National College of Ireland

BSc in Business Information Systems

2015/2016

Keith Russell

12113972

X12113972@student.ncirl.ie

YourDiabetes

Technical Report

Table of Contents

Executive Summary ... 3

1 Introduction ... 4

1.1 Background ... 4

1.2 Aims ... 5

1.3 Technologies ... 5

2 System ... 6

2.1 Requirements ... 6

2.1.1 Functional requirements ... 7

2.1.2 Data requirements ... 18

2.1.3 User requirements ... 18

2.1.4 Usability requirements ... 19

2.2 Design and Architecture ... 19

2.3 Implementation ... 20

2.4 Testing ... 23

2.5 Graphical User Interface (GUI) Layout ... 23

3 Conclusions ... 27

4 Further development or research ... 28

5 References ... 29

6 Appendix ... 31

6.1 Project Proposal ... 31

6.2 Functional Requirements ... 35

6.3 Monthly Journals ... 65

6.4 Other Material Used ... 69

Executive Summary

There are a vast number of people with diabetes. Many of these people are using technology to help them monitor their condition. It has been encouraged that people exercise, log food diaries and record their blood glucose levels. However, according to Diabetes Ireland, there does not seem to be an application that combines all these together.

This report describes and evaluates the range of possible design solutions the team have generated.

Using SQL (for reporting and database updates), Visual Studio and android technologies all data can be combined into one area for ease of use for the patient. People are on the go a lot more nowadays and it would make it seamless for patients to have an application available like this to be available.

1 Introduction

This template for technical report is provided for your convenience. It should be seen as a guide rather than an obligatory form. Your individual report might require changes in terms of format or content (i.e., headings) or both.

Print on one side of the paper only (this will be the right hand side when the pages are bound).

1.1 Background

I, currently suffer from type 2 diabetes and am sure people are aware of someone that has either type 1 or type 2 diabetes. What is diabetes?

“Diabetes mellitus is a lifelong condition caused by a lack, or insufficiency of insulin. Insulin is a hormone – a substance of vital importance that is made by your pancreas. Insulin acts like a key to open the doors into your cells, letting sugar (glucose) in. In diabetes, the pancreas makes too little insulin to enable all the sugar in your blood to get into your muscle and other cells to produce energy. If sugar can’t get into the cells to be used, it builds up in the bloodstream. Therefore diabetes is characterised by high blood sugar (glucose) levels”

The goal of diabetes organisations:

“Greater focus on prevention, early detection and effective management is needed” *

If people are able to effectively manage their condition, then the better it is for them.

There is plenty of technology available to help people with various conditions so why not diabetes?

1.2 Aims

The aims of this project are

- Simple and easy to use
- Combine data from apps and devices into one
- Make it mobile

1.3 Technologies

The technology used in this application will be:

- Visual Studio 2015 and C# for programming environment (
- Xamarin for Android (Xamarin Studio Community) Version 5.10.3
- SQL for database CRUD(create, read, update and delete) functionality

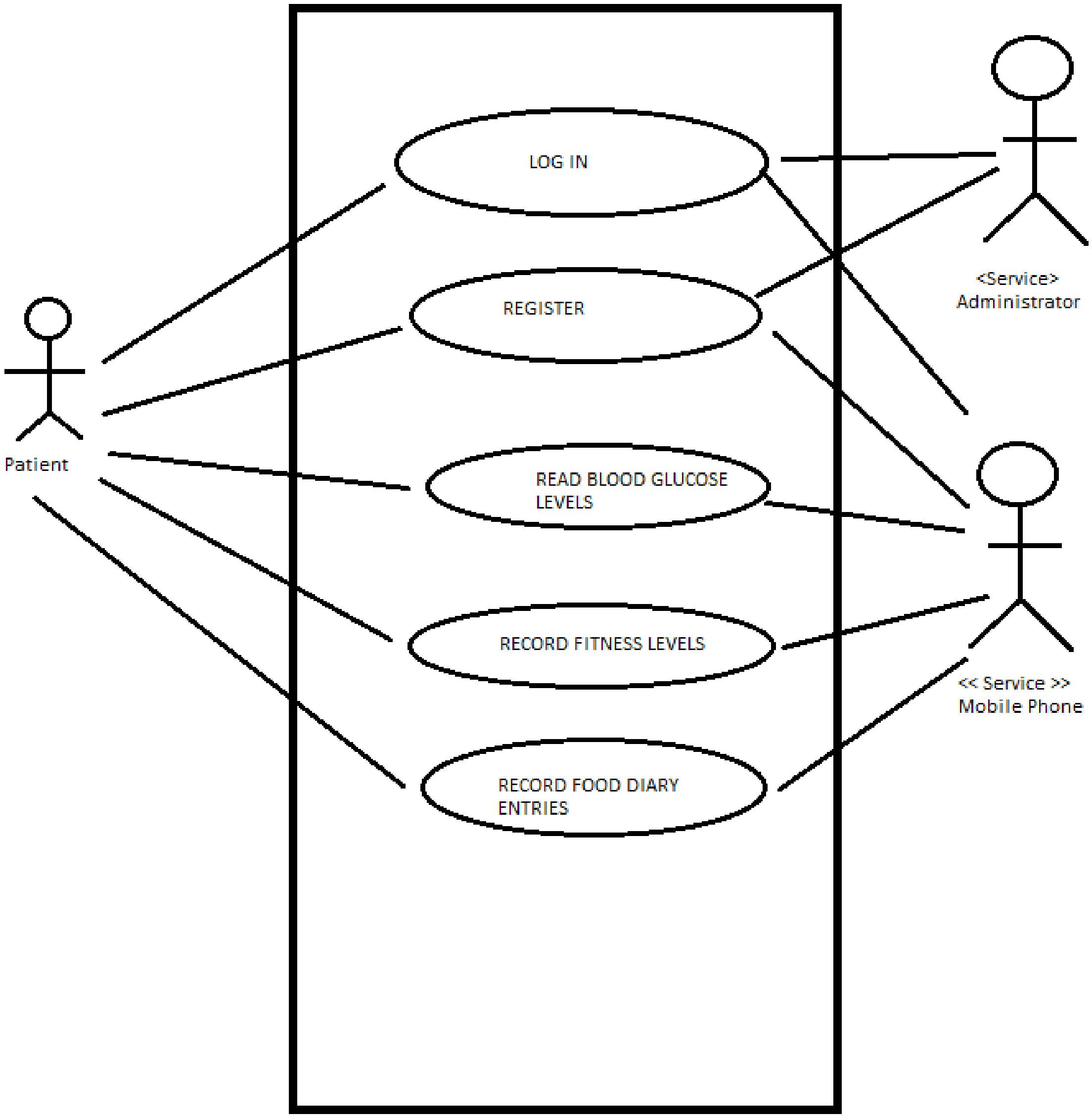
2 System

2.1 Requirements

The requirements have changed from the original due to the complexity of technology required. The new requirements now focus on a much simpler approach yet still allow for central database for fitness, blood glucose entries and food diary entries and create the basis for future development.

2.1.1 Functional requirements

2.1.1.1 Use case diagram


LOG IN
REGISTER
READ BLOOD GLUCOSE LEVELS
RECORD FITNESS LEVELS
RECORD FOOD DIARY ENTRIES
Patient
<Service> Administrator
<< Service >> Mobile Phone

2.1.1.2 Requirement 1 <Register/Signup>

2.1.1.2.1 Description & Priority

This requirement allows the user to sign up to the system and have a record on the database which will allow him/her to use that account to log a blood glucose entry, fitness exercise regime and food diary entry. They will also be able to manage online backups to the cloud. So an account is important for activation

2.1.1.2.2 Use Case

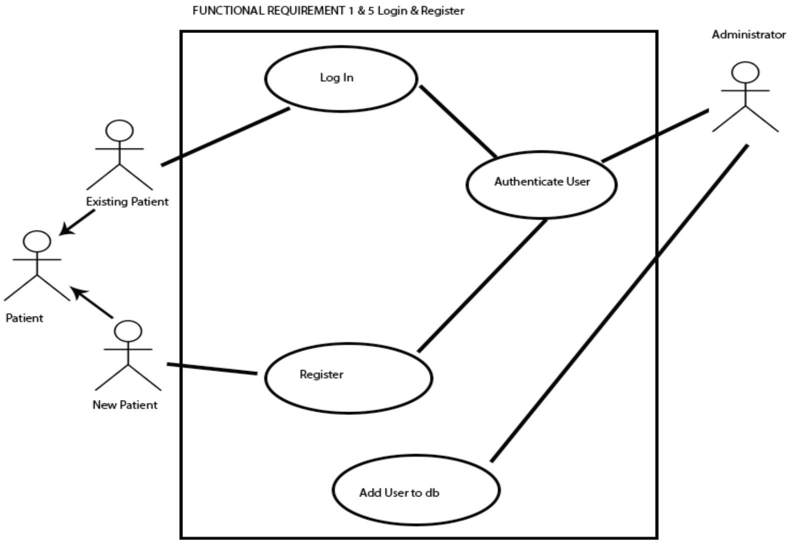
Scope

The scope of this use case is to set up a new user, make sure that they are not already on the system and create a relevant records for app usability

Description

This use case describes the approach the new user would use to register for the application

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the front/holding page to log on or register

Activation

This use case starts when a Patient clicks on the register link on the front page.

Main flow

1. The system identifies clicking of the Register user link
2. The Patient is identified as New Patient in this instance
3. The New Patient enters the registration details
4. The system notifies any registration errors E1
5. The Administrator authenticates the user
6. The Administrator adds a new record to the database

Exceptional flow

E1: Registration errors

1. The system highlights registration errors
2. The New Patient returns with correct data
3. The use case continues at position 5 of the main flow

Termination

The system presents the main menu for the user

Post condition

The system goes into a wait state

2.1.1.3 Requirement 2 <Log your blood glucose levels>

2.1.1.3.1 Description & Priority

The system should be able to get the user to enter their blood glucose levels.

2.1.1.3.2 Use Case

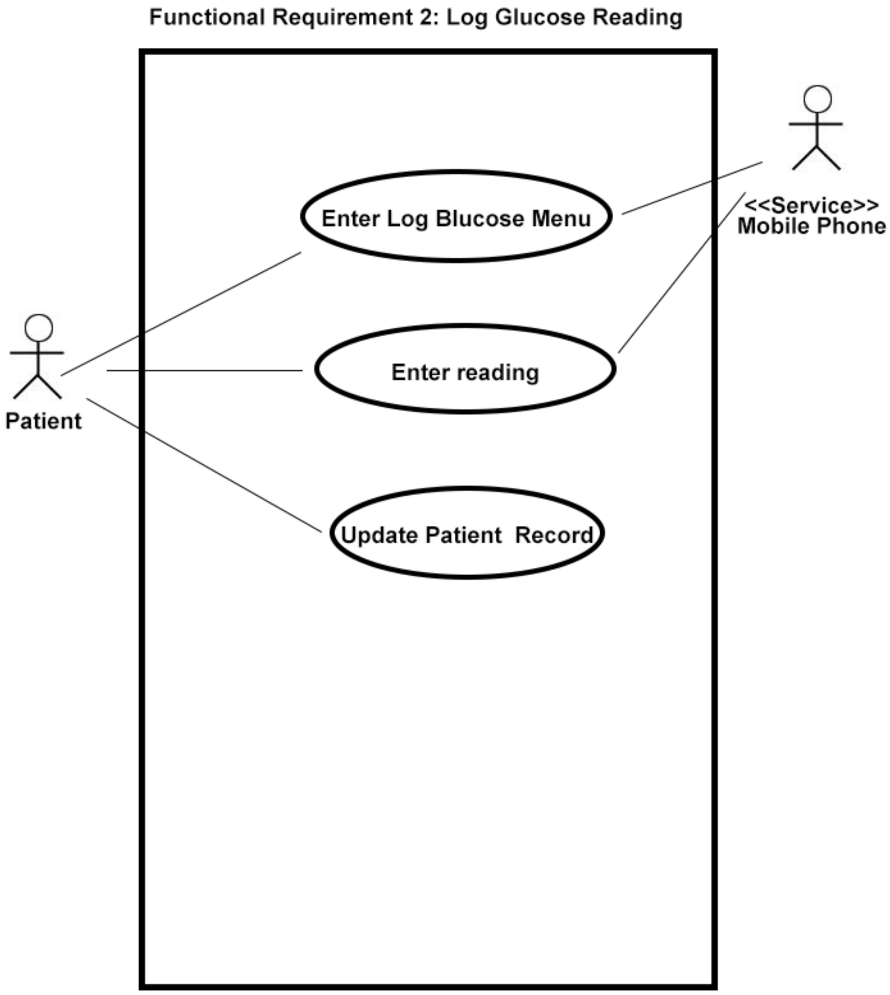
Scope

The scope of this use case is that the user is able to log their blood glucose levels at any time

Description

This use case describes the process allowing the user to enter their blood glucose levels

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> clicks on “Log Glucose Reading” on the main menu

Main flow

- The system identifies the action taken and prompts the user to enter the blood glucose data on the phone.

- The user enters data on the phone
- The system notices any data validation errors (see E1)
- The user clicks “cancel” or “back” on the phone (see A1)
- The user clicks “submit reading”

Alternate flow

A1 : <The <Patient> decides not to enter a reading>

1. The system notices that the user has not pressed the submit button
2. The system goes into post condition

Exceptional flow

E1: <Reading errors>

1. The system notices that incorrect data has been entered
2. The <Patient> returns with correct data
3. The use case continues at position 4 of the main flow

Termination

The system updates the user record with the latest reading

Post condition

The system returns to the main menu

2.1.1.4 Requirement 3<Log your fitness entries>

2.1.1.4.1 Description & Priority

The system should be able to get the user to enter their fitness levels.

