

**DIABETES MANAGEMENT SYSTEM**

INTR 8015 PROJECT IMPLEMENTATION PHASE



**Student ID:** R00076703

**Supervisor:** Karl Grabe

**Second Reader:** Noreen Gubbins

**Date:** 04 January 2015

**Version: 1.1**

**BSc (Hons) in IT Management**

**Department of Computing**

**Declaration**

I hereby certify that this material which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent, that such work has been cited and acknowledged within the text of my work.

I understand that my project documentation may be stored in the library at CIT, and may be referenced by others in the future.

**Signed: ­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Date:** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Table of Contents**

**PAGE**

**Executive Summary……………………………………… 4 – 4**

1. **Introduction…………………………………………………5 – 6** 
   1. Project Background**………………………………………**5 – 5
   2. Goals of the System**………………………………….**6 – 6
   3. Objective of the System**……………………………...**6 – 6
2. **Methodology………………………………………....86 – 103** 
   1. Revision Control Software**……………………………**
   2. Eclipse PHP Development Tools**….………………….**
   3. Lifecycle**………………………………………….......**
3. **Analysis and Design…………………………..……...42 – 85** 
   1. Problem Definition**………………………………….**
   2. The Stakeholders**……………………………………**
   3. Project Constraints**………………………………….**
   4. Relevant Facts and Assumptions**……………….......**
   5. Product Boundary Use Case Diagram**……………...**
   6. Project Use Case List**……………………………….**
   7. Project Use Case Narrative**…………………………**
   8. Summary of Prototyping**…………………………....**
      1. FunctionalRequirements*………….81 – 81*
      2. Non-Functional Requirements*……82 - 82*
   9. Data Dictionary**…………………………………….**
   10. Collaboration Diagrams**……………………………**
   11. Sequence Diagrams**………………………………..**
   12. Business Data Model / Class Diagrams**……………**
   13. Activity Diagram**…………………………………..**
   14. Database Model**……………………………………**
4. **Implementation…………………………….……..106 – 107** 
   1. User Manual**…………………………………**
   2. Validation Testing**…………………..……….**
5. **Evaluation…………………………………….……..106 – 107** 
   1. Revised System Objectives**……………………**
   2. Problems Encountered**…………………………**
   3. OutLook**……………………………………….**
6. **Conclusions…………………………………….……..106 – 107**
7. **Bibliography………………………………………....108 – 108**

**EXECUTIVE SUMMARY**

Over the past 50 years, there has been a dramatic increase in the prevalence of interrelated metabolic disease states, including obesity, insulin resistance, and type 2 diabetes. Diabetes Mellitus is a common condition that affects millions of people worldwide. For instance, if diabetes is not treated, it can lead to serious complications such as blindness, amputations, and even death. However, before people develop diabetes, they always develop pre-diabetes. Healthy people and people with pre-diabetes may be able to prevent or delay type 2 diabetes from developing.

In order to prevent the risks of serious complications that can be caused by diabetes, the blood glucose level must be monitored. The blood glucose level or blood sugar concentration is the amount of glucose (sugar) present in the blood of a human. Therefore, to monitor the blood glucose level in the blood an online application known as the **Diabetes Management System** (DMS) is developed.

Diabetes Management System is a powerful and efficient way to track and monitor the management of diabetes over time. The system will be developed as an online application, where patients and doctors can track, monitor, and keep an accurate reading of blood glucose level tests.

The Diabetes Management System (DMS) will assist in the effective management of diabetes. It will provide a means of tracking, organising, and managing all of the activities involved in monitoring diabetes. DMS will work in parallel with the existing paper-based system i.e. the Diabetes Monitoring Diary. The DMS will make it possible to track patient’s data stored in the paper-based system. Moreover, the new system (DMS) can be used to replace the existing system (DMD), in order to make monitoring of diabetes easy for people with diabetes. See below.

**EXISTING** **NEW**

PAPER-BASED WEB-BASED

Mary Kate Mary Kate

**Existing System** – Diabetes Monitoring Dairy (DMD)

**New System** – Diabetes Management System (DMS)

1. **INTRODUCTION**
   1. PROJECT BACKGROUND

As a final year student in Information Technology Management (ITM), I undertake the effort to develop an interactive web-based Diabetes Management System. The application provides an easily-accessible means of managing an Endocrinology/Diabetes and its various Patients, Doctors and Nurses. This system is accessible by all stakeholders and mediates access via a security system. Diabetes Management System will transform the paper-based filling system i.e. Diabetes Monitoring Diary to a web-application system and enable simultaneous multi-user access to a central database.

