to do a lot of research on which technologies I am going to use, some might be more beneficial to the project than others.

12.0 Supervisor Meetings

Date of Meeting: 23/10/2015

Items discussed: Project Proposal, ways to approach project, Agile concept, Project Management and requirements gathering.

Action Items: We discussed how the Agile process works, this involves breaking down the project into shorter mini projects where by at the end of a

certain time period (in this case three weeks) I have to have something to show the client i.e. supervisor at the end of every three week period.

This month though I concentrated on my requirements gathering and getting the requirements specification completed.

13.0 Month: November

13.1 My Achievements

This month, I researched which type of software I will use for my project. There seems to be to be said for using Ruby on Rails but I will have to learn a whole new programming language. I have to weigh up the pros and cons of learning a new language. Time is a major factor in this decision.

My contributions to the projects included starting the Analysis Design document. This document will help me understand the system requirements and will help in the direction I need to complete the project.

13.2 My Reflection

I felt, it worked well to complete the requirements specification document, before I done this I had a vague idea of what I wanted the application to do, after completing research and gaining feedback from potential end users, I had a much clearer picture of what is actually required. Before I started to put all of this down in a document, everything was a blurry as I knew what I would like the application to do but I didn't know if people would actually use it or use the functionality that I had envisaged. After carrying out surveys, using potential end users from every walk of life, I gained the knowledge of what exactly the potential end user wants and what they would use and not use in the application.

13.3 Intended Changes

Next month, I will have to complete the Analysis Design document. This will provide me with clarity in regards to how the project will be actually organised and completed.

14.0 Supervisor Meetings

Date of Meeting: No meeting this month.

Items discussed: N/A

Action Items: N/A

15.0 Month: December

15.1 My Achievements

This month, I unfortunately did not get anything done with my project. There was the death of a close relative and I everything stopped in regards to college and the project.

15.2 My Reflection

However, I was not successful in completing the Analysis Design document. This will have to be done in January. This has put my plan for the project back by a good margin but unfortunately these things happen and cannot be accounted for in the planning of a project.

15.3 Intended Changes

Next month, I will try to get back on track with the project. I need to complete the Design Analysis document and hopefully start doing some actual coding.

Supervisor Meetings

Date of Meeting: No meeting this month

Items discussed: N/A

Action Items: N/A

16.0 Month: January

16.1 My Achievements

This month, I was able to finally get the design analysis document finished. This document helped greatly as it gave me the whole picture of what exactly is needed to complete the project.

My contributions to the projects included finishing the design analysis document, preparing for the mid-point presentation and figuring out how certain aspects of the project will work, i.e. how the SMS and push notification functions will work.

16.2 My Reflection

I felt, it worked well to meet up with Manuel as he helped a lot and pointed me in the right direction.

16.3 Intended Changes

Next month, I will try to start actually coding. I want to be 100% certain of how the application will work before I start to code. I don't want to waste time ploughing ahead and then realising that I was going down the wrong route.

I realised that I need to learn more about MVVM and XAML. The MVVM is the model I will use for the application and XAML is the language used for the UI.

17.0 Supervisor Meetings

Date of Meeting: 12/01/2016

17.1 Items discussed: We went through my Requirements specification document to see if I was heading in the right direction. Manual advised me on looking into what the system needs from the users as opposed to what the user needs from the system, which I mainly concentrated in the requirements specification. I was advised to get my documentation correct in order to help me when I do start coding. I have to start working using the agile scrum process which requires me to get sections of my project done in a certain block of time, then go to my potential end users for feedback and see if there's anything that

needs to be changed, this means that by the end of the project I should have a fully functioning application that the end users are happy with. As there will be constant feedback from end users there (hopefully) will be no issues regarding the application at delivery time.

17.2 Action Items: Finish Analysis Design spec. Start coding and building the application.

18.0 Month: February

18.1 My Achievements

At the start of the month I concentrated on getting everything ready for the mid-point presentation. I had started to do a small bit of code but I realised that this wasn't good enough to show at the presentation so I just done some screen shots of the app aims to look like. The presentation went well, I received some positive feedback but I think the fact I did not have any prototype to show went against me. I just felt that there was no point in showing a small bit of code that does not work.

It was pointed out to me to consider security with regards to the database and where it will be hosted as different geographical locations have different laws. The app will hold sensitive information so it's important for me to consider this. This was something that I had never even thought of before the presentation.

18.2 My Reflection

I felt, it worked well to have the mid-point presentation. I thought that it really focused me on what needs to be done and what I have done so far that wasn't up to scratch. It pointed out exactly where I stood with regards to the project.