2.1.1.4.2 Use Case

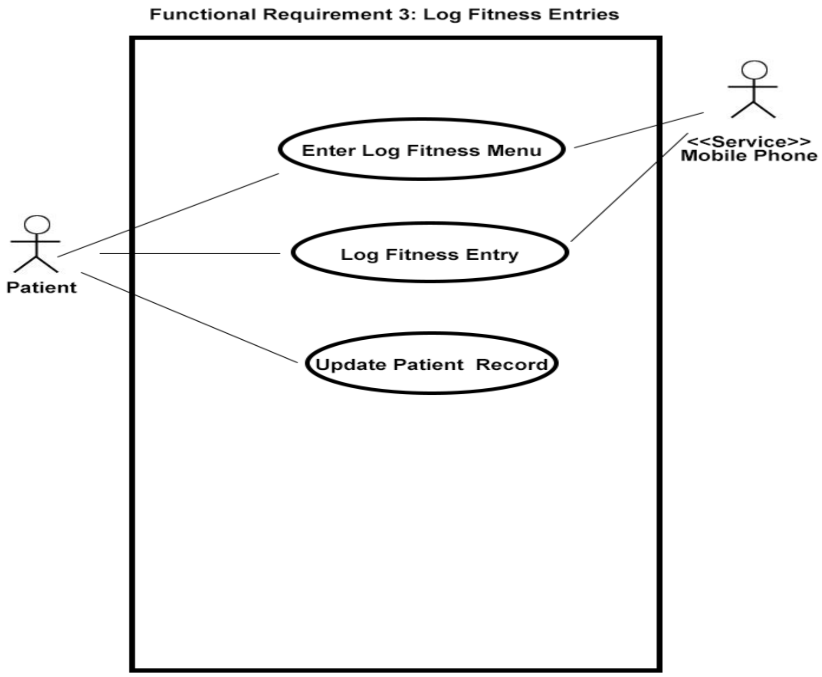
Scope

The scope of this use case is that the user is able to log their fitness levels at any time

Description

This use case describes the process allowing the user to enter fitness levels

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> clicks on “Log Fitness Entry” on the main menu

Main flow

1 The system identifies the action taken and prompts the user to enter the fitness data on the phone.

2 The user enters data on the phone

3 The system notices any data validation errors (see E1)

4 The user clicks “cancel” or “back” on the phone (see A1)

5 The user clicks “Submit Entry”

Alternate flow

A1 : <The <Patient> decides not to enter a fitness entry>

1. The system notices that the user has not pressed the submit button
2. The system goes into post condition

Exceptional flow

E1: <Reading errors>

1. The system notices that incorrect data has been entered
2. The <Patient> returns with correct data
3. The use case continues at position 4 of the main flow

Termination

The system updates the user record with the latest fitness entry

Post condition

The system returns to the main menu

2.1.1.5 Requirement 4<Log your food diary entries>

2.1.1.5.1 Description & Priority

The system should be able to get the user to enter their fitness levels.

2.1.1.5.2 Use Case

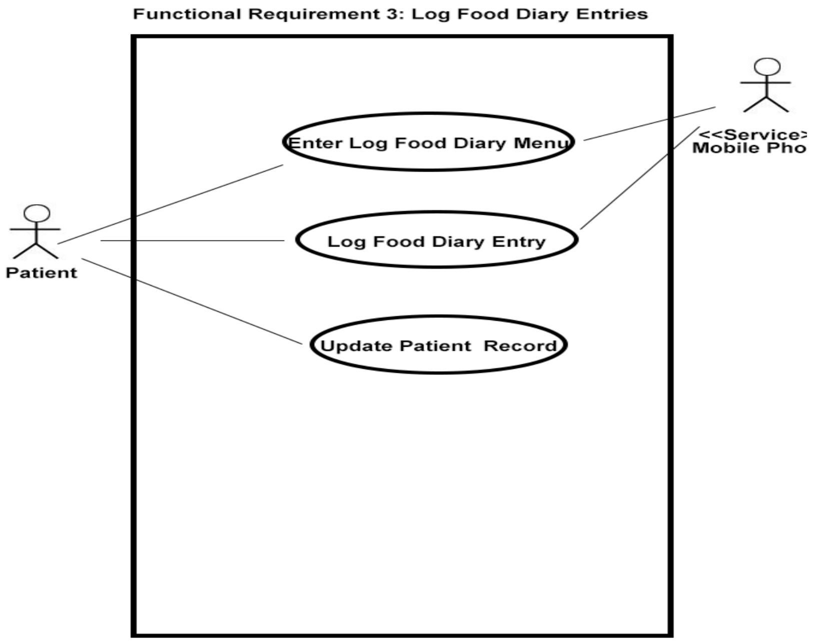
Scope

The scope of this use case is that the user is able to log their food diary levels at any time

Description

This use case describes the process allowing the user to enter food diary levels

Use Case Diagram


Functional Requirement 3: Log Food Diary Entries
Enter Log Food Diary Menu
Log Food Diary Entry
Update Patient Record
Patient
<<Service>
Mobile Pho

Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> clicks on “Log Food Diary Entry” on the main menu

Main flow

1. The system identifies the action taken and prompts the user to enter the fitness data on the phone.
2. The user enters data on the phone

3. The system notices any data validation errors (see E1)
4. The user clicks “cancel” or “back” on the phone (see A1)
5. The user clicks “Submit Entry”

Alternate flow

A1 : <The <Patient> decides not to enter a fitness entry>

1. The system notices that the user has not pressed the submit button
2. The system goes into post condition

Exceptional flow

E1: <Reading errors>

1. The system notices that incorrect data has been entered
2. The <Patient> returns with correct data
3. The use case continues at position 4 of the main flow

Termination

The system updates the user record with the latest fitness entry

Post condition

The system returns to the main menu

2.1.1.6 Requirement 5<Log into the system>

2.1.1.6.1 Description & Priority

User must be able to log in with their credentials. The idea will be that it will be difficult to retrieve data because of the sensitivity.

2.1.1.6.2 Use Case

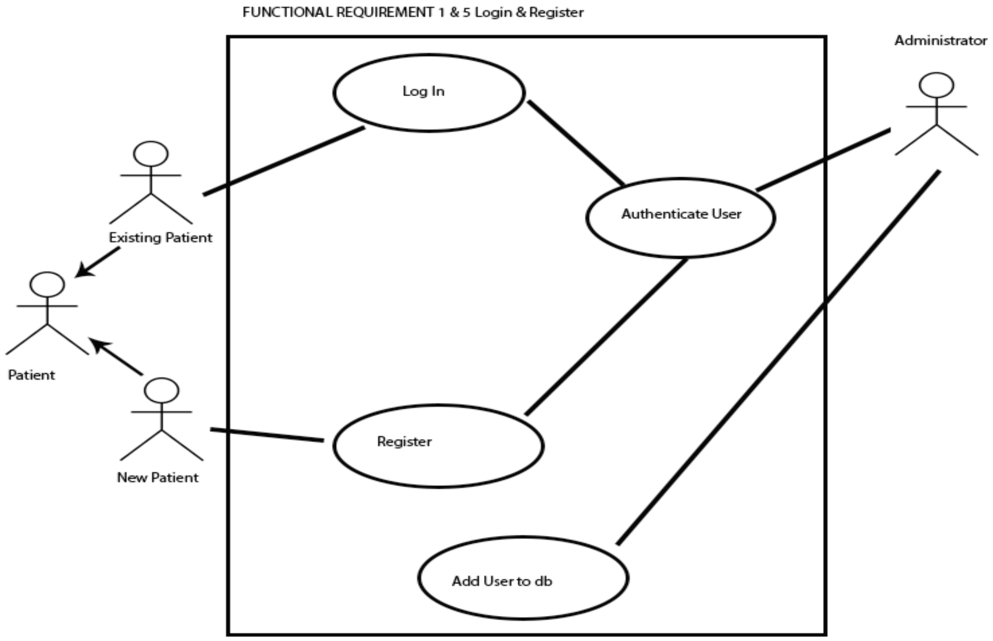
Scope

The scope of this use case is to get users to log in and be authenticated

Description

This use case describes the process in which users log in to the system for use of the application

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode with the <Patient> is at the Front Page of the UI. The prompts here are Log In and Register

Activation

This use case starts when a <Patient> enters their user name and password, and then clicks on Log-in

Main flow

1. The system identifies the log in button pressed
2. The system sends the user a verification code via text
3. The <Patient> enters the verification code and accesses the main menu

4. The system is unable to verify the <Patient> credentials (see A1)
5. The system is unable to find the <Patient> on the system

Alternate flow

A1: <unable to verify user credentials>

4. The system notifies the user that the credentials are incorrect
5. The <Patient> provides an email address to change the password
6. The use case continues at position 1 of the main flow

A2 : <no account has been made for this user>

1 The system finds out the user does not exist in the database

2. The user is prompted that this is the case

3. <Patient> prompted to register (see requirement 1)

Termination

The system returns to the login screen

Post condition

The system goes to the main menu for the application

2.1.2 Data requirements

Data entered must be valid for the database to store.

2.1.3 User requirements

After discussion with people with sufferers of Diabetes, people have come up with the following requirements they need

- Easy blood meter reading entry
- Access to food diary to assist with dietary requirements
- Integration with fitness regime app
- Inform user when to check type (alarm)
- Nutritionist links and information
- Message board

- Forums

From the project scope and constraints within it, the users will be able to log their blood glucose, fitness and food diary data

Further versions will assist with other requirements.

2.1.4 Usability requirements

Must have a simple user interface at all times (big buttons and easy to use on android phones)

2.2 Design and Architecture

The design of the application will use layouts in Visual Studio.

Depending on the layout there will exist buttons, labels (textView in Visual C#), text boxes (editText in Visual C#) and Data Views (List Views)

Each of these layouts will be represented by activities.

These activities will reference objects in the relevant layouts to “act out” the implementation of the functions

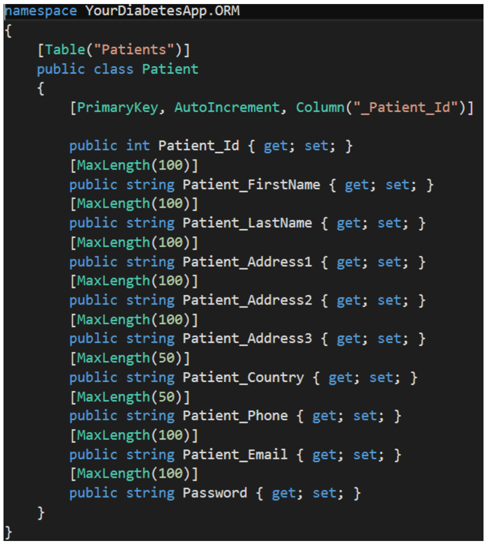
Each android device will contain a local SQLite database in which CRUD (Create Read Update Delete) operations will be made

Xamarin technology will be used to help with the functionality and portability for Android Mobile.

2.3 Implementation

Each SQL table instance (Patient, Reading, Fitness and FoodDiary) were created so that they can be referenced as and when required to do CRUD (Create Read Update and Delete) functions.

Below is a snippet of the Patient Table



```
namespace YourDiabetesApp.ORM
{
    [Table("Patients")]
    public class Patient
    {
        [PrimaryKey, AutoIncrement, Column("_Patient_Id")]

        public int Patient_Id { get; set; }
        [MaxLength(100)]
        public string Patient_FirstName { get; set; }
        [MaxLength(100)]
        public string Patient_LastName { get; set; }
        [MaxLength(100)]
        public string Patient_Address1 { get; set; }
        [MaxLength(100)]
        public string Patient_Address2 { get; set; }
        [MaxLength(100)]
        public string Patient_Address3 { get; set; }
        [MaxLength(50)]
        public string Patient_Country { get; set; }
        [MaxLength(50)]
        public string Patient_Phone { get; set; }
        [MaxLength(100)]
        public string Patient_Email { get; set; }
        [MaxLength(100)]
        public string Password { get; set; }
    }
}
```

The application also has BloodGlucoseApp, FoodDiaryApp and FitnessApp tables so that they can be used for future use when the app communicates with devices and third party applications for data.

The next stage is to use functions to make CRUD requests.

This is done in SQLITE.

The nuget package SQLITE.net (by Frank Kruger) has been installed into the application.