My inspiration for the Diabetes Management System came from my Mum. She was recently diagnosed with TYPE 1 Diabetes. As a result of this, she needs to take insulin by injection, measure her daily blood glucose level and record the result of her blood glucose levels in her Diabetes Monitoring Diary. The diary enables her to keep a record of her daily blood glucose level and any appointments with her diabetes nurse or doctor. Moreover, the information in the diary helps her doctor/nurse to make the right adjustments to her insulin doses.

Therefore, in order to curb society’s risk for many serious health problems caused by Diabetes, I have decided to develop a **Diabetes Management System** (also known as the Diabetes Monitoring Application) for South Terrace Cork Hospital. The Diabetes Management System will be used for monitoring daily blood glucose level, making the right adjustments to insulin doses and keeping a record of any appointments with the diabetes nurse or doctor.

South Terrace Cork Hospital is the name of the business chosen to help gather more information on the requirement specifications needed for the Diabetes Management System. It is one of the leading diabetes clinics in Ireland, with the main clinic located in Northwood Business Campus Santry, Dublin 9 and seven other clinics located throughout Ireland. With the newly built System, the business will be able to provide an effective management of co-morbidities and diabetes for individuals and all groups at risk of developing health issues.

In addition, the system will be designed for hospitals and healthcare providers with the intention of preventing or delaying the short and long term complications caused by Diabetes. For instance, when a doctor defines a treatment plan for a diabetic based on his/her diabetes type, sex, age, etc. factors. The Diabetes Management System (DMS) can be used to send a reminder to patients to ensure better compliance with the treatment plan. And, once the patients, undergo tests, the data such as blood glucose level and insulin injections, is captured in the system. Then, doctors can view the progress of individual patient and take necessary interventions if needed.

* 1. GOALS OF THE SYSTEM
     1. The system will enable patients to enter their personal details, which will be saved to the Diabetes Management System database.
     2. The system will enable patients to keep track of their blood glucose level remotely. This process will reduce the time spent in filling out this information that is driven by the paper-based system i.e. the diabetes monitoring dairy.

With the Diabetes Management System, patients will be able to keep record of all their blood glucose level tests at any time.

* + 1. The system will be able to keep track of the patients’ insulin doses and any adjustments made by a doctor or nurse.
    2. The system will enable patients to update and view their personal information in real-time.
    3. The system will enable patients to leave comments and personal notes.
    4. The system will enable healthcare professionals to make appropriate insulin adjustments for their patients.
    5. The system will enable the secretary to register patients into the diabetes management system.
  1. OBJECTIVE OF THE SYSTEM
     1. The plan for the new system is to provide the capability for patients to update data in real-time. This process will reduce, if not eliminate some of the current problems that diabetic face – including blood glucose level test results that are out-of-date and incorrect information on the patients insulin injection dose.

In return, patients’ blood glucose level test results will be recorded in the system database, which will be up-to-date.

* + 1. To provide world ready diabetes management software that can be adapted to various web-browsers. The software will make diabetes management easier and more efficient than ever before.

* + 1. The aim is to design an online application that will be integrated into a mobile application. This application will help people with Type 1 diabetes and Type 2 diabetes manage the data associated with:

1. Records of actual insulin delivery
2. Statistical Evaluation
3. Coaching of dose corrections
4. Blood test results from a glucose meter.
5. Manual log entries for exercise and other factors.
6. **METHODOLOGY**

In order to produce a world ready application, a combination of iterative and prototype-based development **methodologies** is used during the development of Diabetes Management System.

The **tools** that enabled the process for the completion of the project are as follows:

**Revision Control Software**

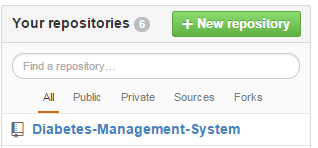
The revision control software also known as the source control software is a management tool used to help build better software and also provide a great opportunity to work efficiently. I decided to use ***Git***, a distributed source control system, and its companion website ***GitHub***. GitHub is an instance of a hosting service that uses the Git revision control system. They are powerful mechanism tools that provided numerous features, which made code management effortless. Moreover, it is a web-based hosting service for software projects.