18.3 Intended Changes

Next month, I aim to have at least 2 or 3 pages of the app linked up with some functionality and more importantly, get the database set up and working correctly. As the app will be using the database for most of its functionality, it is vital that

this gets going asap. It's the tricky part of the project but once that is working, then a lot of the other aspects will fall into place.

19.0 Supervisor Meetings

Date of Meeting: N/A

Month: March

19.1 My Achievements

I was much happier at the end of this month than at the end of last month with regards to actual coding. I got the login and registration pages completed, they are saving the users details to the database and it is working correctly. Although I should have had this done a lot sooner, I felt happy that it finally worked. There were many issues trying to get it to work correctly, just little things but when I fixed the bugs, I was very happy. There is still a LOT of work to be done, I have a few days off work each week for the next five weeks so hopefully this time off will give me the opportunity to get the project completed with all of the functionality working that I had set out at the start of the project.

19.2 My Reflection

I was struggling to get the time to do much work on the project as there were CA's and projects to be completed for my other two modules. They took over for a few weeks and I can see now why I should be a lot further into the project than I am as there is very little spare time to work on it, I have two exams coming up very soon so there will have to be time dedicated to them too. Even though at the start of the project, we were told about time management, I seem to have totally messed this up. In a perfect world I would not have had much to do in the other modules and I could dedicate my time to the project but I should have realised earlier that this was never going to be an option. Looking back now I would have done things so much differently but I suppose hindsight is a great thing.

19.3 Intended Changes

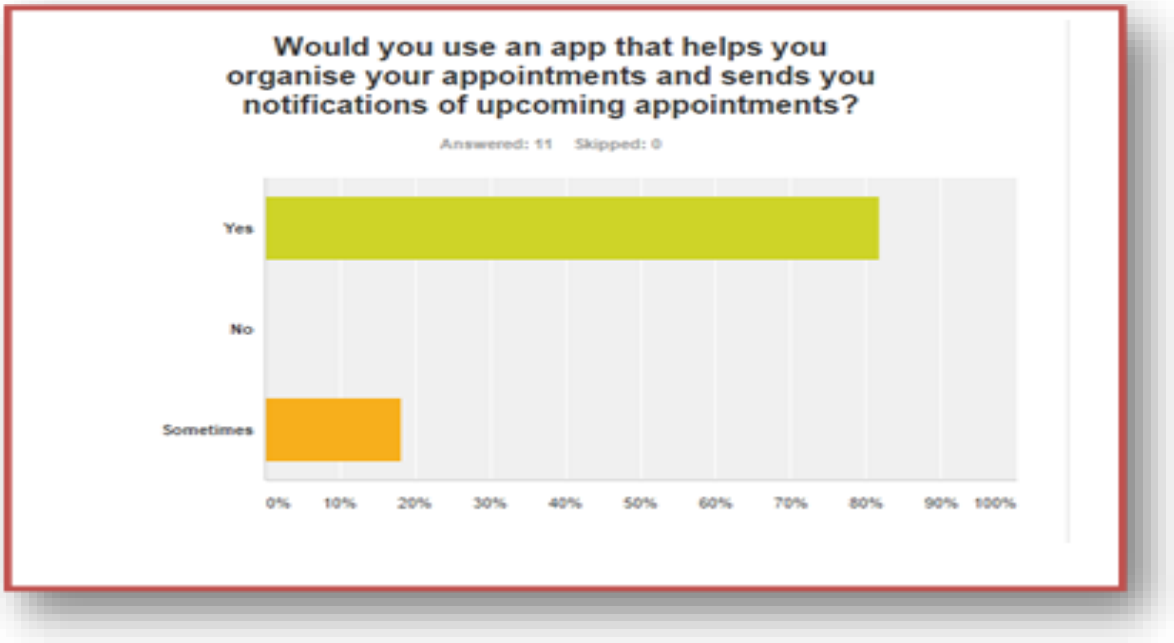
Next month, I will be completing the other pages of my app. At the moment all I am concentrating on is the code and functionality, once I have that all working correctly I will concentrate on the look and feel of the app. I just want to get it working correctly before I look at the aesthetics'.