Functions

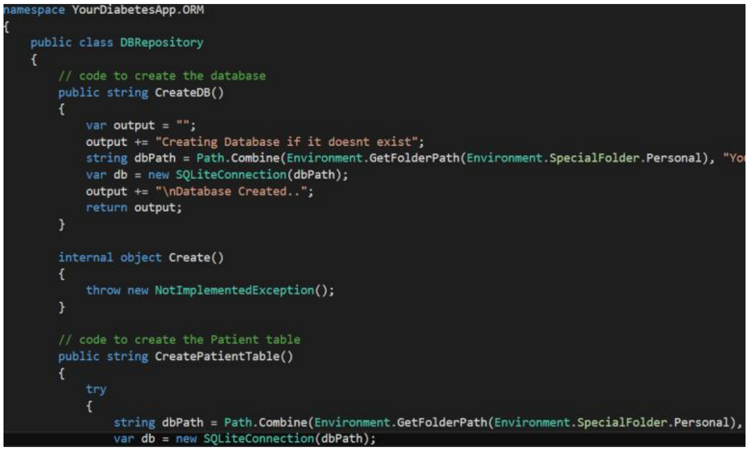
Code needed to be set up to

- Create the database
- Create the tables(as mentioned)
- Read from the tables
- Update the data to the table
- Delete relevant table data

All these functions are in the DBRepository file in the ORM directory

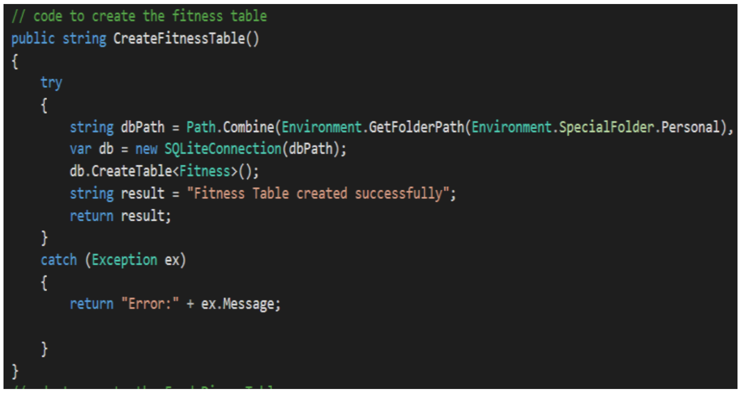
Some code snippets:

(Create the database)



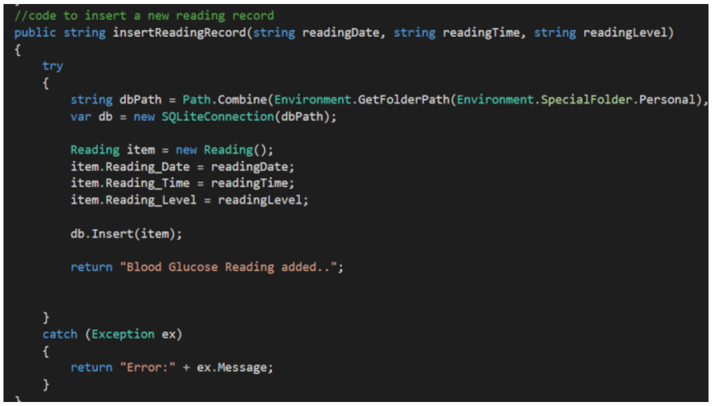
```
namespace YourDiabetesApp.ORM
{
    public class DBRepository
    {
        // code to create the database
        public string CreateDB()
        {
            var output = "";
            output += "Creating Database if it doesnt exist";
            string dbPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), "Yo
            var db = new SQLiteConnection(dbPath);
            output += "\nDatabase Created..";
            return output;
        }

        internal object Create()
        {
            throw new NotImplementedException();
        }

        // code to create the Patient table
        public string CreatePatientTable()
        {
            try
            {
                string dbPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal),
                var db = new SQLiteConnection(dbPath);
```

(Create a table)



```
// code to create the fitness table
public string CreateFitnessTable()
{
    try
    {
        string dbPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal),
        var db = new SQLiteConnection(dbPath);
        db.CreateTable<Fitness>();
        string result = "Fitness Table created successfully";
        return result;
    }
    catch (Exception ex)
    {
        return "Error:" + ex.Message;

    }
}
```

(Insert a new record)

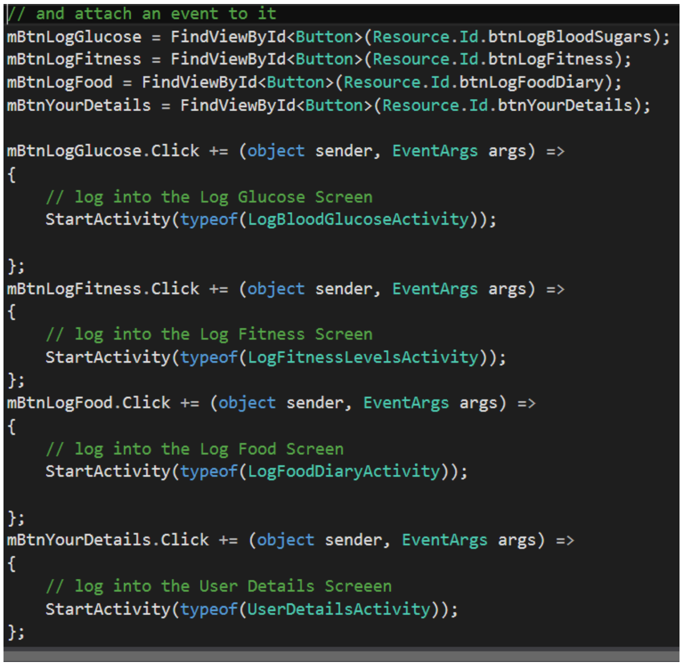


```
//code to insert a new reading record
public string insertReadingRecord(string readingDate, string readingTime, string readingLevel)
{
    try
    {
        string dbPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal),
        var db = new SQLiteConnection(dbPath);

        Reading item = new Reading();
        item.Reading_Date = readingDate;
        item.Reading_Time = readingTime;
        item.Reading_Level = readingLevel;

        db.Insert(item);

        return "Blood Glucose Reading added..";


    }
    catch (Exception ex)
    {
        return "Error:" + ex.Message;
    }
```

Each Screen is called a Layout in Visual Studio C# and each have to be associated with an activity so that certain actions can be taken when a user does something on that screen.

In this project each CRUD activity has to be set up so the user can perform the tasks they require.

Here is code for users to start each activity for all the buttons in the main menu:



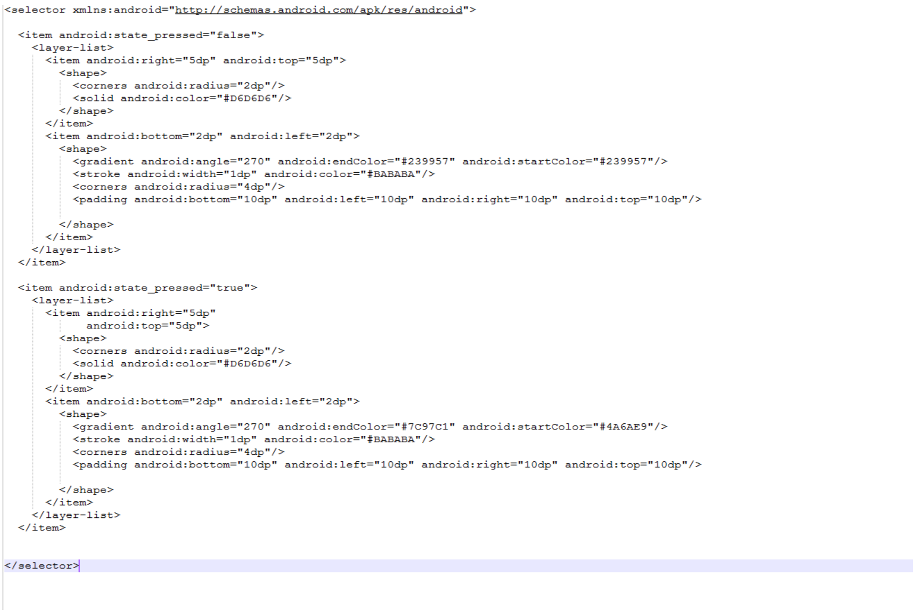
```
// and attach an event to it
mBtnLogGlucose = FindViewById<Button>(Resource.Id.btnLogBloodSugars);
mBtnLogFitness = FindViewById<Button>(Resource.Id.btnLogFitness);
mBtnLogFood = FindViewById<Button>(Resource.Id.btnLogFoodDiary);
mBtnYourDetails = FindViewById<Button>(Resource.Id.btnYourDetails);

mBtnLogGlucose.Click += (object sender, EventArgs args) =>
{
    // log into the Log Glucose Screen
    StartActivity(typeof(LogBloodGlucoseActivity));

};
mBtnLogFitness.Click += (object sender, EventArgs args) =>
{
    // log into the Log Fitness Screen
    StartActivity(typeof(LogFitnessLevelsActivity));
};
mBtnLogFood.Click += (object sender, EventArgs args) =>
{
    // log into the Log Food Screen
    StartActivity(typeof(LogFoodDiaryActivity));

};
mBtnYourDetails.Click += (object sender, EventArgs args) =>
{
    // log into the User Details Screeen
    StartActivity(typeof(UserDetailsActivity));
};
```

And for the LogBloodGlucoseActivity to display the relevant layout, it needs this:

```
SetContentView(Resource.Layout.LogBloodGlucoseLayout);
```

Buttons:

Each button has a XML reference so that it can act and display a certain way when the user clicks on it.



```
<selector xmlns:android="http://schemas.android.com/apk/res/android">

  <item android:state_pressed="false">
    <layer-list>
      <item android:right="5dp" android:top="5dp">
        <shape>
          <corners android:radius="2dp"/>
          <solid android:color="#D6D6D6"/>
        </shape>
      </item>
      <item android:bottom="2dp" android:left="2dp">
        <shape>
          <gradient android:angle="270" android:endColor="#239957" android:startColor="#239957"/>
          <stroke android:width="1dp" android:color="#BABABA"/>
          <corners android:radius="4dp"/>
          <padding android:bottom="10dp" android:left="10dp" android:right="10dp" android:top="10dp"/>

        </shape>
      </item>
    </layer-list>
  </item>

  <item android:state_pressed="true">
    <layer-list>
      <item android:right="5dp"
            android:top="5dp">
        <shape>
          <corners android:radius="2dp"/>
          <solid android:color="#D6D6D6"/>
        </shape>
      </item>
      <item android:bottom="2dp" android:left="2dp">
        <shape>
          <gradient android:angle="270" android:endColor="#7C97C1" android:startColor="#4A6AE9"/>
          <stroke android:width="1dp" android:color="#BABABA"/>
          <corners android:radius="4dp"/>
          <padding android:bottom="10dp" android:left="10dp" android:right="10dp" android:top="10dp"/>

        </shape>
      </item>
    </layer-list>
  </item>


</selector>
```

2.4 Testing

Initially the development of the application was done on a Samsung Galaxy S5, but it also works on an S4. These were the only 2 devices available. Minimum android version is 4.03

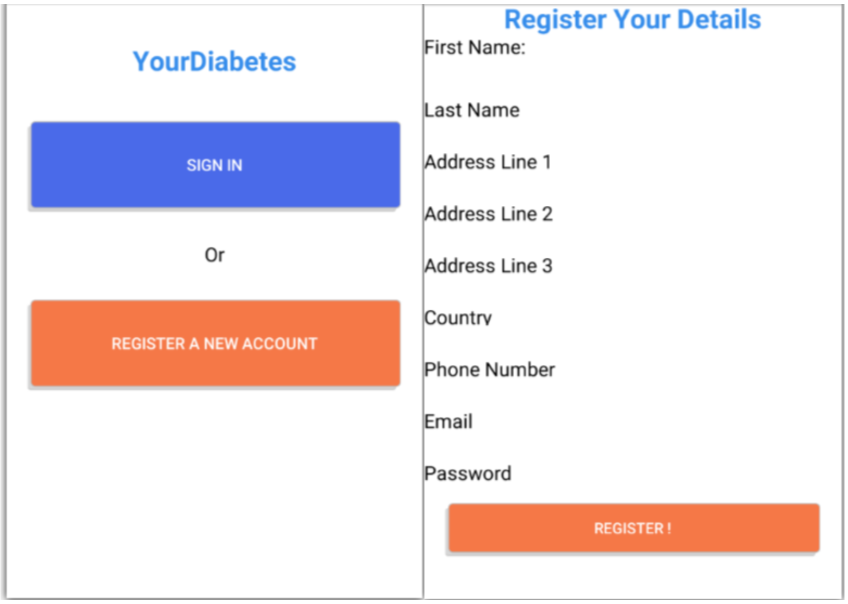
Testing was also done manually as the app was developed e.g. data types, SQL CRUD activities

Comments are available within the code to show this.

2.5 Graphical User Interface (GUI) Layout

Start-up Screen:

This allows the user to either login or register



YourDiabetes

Register Your Details

First Name:

Last Name

Address Line 1

Address Line 2

Address Line 3

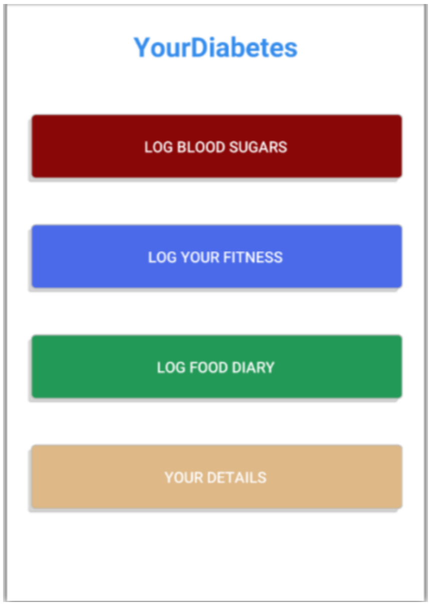
Country

Phone Number

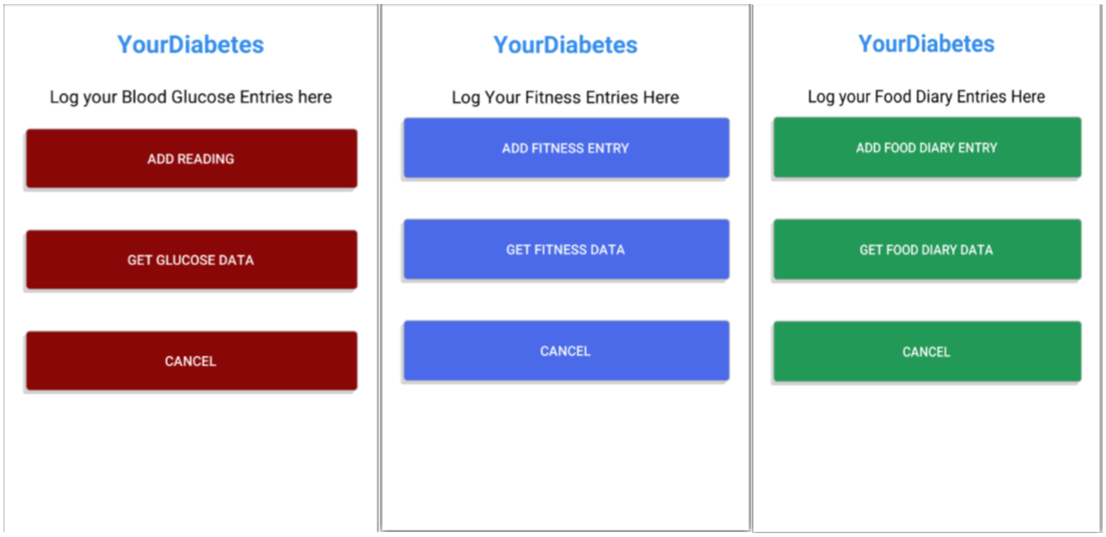
Email

Password

Below is the main menu. The colour association is red for blood, blue for fitness, and green for food which hopes to guide people toward the action they need to take on the application.