The basic unit of work in GitHub is the **commit**. A commit is a code change with an accompanying comment by its author. Whenever a change takes place, the details are immediately published on the GitHub website: who made the changes, when it was made, what lines were changed, and many more. And a comprehensive history maintained with all the details of the changes made in an easily accessible way.

An example of this is shown below:

*Commits published on GitHub*

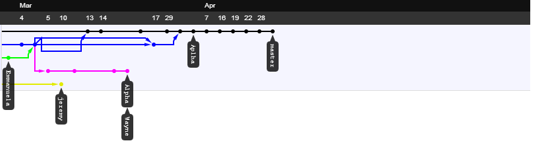
A collection of source code and its associated history is called a **repository** also known as **‘repo’** in GitHub. In order words, a repository was created to give full access, making it possible for me to make necessary changes at my spare time by **pushing** commits to the Git and to receive updates with new changes by **fetching** from the GitHub. In addition, these are simple commands triggered by a button press in the development tools.



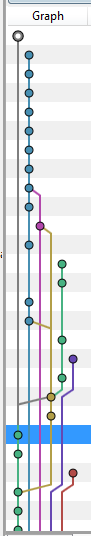
*Repository created on GitHub*

One of Git’s most important features is the ability to create ***branches***. Branches in GitHub are concurrent lines of development independent of the ‘**master**’ code, in which changes can be made without the fear of interference.

The most helpful feature of GitHub is ***merging***. When a feature is completed and tested, its branch can be merged back into the master branch to integrate the changes into it. The process of merging is mostly automatic, and conflicts are easily resolved manually. Because branches diverge from the master codebase, over time they grow out of sync as other work is completed and integrated. But merging the work in both ways (i.e. by merging the master branch into the feature branch), it can be brought up to the date with the latest changes, without any interference to the main codebase.



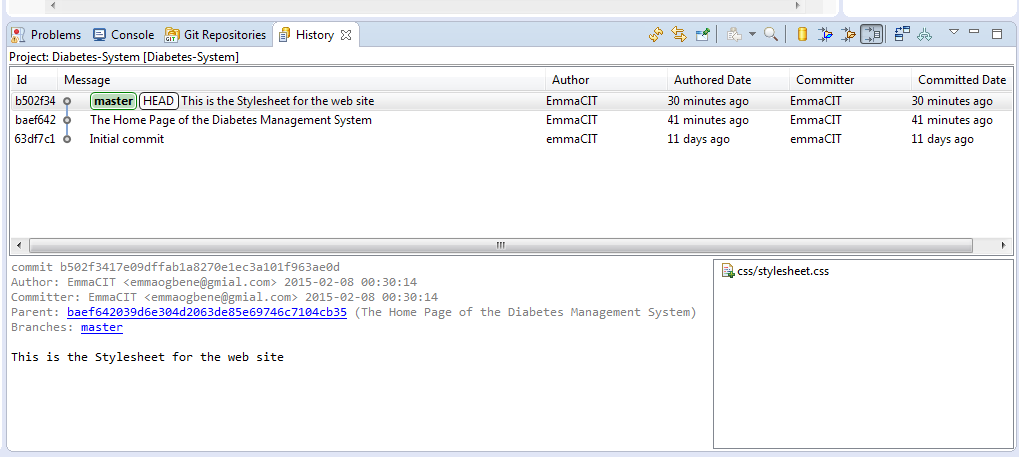
*An excerpt from repository history:* ***dots are commits, lines are branches, and arrows signify merges***



*An excerpt from source Tree history: dots are commits, lines are branches*

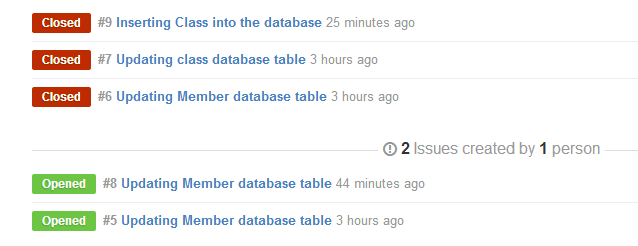
**Eclipse PHP Development Tool**

Git is completely integrated into Eclipse PHP, the development environment that is used for the project. It integrates all of Git’s functionality in a way that is quite simple to use.



*E.g. of Commits and branches in Eclipse’s history view*

In conclusion, Git solves almost every technical issue involved with team-based development. GitHub provides an issue-tracking system which is integrated with the commit log. This will be used to keep track of bugs founded and to specify the features needed. Whenever an issue is created, it is notified by email. The issue tracker is also integrated with the commit history – commits can be marked as related to a given issue and they show up on the issue page on the website. This will make it very easy to see what is done about each problem as they are encountered and fixed.