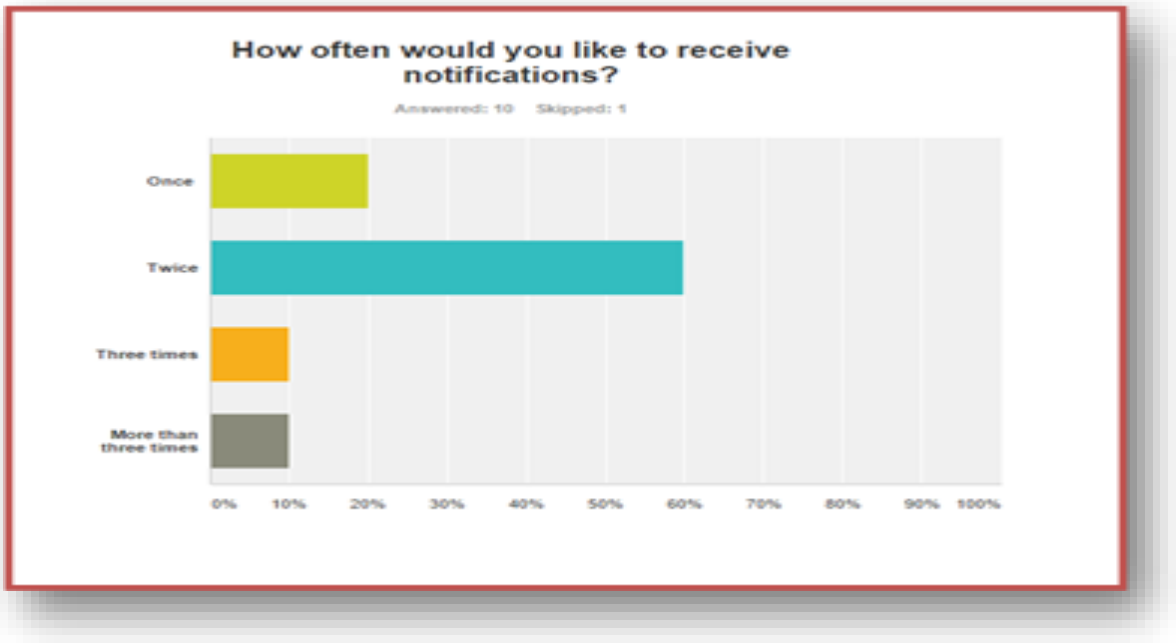
20.0 Supervisor Meetings

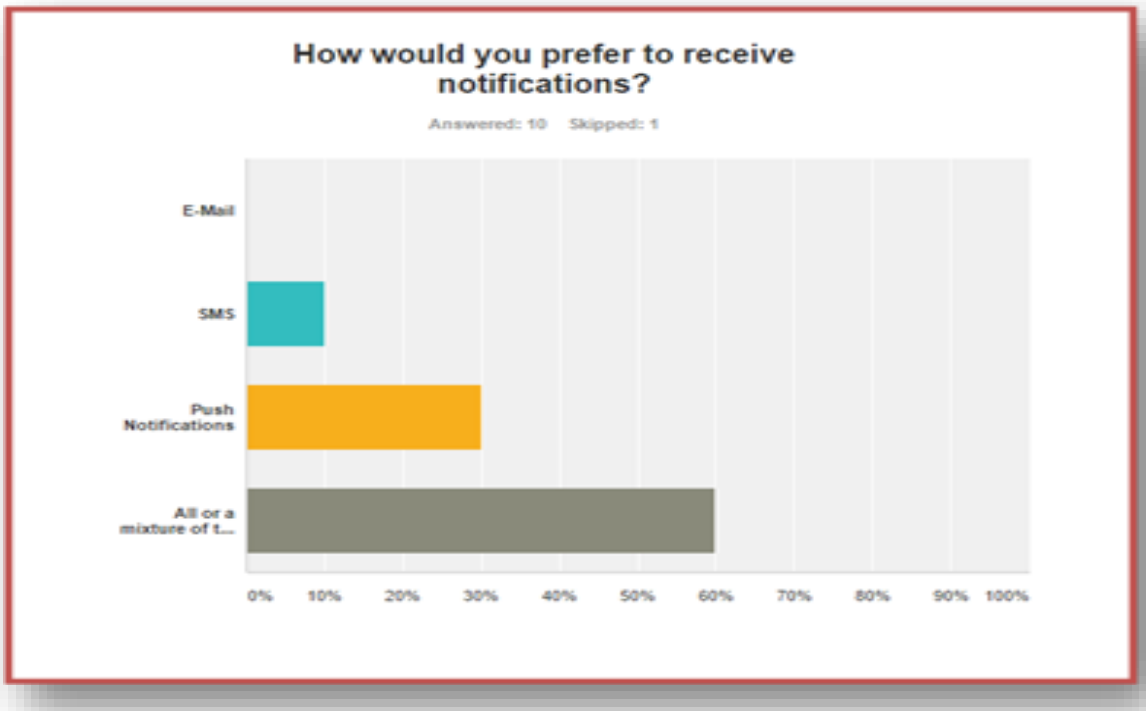
Date of Meeting: N/A