YourDiabetes



YourDiabetes

Log your Blood Glucose Entries here

YourDiabetes

Log Your Fitness Entries Here

YourDiabetes

Log your Food Diary Entries Here

Above are the relative sub – menus. The colour co-ordinations are aimed to help where the user is. Clarity and ease of use is a goal.

3 Conclusions

Opportunities of this project include:

- One stop shop for food diaries/fitness and blood glucose entries
- Once developed, patients can see how they are doing in a graphical format that is easy to understand.
- If they are already using an app for fitness or food diary, then it can be added to the application as their preferred application rather than use one set by the app.
- If they are already using a Bluetooth device then the technology should be made available for people to log their entries.
- Consultants and doctors can, in time, view monthly/bi-annual/annual data and better assist patients with the condition of diabetes

Current limitations of this project include:

- Developed for android only.
- No backup available for users

Learning curve:

I had no previous experience in creating a mobile application but believed that this app would be beneficial because of its commercial potential, particularly given the amount of people that have a smartphone and they would not have to log onto a website on a laptop/desktop.

Blockers encountered:

Free student licence for Visual Studio 2015 and Xamarin were difficult to configure at first. This stopped me from coding initially

Then, once development began, I discovered that some Xamarin issues regarding the size of the project became an issue. After threading through Google, I came to no solution and found out that upgrading to the latest version of Xamarin for Android solved the issue

Another issue that arose was that the Resource file in Visual Studio did not recognise new layouts. After searching the web, I discovered that by creating a new project and copying the code across from the previous one solved the problem.

4 Further development or research

With more resources, where could the results of this project lead to?

This application could go cross-platform i.e. iPhone and Windows

With more knowledge of being able to securely access existing web applications, this application could also retrieve the existing data from these applications and log it locally into “YourDiabetes”.

Many applications and devices can be linked to this project which will give the user more choice to track their blood glucose levels, fitness and food diary entries.

5 References

*http://http://www.publichealth.ie/press-release/roi-press-release-many-people-are-living-diabetes-often-undiagnosed-and-number

Interacting with database in Android App using C# with Xamarin https://youtu.be/mAw0Fa92qNE

http://www.stackoverflow.com

http://www.codeproject.com/

Xamarin tutorials

https://developer.xamarin.com/recipes/android/fundamentals/intent/open_a_webpage_in_the_browser_application/

Displaying the data in a list view

https://www.youtube.com/watch?v=OHvY1DUxzfo&index=3&list=PLCuRg51-gw5VqYchUekCqxUS9hEZkDf6l

Creating a login and register

https://www.youtube.com/watch?v=tf2O_t-ayJ8&list=PLCuRg51-gw5VqYchUekCqxUS9hEZkDf6l&index=6

https://www.youtube.com/watch?v=RBQeTUNu_Kk&list=PLCuRg51-gw5VqYchUekCqxUS9hEZkDf6l&index=7

https://www.youtube.com/watch?v=j_0s9IslwYs&list=PLCuRg51-gw5VqYchUekCqxUS9hEZkDf6l&index=8

SQLite 1-m relationships

https://bitbucket.org/twincoders/sqlite-net-extensions

6 Appendix

6.1 Project Proposal

Project Proposal

YourDiabetes

Keith Russell, 12113972, keith.russell@student.ncirl.ie

BSc (Hons) in Business Information Systems

3rd February 2016

6.1.1 Objectives

Develop an online application that will integrate the following

- Blood Monitoring
- Food Diaries
- Exercise regime (Map my fitness)
- Inform user when to check bloods (based on type)
- Reminder for injections (alarm)
- Nutritionist links and info
- Message board
- Forums

6.1.2 Background

There are an increasing number of people in Ireland being diagnosed with Diabetes, particularly type 2.

“By 2020 the number of adults aged 45+ years with diabetes (diagnosed and undiagnosed) is expected to rise to more than 175,000 (9.1%).
A lot of people do not know they are diagnosed, this is due to irregular check-ups with their doctor.” -http://www.publichealth.ie/press-release/roi-press-release-many-people-are-living-diabetes-often-undiagnosed-and-number

When diagnosed, patients are referred to clinics supported by the HSE with free eye-checks and tracking units available on the long term illness card. Patients can then track their progress by regularly taking their blood samples and noting their blood levels down in a journal. Every year the patient can then speak with a referred consultant. However, when the consultant is spoken to there is a lot of “word of mouth” either by referral from the doctor or how the patient feels they are getting on.

I have type two diabetes and am in contact with my consultant every year, along with regular check-ups with my doctor and other diabetic test centres (eye etc.).

Patients are advised to manage their health and lifestyle themselves to ensure a better quality of life and possibly reduce the level of the condition.

This has spurred me on to develop a project that will use technology to enhance the management of diabetes for both type 1 and type 2

6.1.3 Technical Approach

Research - look at competition and see where similarities lie. Speak with people that have diabetes and get them to travel the development/test journey with me giving input where applicable

Literature review - Learn to implement the project using C# for mobile. Discover articles that relate to what is going on that affect diabetic patients so that the focus is on the end users.

Requirements capture - Due to the scale of the project it is important to define what a “requirement” is and what is a “nice to have”

As of writing Requirements are

- Blood Readings (via Bluetooth)
- Food diaries app
- Exercise regime app
- Account login for all apps

Nice to have:

- Alarm clock to tell you have insulin to inject (type one)
- Form for discussion
- activity tracker in conjunction with blood levels to see how you’re doing

Implementation -

- Discover how to get one account for all apps.
- Use Bluetooth to read bloods
- Integrate good food app that is easy to use
- Integrate MapMyFitness for exercise regime.

6.1.4 Special resources required

- Android phone
- Diabetes reading meter with Bluetooth enabled

6.1.5 Project Plan

Not available

6.1.6 Technical Details

The project will be developed using C# on Visual Studio Platform with Xamarin Studio for Android. Mashup technology will help to integrate data from other apps. SQL will be used to create, read, update and delete data to and from the database. Information will be backed up to the cloud

6.1.7 Evaluation

People will test this application that are already using diabetes applications to help them with their condition to see how easy it is to add update and retrieve data as required

6.2 Requirements Specification

HDSSD

Requirements Specification (RS)

YourDiabetes

Keith Russell

[03/02/16]

Requirements Specification (RS)

Document Control

Revision History

14/10/2015	1	Create	AB	X	X
21/10/2005	2	Update	CD		

Distribution List

Eamon Nolan	Lecturer	
Eugene McLaughlin	Lecturer	
Paul Hayes	Supervisor	

Related Documents

Title of Use Case Model	
Title of Use Case Description	

Table of Contents

1 Introduction

1.1 Purpose

1.2 Project Scope

1.3 Definitions, Acronyms, and Abbreviations

2 User Requirements Definition

3 Requirements Specification

3.1 Functional requirements

3.1.1 Use Case Diagram

3.1.2 Requirement 1 <name of requirement in a few words>

3.1.3 Requirement 2 <name of requirement in a few words>

3.2 Non-Functional Requirements

3.2.1 Performance/Response time requirement

3.2.2 Availability requirement

3.2.3 Recover requirement

3.2.4 Robustness requirement

3.2.5 Security requirement

3.2.6 Reliability requirement

3.2.7 Maintainability requirement

3.2.8 Portability requirement

3.2.9 Extendibility requirement

3.2.10 Reusability requirement

3.2.11 Resource utilization requirement

4 GUI

5 System Architecture

6 System evolution

6.2.1 Introduction

6.2.1.1 Purpose

The purpose of this document is to set out the requirements for the development of YourDiabetes.

The intended customers are people that have the diabetes condition and require a better management plan to help them control the condition

6.2.1.2 Project Scope

The scope of the project is to develop a mobile application suitable for android. The system shall have a database that will create a user account for use with MapMyFitness and MyFitnessPal. The data will be backed up on a cloud periodically.

I have been in discussion with people from the Diabetes in Ireland Facebook group which have assisted me in creating the scope.

Therefore it has been agreed that User Login/Registration, integration with MapMyFitness and MyFitnessPal, Blood Tracking via Bluetooth and Cloud Backup will be the major functions of this

Constraints include:

Only one Bluetooth device will be used

Simple UI for version 1

6.2.1.3 Definitions, Acronyms, and Abbreviations

MMF = MapMyFitness

MFP = MyFitnessPal

DB = Database

CRUD = Create, Read, Update, Delete

6.2.2 User Requirements Definition

After discussion with people with sufferers of Diabetes, people have come up with the following requirements they need

Easy blood meter reading entry

Access to food diary to assist with dietary requirements

Integration with fitness regime app

Inform user when to check type (alarm)

Nutritionist links and information

Message board

Forums

From the project scope and constraints within it, the users will be able to use their Bluetooth device for glucose reading, integrate with a sports app and food diary and back the data up to a cloud.

6.2.3 Requirements Specification

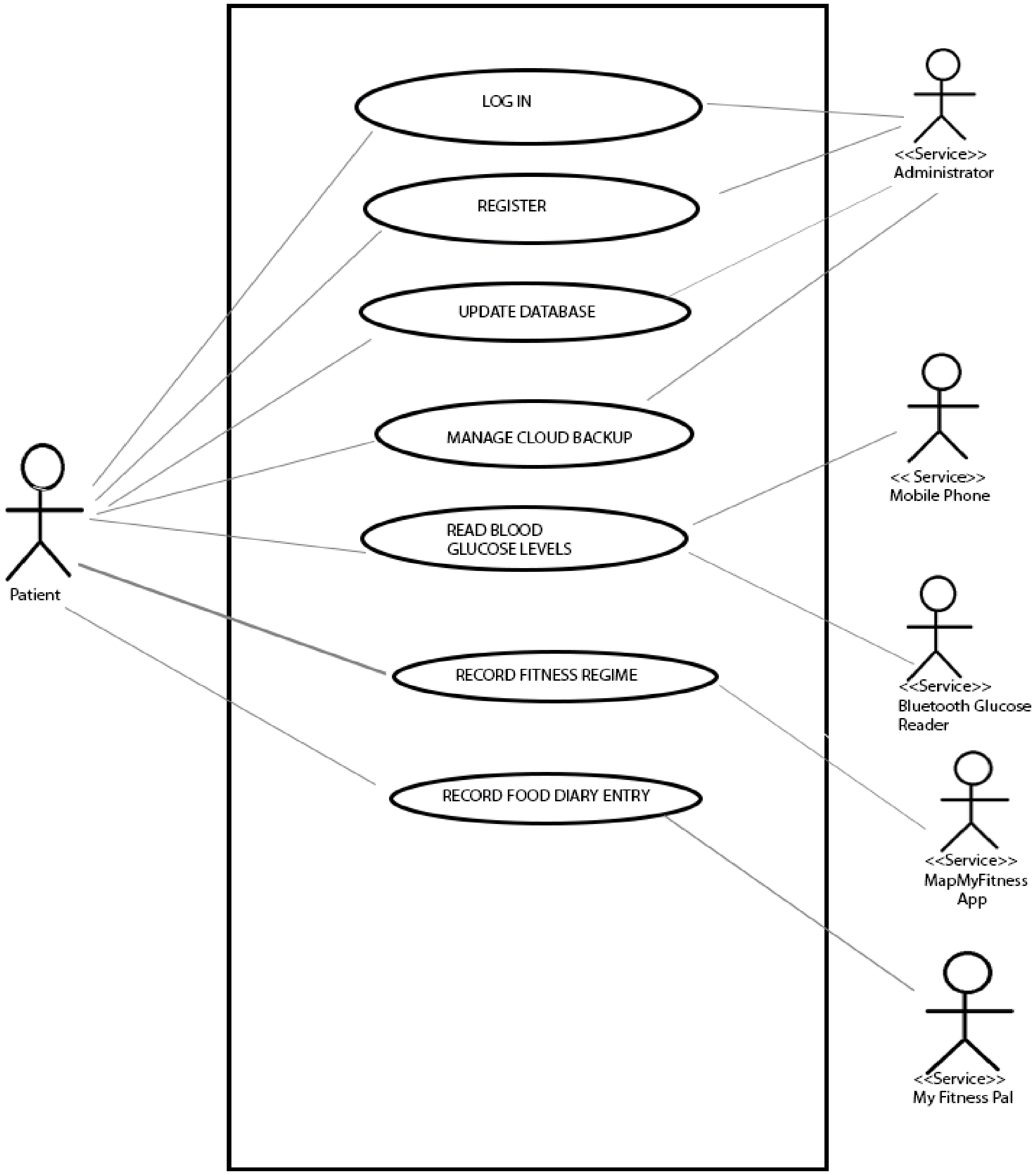
All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

Functional requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe *how* the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

6.2.3.1 Use Case Diagram


LOG IN
REGISTER
UPDATE DATABASE
MANAGE CLOUD BACKUP
READ BLOOD GLUCOSE LEVELS
RECORD FITNESS REGIME
RECORD FOOD DIARY ENTRY
Patient
<<Service>> Administrator
<< Service>> Mobile Phone
<<Service>> Bluetooth Glucose Reader
<<Service>> MapMyFitness App
<<Service>> My Fitness Pal