*E.g. of a GitHub’s issue tracker*

**Life cycle**

Firstly, the development of the application will undertake an iterative/prototype development based approach. One of the greatest aspects of using this model is that, anyone can see the development of the software on a step by step scale. The whole process is iterative and will make it easy to demonstrate things even when some the materials are temporary, it will be easy to change and upgrade.

Additionally, both incremental and iterative model are main procedures in the chosen methodology, which will be used in the development cycle of the Diabetes Management System. Underlying features required will be developed in separate branches, tested, and then merged when it is suitable to be integrated into the main branch.

To sum up what needs to be done, it closely relates to the iterative and incremental model.



Also, the model will allow for the addition of new features to be added quickly and easily.

For a specific instance of this process above, the primary GUI components will be considered: the tabs that will be used to manipulate information in the database.

In order to begin, the requirements laid out in the report will be used. It will provide designs for the appearance and general functionality of each tab. The functional requirements will determine what is required to be done at a finer level of detail.

Branches for each tab will be created and implemented. Multiple tabs will be developed simultaneously in different branches. When each tab is completed and tested in isolation, the related branch will be merged into the master branch and integration testing will be carried out. If by any chance, there are commonalities in the work, various classes will be created to reduce redundancy and simplify development on each of the iteration. As a result of iterative prototyping and incremental improvement, there will be practically no code redundancy in the program. The design is quite elegant, featuring many reusable components.

A complete history of all development activity will be visible on the Git repository on GitHub, line by line.

**Other Tools include:**

1. XAMPP

XAMPP is a free and open source cross-platform web server solution stack package. It consists of the following components such as **Apache HTTP Server, MySQL database, PHP Programming Language, Perl Programming Language, FTP Server** and **SMTP Server**.

For the development of the website, the latest version of XAMPP, will be downloaded and installed. By using XAMPP, it helps the installation process of the named components to be quick and easy. For instance, installing the components individually, factors such as PHP configuration and the setup of MySQL will not have to be considered.

1. PHP

PHP is a general-purpose scripting language that is useful for web-development. It is used to enhance webpages. And, it is a server-side scripting language whereby the PHP code isn’t executed on the user’s computer rather; it is executed on the computer in which the user requested the PHP pages from. As a result, the pages are retrieved and displayed on the user’s web-browser.

1. HTML5

HTML5 will be used in the development of the project. HTML5 is a core technology markup language of the Internet used for structuring and presenting content for the World Wide Web (www). It is the final and complete fifth version of the HTML standard of the World Wide Web Consortium (W3C).

1. CSS

CSS (Cascading Style Sheets) is a style sheet language, which will be used for describing the look and formatting of a document written in a markup language.

1. Visual Paradigm

Visual Paradigm will be used to model the use cases and domain model of the application. It is a software design tool tailored for agile software projects. Also, it supports BPMN, UML, DFD, ERD, and SysML. Visual Paradigm will be used to draw the Use Case diagrams, and the Domain Model of the Diabetes Management System.

1. Pencil

Pencil is a free and open-source GUI prototyping tool that people can easily install and use to create mock-ups in popular desktop platforms. It will be used to mock-up the graphical user interface (GUI) design of the Diabetes Management System.

1. **PROJECT REQUIREMENTS AND SPECIFICATION**
   1. **THE STAKEHOLDERS**
      1. The Client Dr Shirley Cotter, Manager of Diabetes Clinic Ltd.
      2. The Customer
         1. Manager
         2. Doctor
         3. Nurse
         4. Patient
         5. Secretary
      3. Other Stakeholders
         1. The Cork University Hospital
         2. The Diabetes Day Centre
         3. The Clinical Nutritionist
         4. The Eye Clinic
         5. The Diabetes Federation of Ireland
         6. The Diabetes Ireland Care Centre
      4. Maintenance Users and Service Technicians
         1. The Administrator
         2. IT Department Staff
   2. **PROJECT CONSTRAINTS**
      1. **Solution Constraints**

The product shall record patient’s blood glucose level, insulin injections, and blood pressure information and be able to retrieve the data to both the patient and the doctors’ clinic.