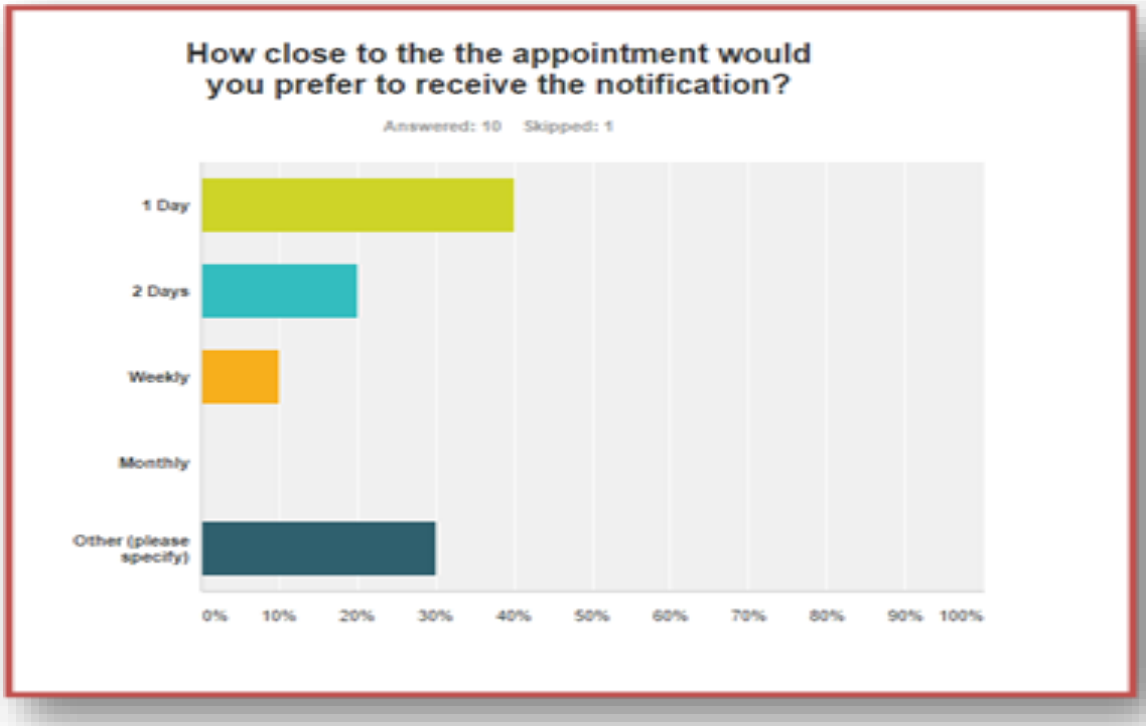
Appendix B - Other Material Used

Screen shots results of the survey conducted.