6.2.3.2 Requirement 1 <Register/Sign Up >

6.2.3.2.1 Description & Priority

This requirement allows the user to sign up to the system and have a record on the database which will allow him/her to use that account to log a blood glucose entry, fitness exercise regime and food diary entry. They will also be able to manage online backups to the cloud. So an account is important for activation

6.2.3.2.2 Use Case

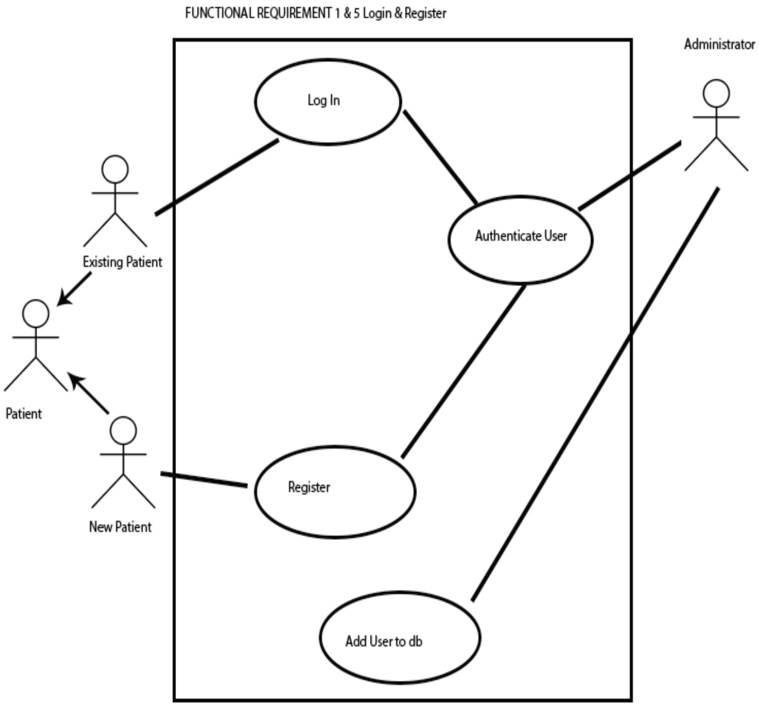
Scope

The scope of this use case is to set up a new user, make sure that they are not already on the system and create a relevant records for app usability

Description

This use case describes the approach the new user would use to register for the application

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the front/holding page to log on or register

Activation

This use case starts when a Patient clicks on the register link on the front page.

Main flow

1. The system identifies clicking of the Register user link
2. The Patient is identified as New Patient in this instance
3. The New Patient enters the registration details
4. The system notifies any registration errors E1
5. The Administrator authenticates the user
6. The Administrator adds a new record to the database

Exceptional flow

E1: Registration errors

1. The system highlights registration errors
2. The New Patient returns with correct data
3. The use case continues at position 5 of the main flow

Termination

The system presents the main menu for the user

Post condition

The system goes into a wait state

6.2.3.3 Requirement 2 <Read your blood glucose levels via Bluetooth>

6.2.3.3.1 Description & Priority

The system should be able to get the user to retrieve the glucose level via Bluetooth. This is one of the key elements of the application for ease of use.

6.2.3.3.2 Use Case

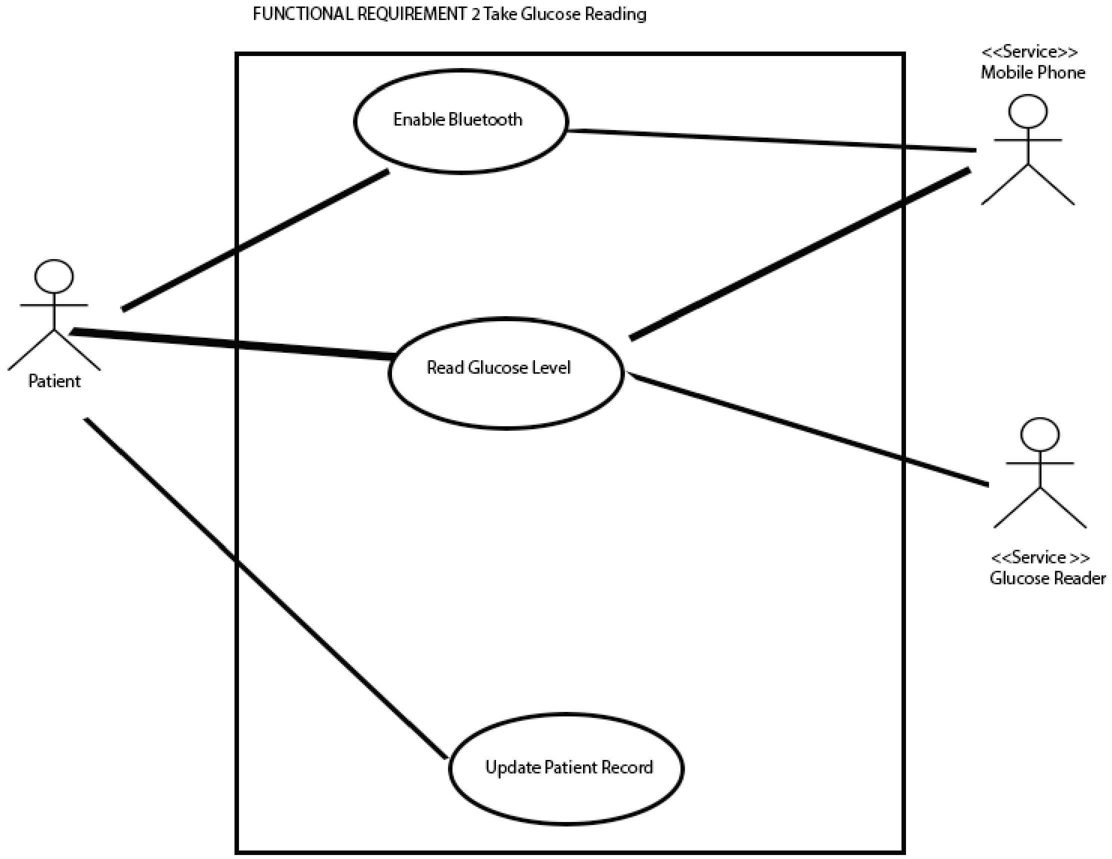
Scope

The scope of this use case is to ensure that the Bluetooth is enabled on the phone and the user is able to retrieve the data from the reader and the system will update the record

Description

This use case describes the process from enabling Bluetooth to updating the user record.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> clicks on “Take Glucose Reading” on the main menu

Main flow

1. The system identifies the action taken and prompts the user to enable the Bluetooth on the phone.
2. The system identifies the action taken and prompts the user to enable the Bluetooth on the phone.
3. The Bluetooth is already enabled (See A1)
4. The system searches for the Bluetooth device(See E1)
5. The system reads the latest reading from the Glucose reader
6. The <Patient> is prompted to update the record

Alternate flow

A1 : <Bluetooth Already Enabled>

1. The system notices that the Bluetooth is already enabled
2. The use case continues at position 3 of the main flow

Exceptional flow

E1: <Unable to find Glucose Reader>

1. The system prompts the user that the device cannot be found
2. The <Patient> enables Bluetooth on the Glucose Reader
3. The use case continues at position 4 of the main flow

Termination

The system updates the user record with the latest reading

Post condition

The system returns to the main menu

6.2.3.4 Requirement 3 <Integrate account with MapMyFitness>

6.2.3.4.1 Description & Priority

One of the main functions of the app is to integrate with 3rd party applications for users to log their exercises. This is important because the users can track glucose and exercises and see how well they are doing.

6.2.3.4.2 Use Case

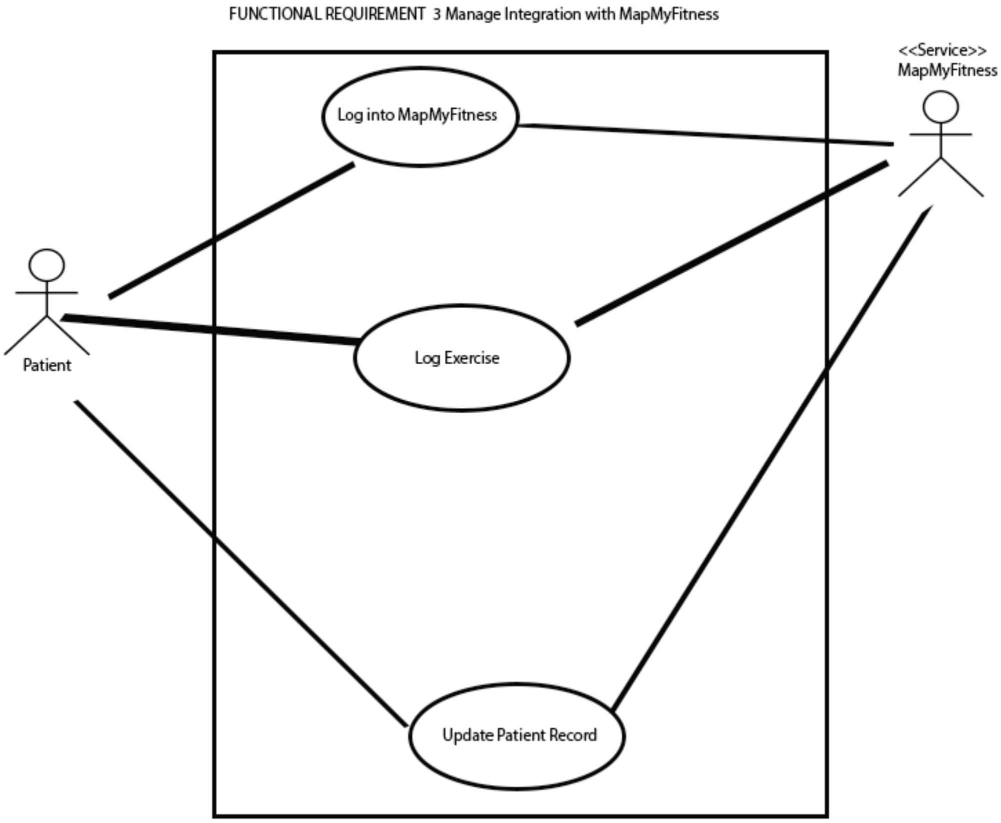
Scope

The scope of this use case is to allow Patients to use their YourDiabetes account for use with MapMyFitness.

Description

This use case describes the procedure to log an exercise using the MapMyFitness App

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> selects Log exercise

Main flow

1 The system identifies the user account credentials and attempts to log in to MapMyFitness.
2. The <Patient> may not have an account (See A1)
3. The <Patient> is now able to log an exercise
4 The system presents a <MapMyFitness> UI so the <Patient can log data.
5 The <Patient> cancels from procedure (See E1)
6. The system prompts user to update account with data from MMF

Alternate flow

A1 : <The Patient May not have an account using YourDiabetes>

1. The system uses the user credentials to attempt to create an account and informs that they may receive an email regarding such
2. The <MapMyFitness app> confirms with the system it has created an account.
3. The use case continues at position 3 of the main flow

Exceptional flow

E1 : <User cancels logging exercise>

1. The system returns to situation where they are prompted to log in to MapMyFitness

2. The use case continues at post condition

Termination

The system updates the user record with the latest reading

Post condition

The system returns to the main menu

6.2.3.5 Requirement 4 <Integrate with MyFitnessPal>

6.2.3.5.1 Description & Priority

One of the main functions of the app is to integrate with 3rd party applications for users to log their food diary. This is important because the users can track glucose and carbs/sugars and see how well they are doing

6.2.3.5.2 Use Case

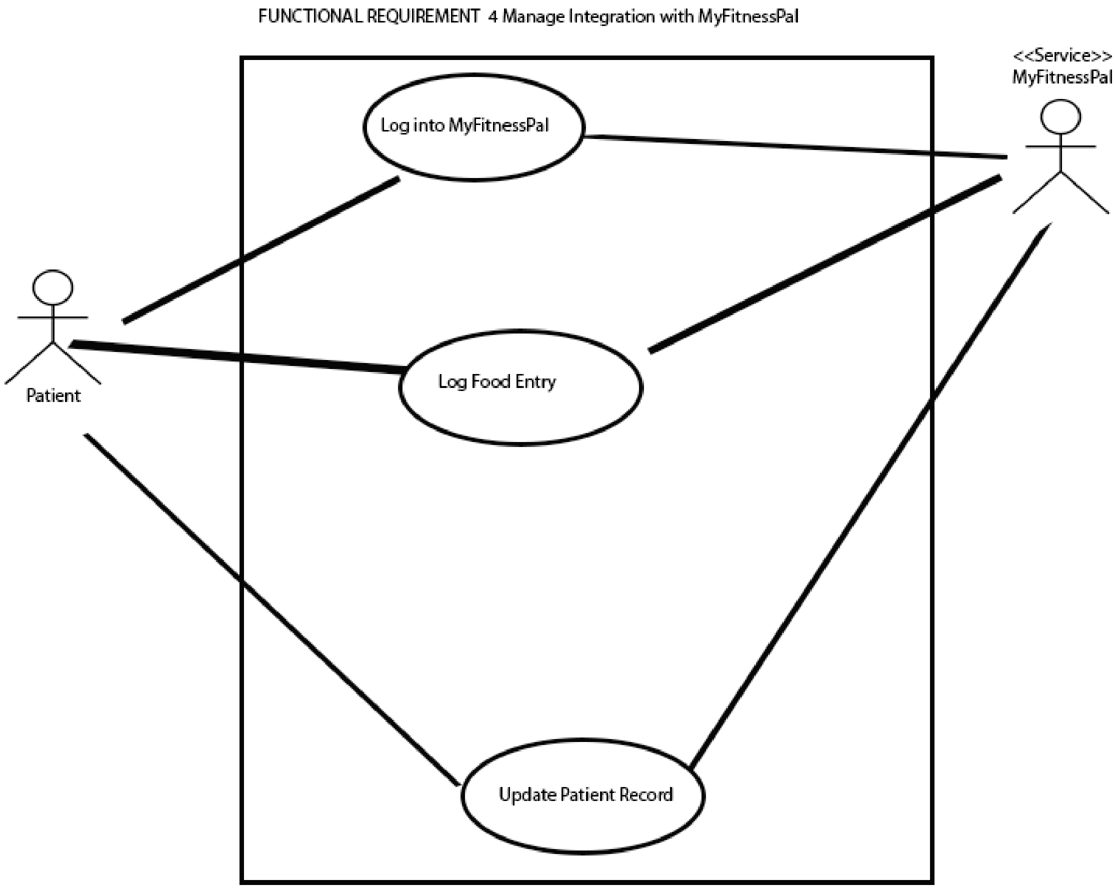
Scope

The scope of this use case is to allow Patients to use their YourDiabetes account for use with MyFitnessPal.

Description

This use case describes the procedure to log an exercise using the MyFitnessPal App.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> selects Log exercise

Main flow

1 The system identifies the user account credentials and attempts to log in to MyFitnessPal.

2. The <Patient> may not have an account (See A1)

3. The <Patient> is now able to log an exercise

4 The system presents a <MyFitnessPal> UI so the <Patient can log data

5 The <Patient> cancels from procedure (See E1)

6. The system prompts user to update account with data from MFP

Alternate flow

A1 : <The Patient May not have an account using YourDiabetes>

1. The system uses the user credentials to attempt to create an account and informs that they may receive an email regarding such
2. The <MyFitness Pal app> confirms with the system it has created an account.
3. The use case continues at position 3 of the main flow

Exceptional flow

E1 : <User cancels logging exercise>

1. The system returns to situation where they are prompted to log in to MyFitnessPal

2. The use case continues at post condition

Termination

The system updates the user record with the latest reading

Post condition

The system returns to the main menu

6.2.3.6 Requirement 5 <Reporting>

6.2.3.6.1 Description & Priority

6.1.1.6.2 Use Case

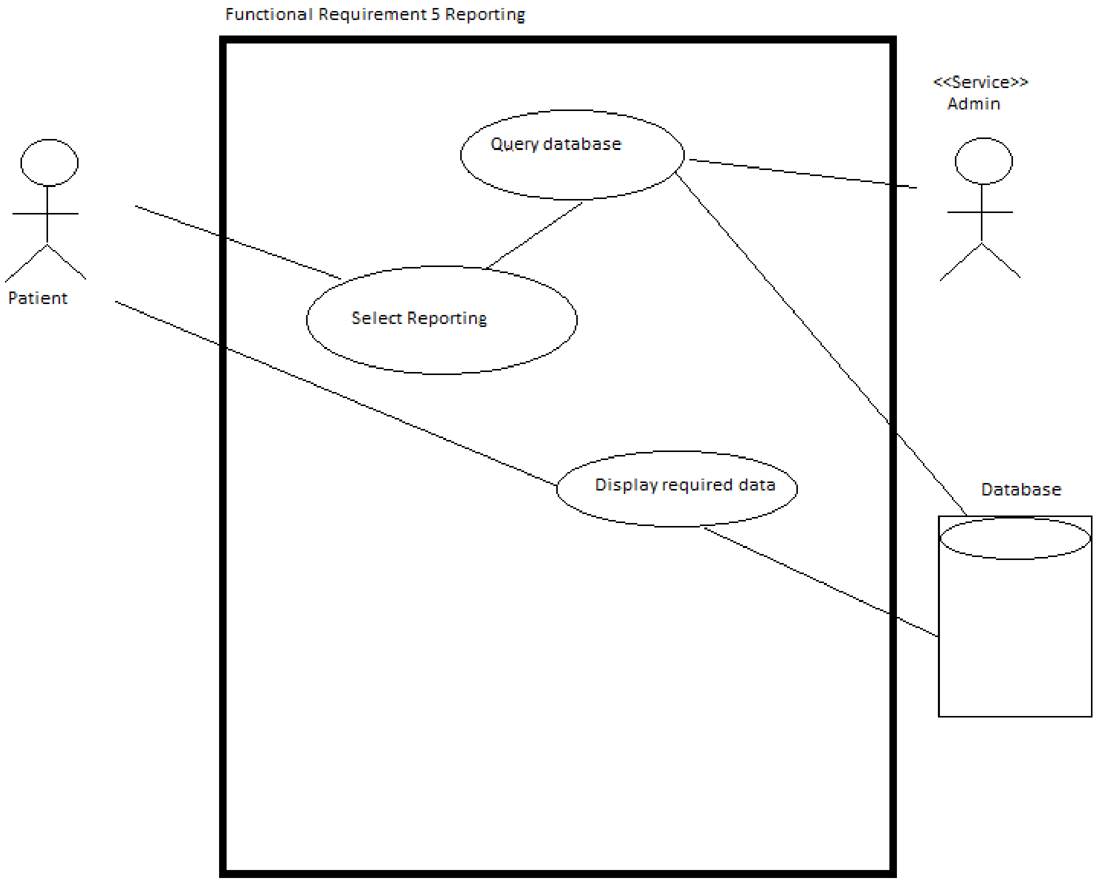
Scope

The scope of this use case is to allow Patients to view reports based on the data collected

Description

This use case describes the procedure to view a report on the application

Use Case diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> selects Report

Main flow

1 The system identifies the user query from the menu

2. The <Patient> may not know how to report in the new app (See A1)

3. The <Patient> is now able to query the database

4 The system provides options for the period they want to query

5 The <Patient> cancels from procedure (See E1)

6. The system provides the user with required report

Alternate flow

A1 : < The patient may not know how to report in the new app >

1. The system provides demos as to what can be looked up for in the new app

2. The use case continues at position 3 of the main flow

Exceptional flow

E1 : <User cancels reporting query>

1. The user is asked if they want to return to the main menu

2. The use case continues at post condition

Termination

The system closes the query from the database and all links to it.

Post condition

The system returns to the main menu

6.2.3.7 Requirement 5 <Log in to the system>

6.2.3.7.1 Description & Priority

User must be able to log in with their credentials. The idea will be that it will be difficult to retrieve data because of the sensitivity.

6.2.3.7.1 Use Case

Scope

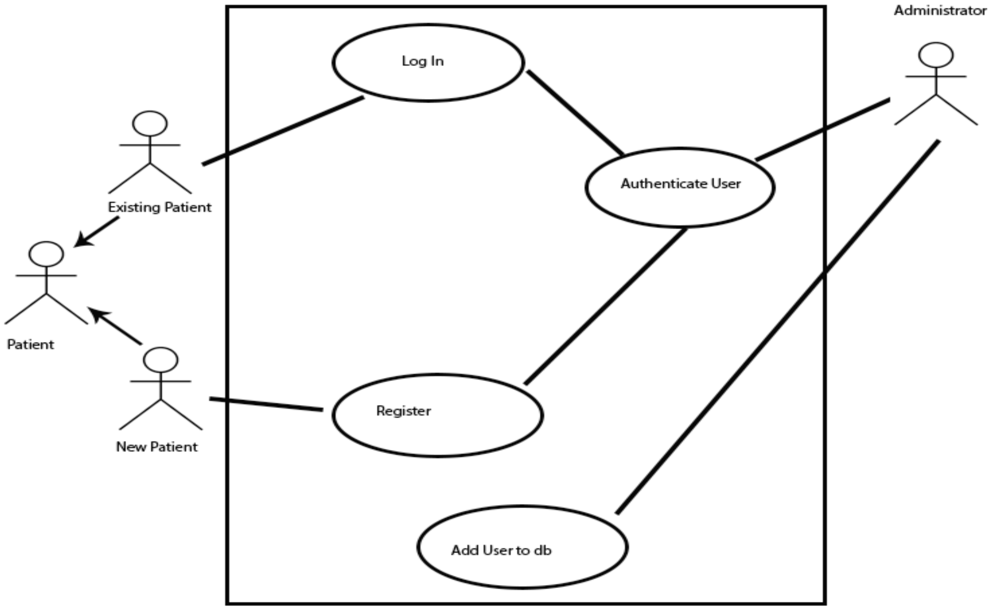
The scope of this use case is to get users to log in and be authenticated

Description

This use case describes the process in which users log in to the system for use of the application

Use Case Diagram

FUNCTIONAL REQUIREMENT 1 & 5 Login & Register



Flow Description

Precondition

The system is in initialisation mode with the <Patient> is at the Front Page of the UI. The prompts here are Log In and Register

Activation

This use case starts when a <Patient> enters their user name and password, and then clicks on Log-in

Main flow

1. The system identifies the log in button pressed
2. The system sends the user a verification code via text
3. The <Patient> enters the verification code and accesses the main menu
4. The system is unable to verify the <Patient> credentials (see A1)
5. The system is unable to find the <Patient> on the system

Alternate flow

A1: <unable to verify user credentials>

1. The system notifies the user that the credentials are incorrect
2. The <Patient> provides an email address to change the password
3. The use case continues at position 1 of the main flow

A2 : <no account has been made for this user>

1 The system finds out the user does not exist in the database

2. The user is prompted that this is the case

3. <Patient> prompted to register (see requirement 1)

Termination

The system returns to the login screen

Post condition

The system goes to the main menu for the application

6.2.3.8 Requirement 6 <Backup to a cloud>

6.2.3.8.1 Description & Priority

This requirement allows the user to maintain their data in a cloud hosting service (yet to be determined).

6.2.3.8.2 Use Case

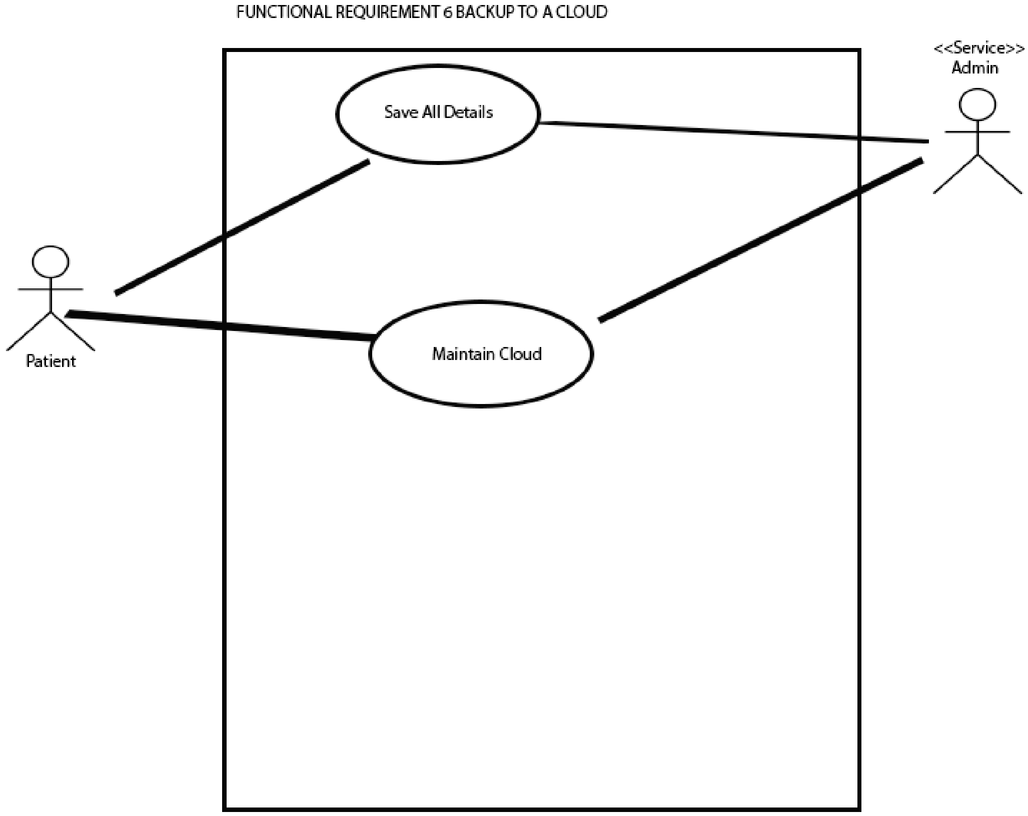
Scope

The scope of this use case is to allow the user to save all current details or maintain how the data is automatically backed up

Description

This use case describes the interaction between services provided and user and admin.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the user is at the main menu state

Activation

This use case starts when a <Patient> selects Manage Cloud Backup

Main flow

1 .The system identifies the selection made by the <Patient> and provides them with 2 options initially. Automatically schedule backup and "Backup latest Local Data"

2. The <Patient> selects Automatically Schedule backup and modifies selection for cloud backup

3. The <Patient> selects Backup Latest Local Data (See A1)

4. The <Patient> clicks return to main menu

Alternate flow

A1: <Patient selects Backup Latest Local Data>

5. The system checks the local data saved on the device and compares it with the cloud data
6. If all data is same the <Patient> is prompted such.
7. Some data is different (see E1)
8 .The use case continues at step 4

Exceptional flow

E1 : <Data is different to that of cloud>

9. The system notifies patient that data is different and prompts for backup data
10. The <Patient> selects backup data
11. The system backs up the data.
12. The use case continues at position 4 of the main flow

Termination

The system goes to the main menu for the application.

Post condition

The system goes to the main menu for the application

Non-Functional Requirements

6.2.3.9 Performance/Response time requirement

6.2.3.10 Availability requirement

Application must be available online and offline, hence the requirement for cloud when mobile is connected.

6.2.3.11 Recover requirement

Data must be able to be recovered at all times hence the use of the cloud as a third party.