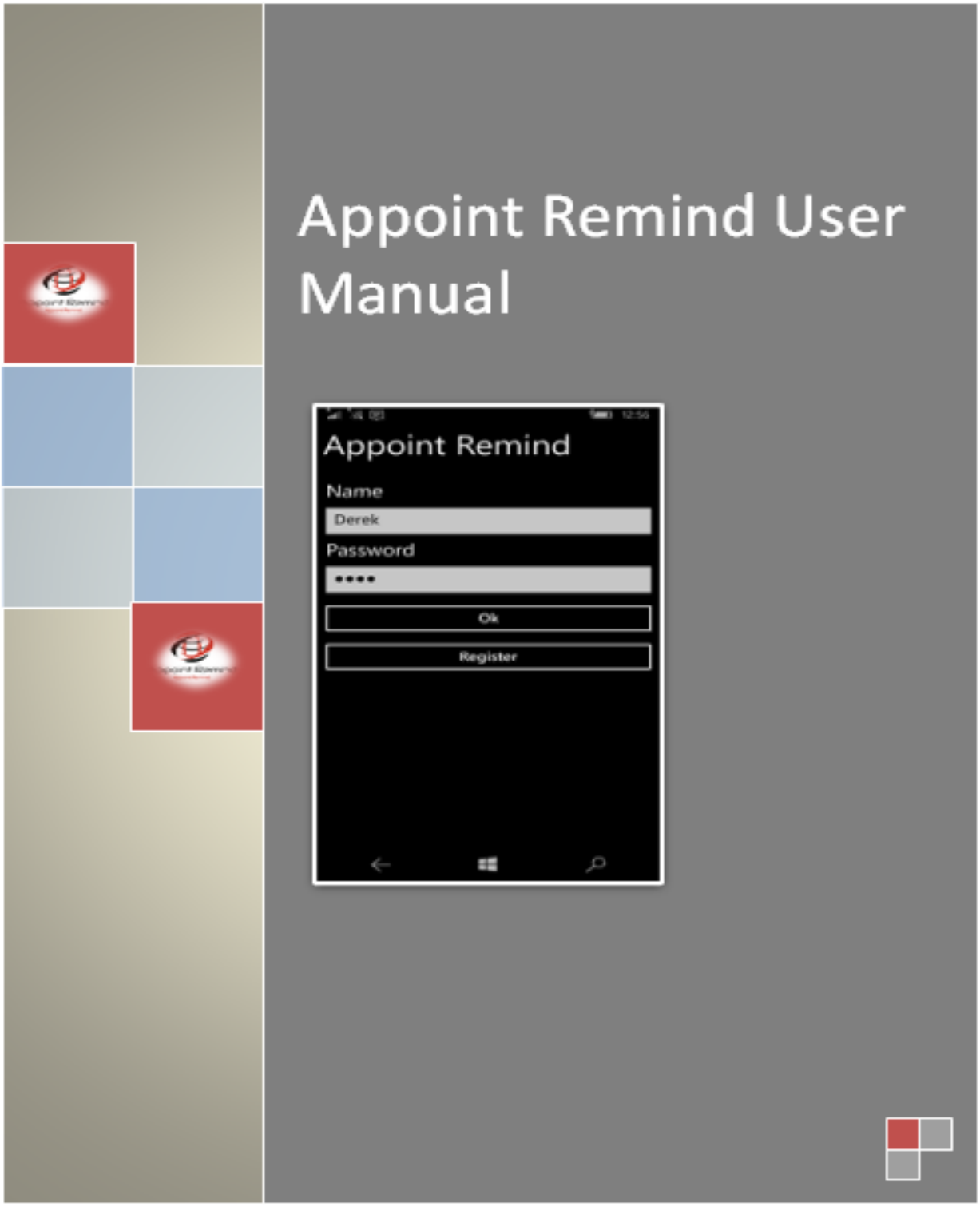





How would you prefer to receive notifications?
Answered: 10 Skipped: 1
E-Mail
SMS
Push Notifications
All or a mixture of t...
0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%


How close to the the appointment would you prefer to receive the notification?
Answered: 10 Skipped: 1
1 Day
2 Days
Weekly
Monthly
Other (please specify)
0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Appendix C


Appoint Remind User Manual
Appoint Remind
Name
Derek
Password
Ok
Register

Appoint Remind will store all your medical appointments, and notify you in advance of each appointment. With this App once you input your information, you don't have to worry about forgetting an appointment again. The App will manage the rest i.e. send you notification of your most recent appointment as many times you request. Wherever your location you will receive alerts in the format you choose.

Follow these Six Steps to access the Application, and register your details.

Step 1

Install Application from Windows Store (Link to windows store)