6.2.3.10 Security requirement

The user details must be made secure as they are patient details

6.2.3.11 Reliability requirement

The details must be correct so that the customer can rely on them to get accurate information about their health

6.2.3.12 Maintainability Requirement

Any changes to the system must not cause problems to the user

6.2.3.13 Portability Requirement

This is important as the primary use of the application is via mobile devices (android)

6.2.3.14 Extendibility Requirement

The database must be able to cope with many users

6.2.3.15 Reusability Requirement

The scope of the project is to use applications already available so that people can maintain their health in one area

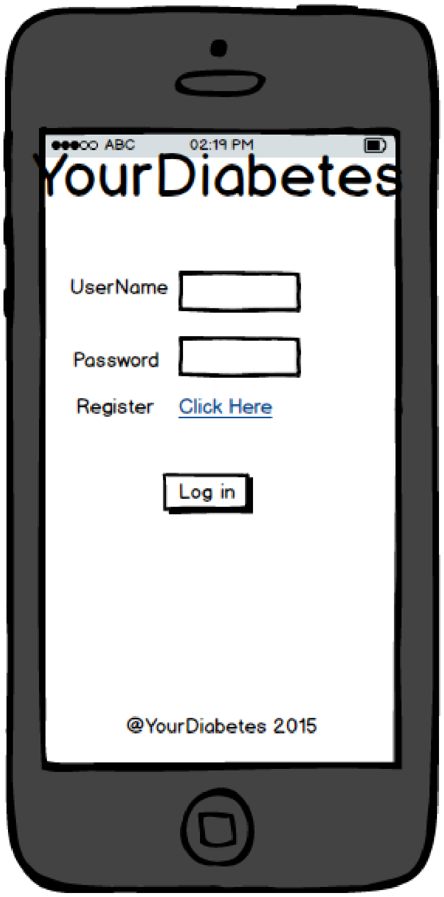
6.2.4 Interface requirements

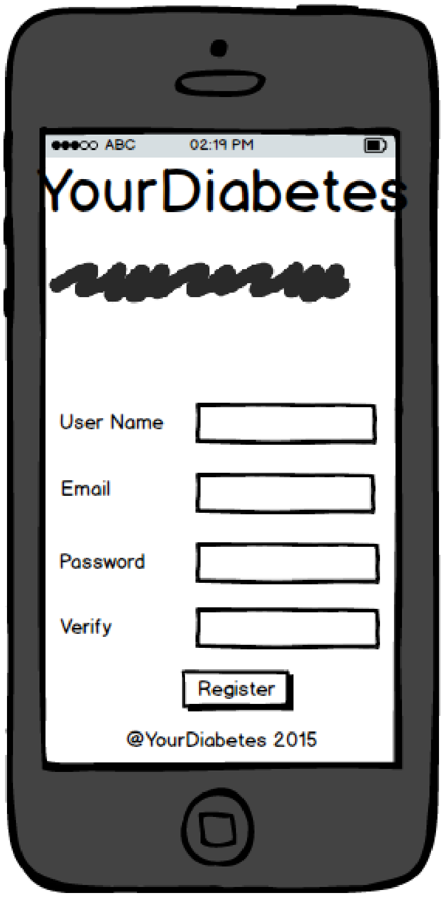
When users are registering they will enter their details via keypad on the mobile. Alternatively they will sign up using their existing Facebook or twitter account.

When adding a reading the Bluetooth device will have to be enabled on the smartphone.

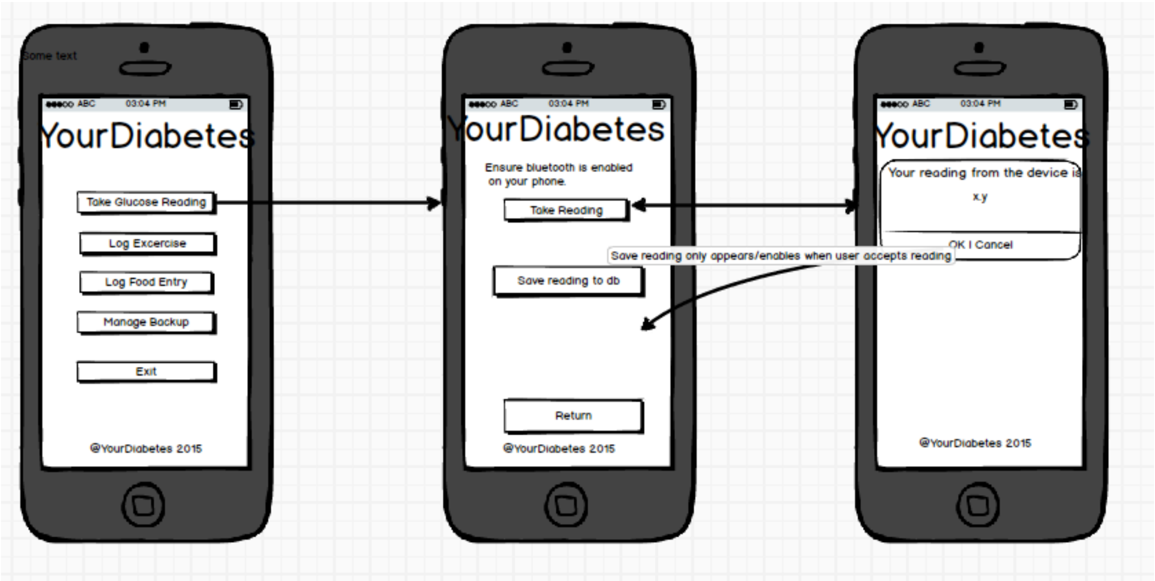
6.2.5 GUI

6.2.5.1 Mock-up GUI for login and Registration

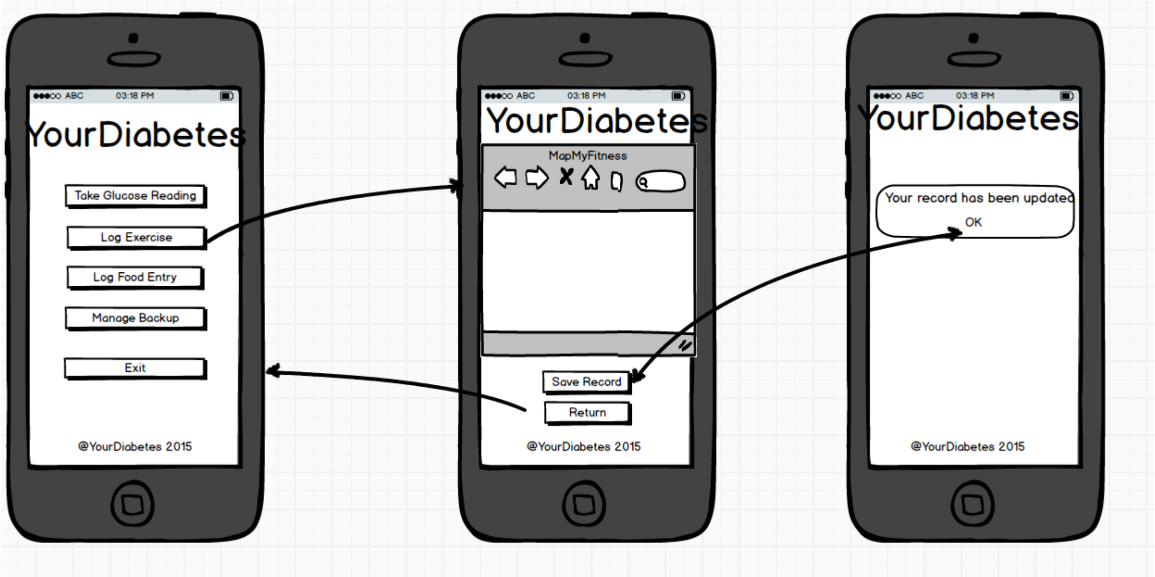




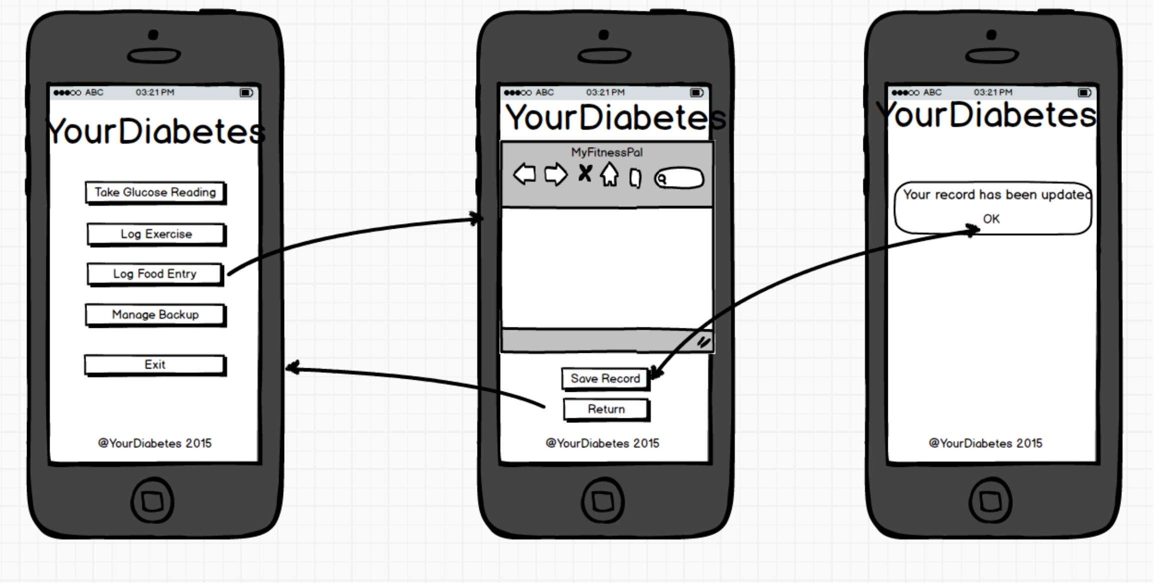
6.2.5.2 Mock-up for Take reading from device



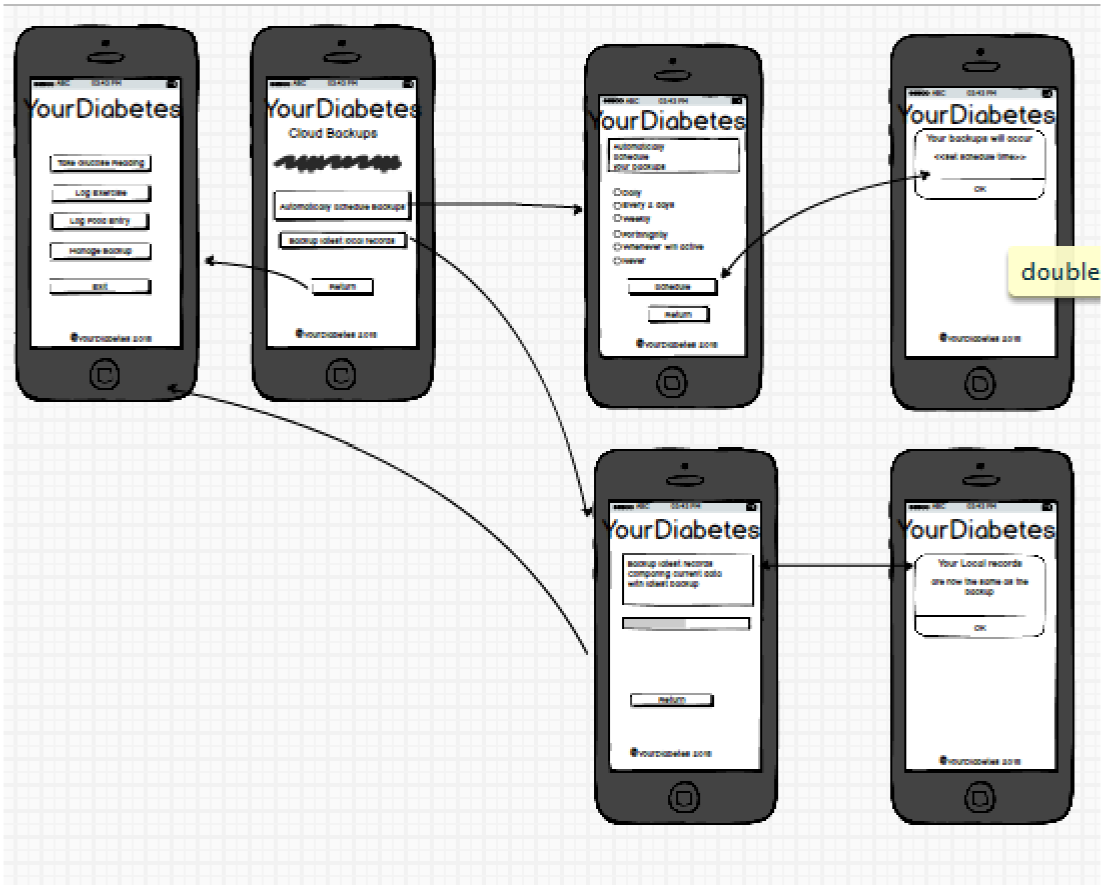
6.2.5.3 Mock-up for Integration with map my fitness



6.2.5.4 Mock-up for integration with myFitnessPal



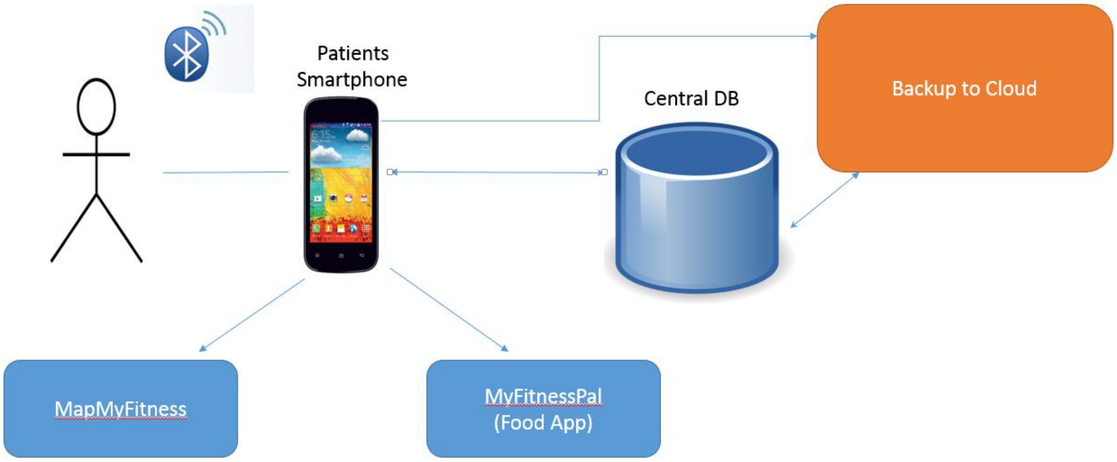
6.2.5.5 Mock-up for backup to cloud



6.2.6 Application Programming Interfaces (API)

The system will use MapMyFitness for exercise logging and MyFitnessPal for food diary

6.2.7 System Architecture



6.2.8 System Evolution

Over time the system will be able to accept different types of Bluetooth devices and have an iOS version. The interfaces will be continually checked for user acceptance. Reporting will be available for doctors and consultants to view the progress of the patient

6.3 Monthly Journals

6.3.1 October 2015

6.3.1.1 My Achievements

This month, I was in contact with Diabetes Ireland, who provided me with a Bluetooth tester.