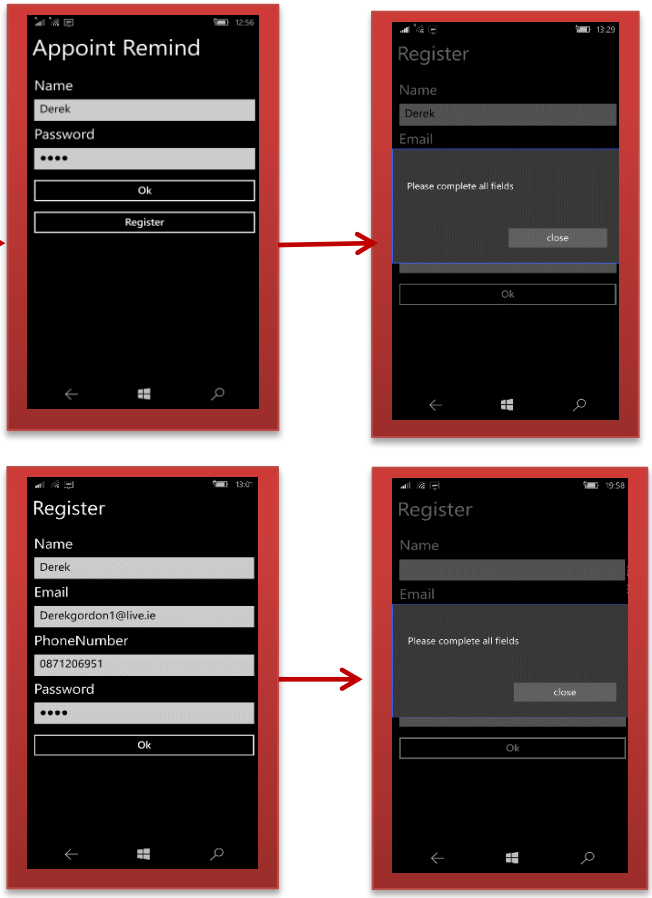
Step 2

Open Application and Register

Enter Name, E-mail, Phone number

You then generate a Password

Click OK and Register



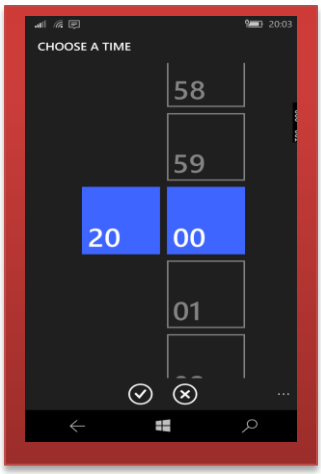
Step 3

Add Event

Enter event details i.e.

Choose a Date and

Time of Event

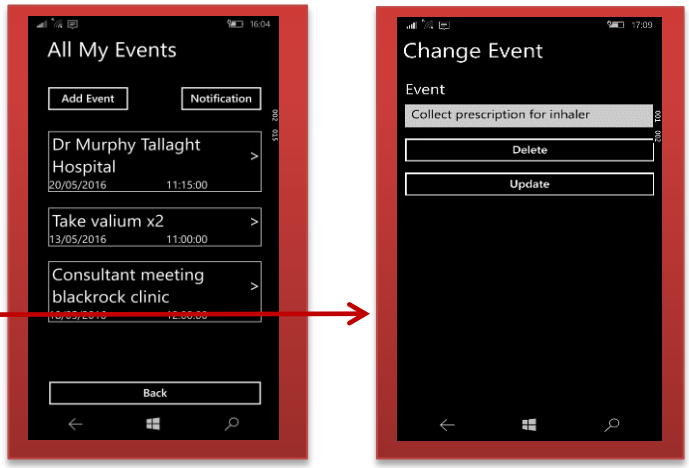


Step 4

View all Events

In this screen you can view all your events

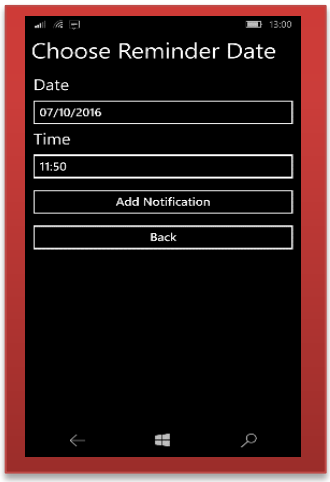
You can also update or delete an event

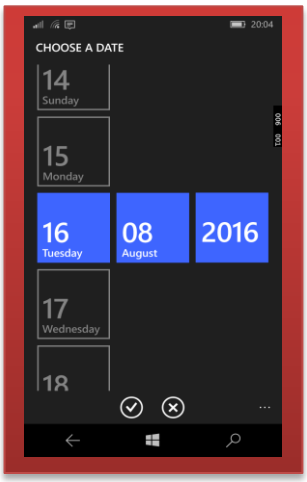


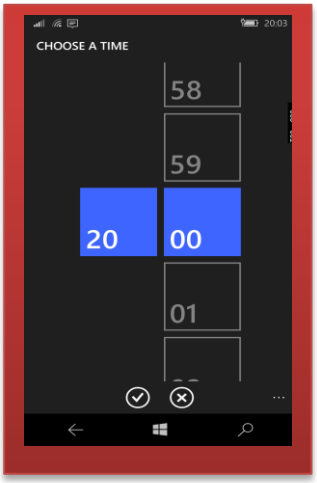
Step 5

Choose Reminder Date

Choose date and time you wish to be reminded of event







Step 6

Finally, choose type of Notification you would Like to receive

SMS

E-mail

Push notification