I was able to complete the requirements specification and identify the major functions of the project. At the time of writing, I put down everything that I thought could be completed along with those that cannot. Some requirements were mentioned to me by other suffers of Diabetes like backing up the data to a cloud.

I have also retrieved contacts for people to draw up the UI for the project. The purpose of this is to allow the end user to be involved in the interface. Experience from other applications can be more beneficial in drawing up the right experience for the customer

6.3.1.1.1 My Reflection

By keeping the project fresh in my mind, I am able to continuously think of new Ideas

I could begin on plotting down time slots for future developments

6.3.1.1.2 Intended Changes

I need to keep the project simple and focus on the needs and not the wants

6.3.1.1.3 Supervisor Meetings

Date of Meeting: 20th October

Items discussed: Complexity of project. Achieving deadlines

Action Items:

Requirements spec

6.3.2 November 2015

6.3.2.1.1 My Achievements

This month, I produced the requirements specification and spoke with Paul Hayes regarding same. See meeting for discussion

6.3.2.1.2 My Reflection

I could have started work on the development of the application and using Microsoft Visual Studio to refresh my work with C#.

6.3.2.1.3 Intended Changes

N/A

6.3.2.1.4 Supervisor Meetings

Date of Meeting: 26th November

Items discussed: Requirements specification and programming language used to develop application

Action Items: Paul suggested speaking with Michael Bradford regarding programming

6.3.3 December 2015

6.3.3.1.1 My Achievements

Very little was done in relation to the project. I spoke with Ron Elliot regarding the project and he seemed quite interested and told me about a project he was involved in. I sent him on the proposal and requirements spec. As per previous supervisor meeting I sent a mail to Michael Bradford regarding C# assistance.

6.3.3.1.2 My Reflection

Again, I could have begun development of the application and using Microsoft visual Studio to refresh my work with c#. However work and personal life took up most of my time.

6.3.3.1.3 Intended Changes

N/A

6.3.3.1.4 Supervisor Meetings

No meetings in December

6.3.4 January 2016

6.3.4.1.1 My Achievements

After studying for and completing the exams for Semester 1, I began work on a prototype for the mid-term presentation. This included development and structuring of the core functionalities of the project to at least get going. I started using Pluralsight videos for education on how to do this. However work and time constraints stopped me from getting a prototype up and running for the time being. I have been trying to get Xamarin Student licence for a while and eventually got it activated

6.3.4.1.2 My Reflection

I could have used the time over Christmas effectively to develop a plan for learning the essentials for the project and got Visual Studio and Xamarin development applications installed in the relevant locations for use.

6.3.4.1.3 Intended Changes

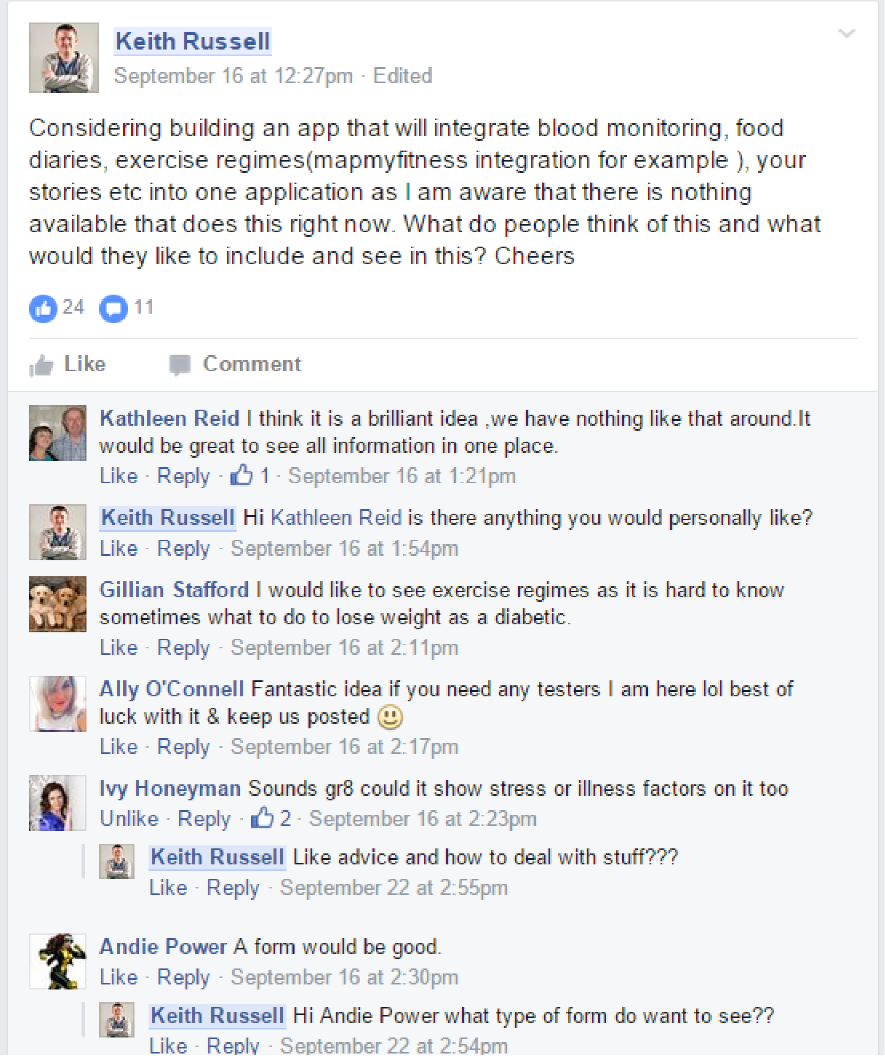
Continue on with the development of the project as the majority of the core idea is there now and press on.

6.3.4.1.4 Supervisor Meetings

No meetings in January because of Exams and return to class on the 26th January. Due to have one on 4th Feb 2016

6.4 Other Material Used

Feedback from people on diabetes Ireland website



Keith Russell
September 16 at 12:27pm · Edited

Considering building an app that will integrate blood monitoring, food diaries, exercise regimes(mapmyfitness integration for example), your stories etc into one application as I am aware that there is nothing available that does this right now. What do people think of this and what would they like to include and see in this? Cheers

👍 24 💬 11

Kathleen Reid I think it is a brilliant idea ,we have nothing like that around.It would be great to see all information in one place.
Like · Reply · 👍 1 · September 16 at 1:21pm

Keith Russell Hi Kathleen Reid is there anything you would personally like?
Like · Reply · September 16 at 1:54pm

Gillian Stafford I would like to see exercise regimes as it is hard to know sometimes what to do to lose weight as a diabetic.
Like · Reply · September 16 at 2:11pm

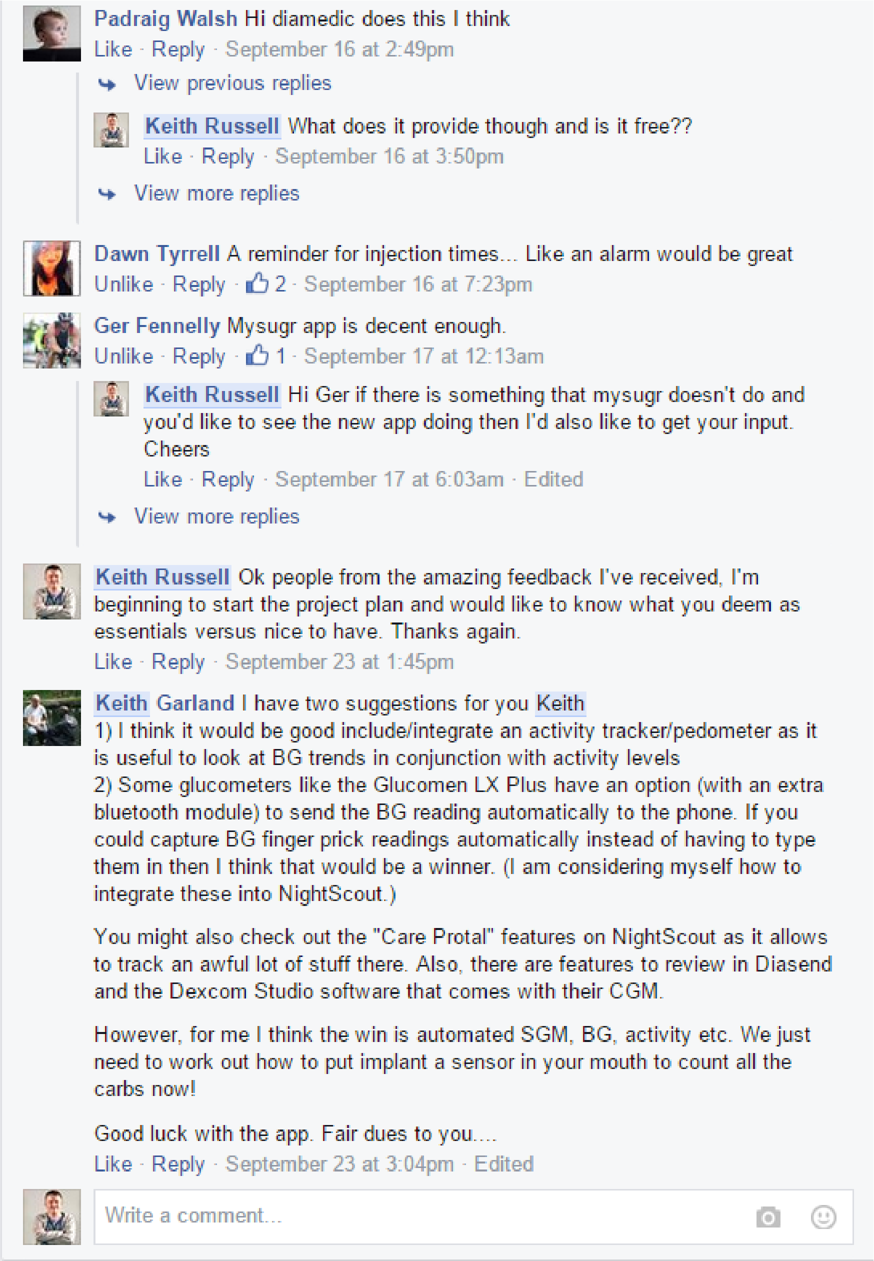
Ally O'Connell Fantastic idea if you need any testers I am here lol best of luck with it & keep us posted 🙂
Like · Reply · September 16 at 2:17pm

Ivy Honeyman Sounds gr8 could it show stress or illness factors on it too
Unlike · Reply · 👍 2 · September 16 at 2:23pm

> **Keith Russell** Like advice and how to deal with stuff???
> Like · Reply · September 22 at 2:55pm

Andie Power A form would be good.
Like · Reply · September 16 at 2:30pm

> **Keith Russell** Hi Andie Power what type of form do want to see??
> Like · Reply · September 22 at 2:54pm



Padraig Walsh Hi diamedic does this I think
Like · Reply · September 16 at 2:49pm

↳ View previous replies

Keith Russell What does it provide though and is it free??
Like · Reply · September 16 at 3:50pm

↳ View more replies

Dawn Tyrrell A reminder for injection times... Like an alarm would be great
Unlike · Reply · 👍 2 · September 16 at 7:23pm

Ger Fennelly Mysugr app is decent enough.
Unlike · Reply · 👍 1 · September 17 at 12:13am

Keith Russell Hi Ger if there is something that mysugr doesn't do and you'd like to see the new app doing then I'd also like to get your input. Cheers
Like · Reply · September 17 at 6:03am · Edited

↳ View more replies

Keith Russell Ok people from the amazing feedback I've received, I'm beginning to start the project plan and would like to know what you deem as essentials versus nice to have. Thanks again.
Like · Reply · September 23 at 1:45pm

Keith Garland I have two suggestions for you Keith
1) I think it would be good include/integrate an activity tracker/pedometer as it is useful to look at BG trends in conjunction with activity levels
2) Some glucometers like the Glucomen LX Plus have an option (with an extra bluetooth module) to send the BG reading automatically to the phone. If you could capture BG finger prick readings automatically instead of having to type them in then I think that would be a winner. (I am considering myself how to integrate these into NightScout.)

You might also check out the "Care Protal" features on NightScout as it allows to track an awful lot of stuff there. Also, there are features to review in Diasend and the Dexcom Studio software that comes with their CGM.

However, for me I think the win is automated SGM, BG, activity etc. We just need to work out how to put implant a sensor in your mouth to count all the carbs now!

Good luck with the app. Fair dues to you....
Like · Reply · September 23 at 3:04pm · Edited