* + 1. **Integration Constraints**

Diabetes Management System shall provide an easily-accessible means of managing a Diabetes Clinic and its various patients, doctors, nurses, and secretary. The system is accessible by all users and mediates access via a security system.

The system enables simultaneous multi-user access to a secured central database.

* + 1. **Budget Constraints**

The budget for the project is €80,000, excluding costs incurred for attending regular meetings.

* + 1. **Schedule Constraints**

The system must be ready in 3 months, excluding the period of research. The software application must be implemented and tested before a final session of software testing. Furthermore, the application must work on majority of web-browsers such as **Internet Explorer**, **Google Chrome**, **Mozilla Firefox**, **Opera**, **Netscape**, and **Safari**, installed on any PCs with Windows, Vista, and Linux operating systems.

* 1. **Relevant Facts And Assumptions**
     1. **Relevant Facts**

Diabetes is now the commonest metabolic illness in the world. Meanwhile, Ireland has a growing population of Diabetes and is increasing at an alarming rate in tandem with rising obesity rates.

County Tipperary has the highest rate of diabetes people with 2.5% and County Kerry has the smallest number of diabetes people with 1%.

Moreover, research indicates that 9.6% of Irish population now have the disease. Diabetes increases the risk for many serious health problems such as Heart attack, Stroke, Kidney failure, Blindness, etc. However, with correct treatment and recommended lifestyle changes, many people with diabetes are able to prevent or delay the onset of complications.

* + 1. **Assumptions**

The graphical user interface (GUI) design must be available to the developers.

The chosen business must agree for the software to be developed by signing a contract.

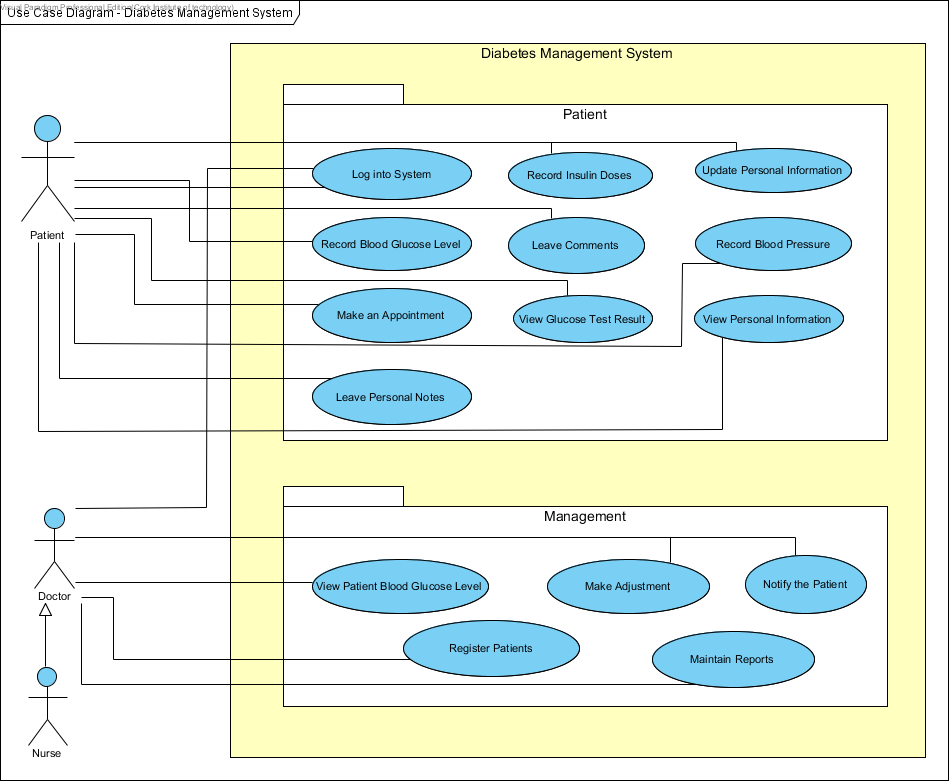
* 1. **PRODUCT BOUNDARY USE CASE DIAGRAM**

The Product Scope diagram identifies the boundaries between the users (Actors) and the Product. This diagram is a summary of all the Product Use Cases.

The Diabetes Management System use case diagram illustrated below shows the actors/users outside the product boundary (i.e. the rectangle). While the product Use Cases (PUCs) are the ellipses inside the boundary. And, the lines denote interfaces between the Product and an Actor. The actors can be either automated or human.

The most important aspect in modelling a system is to capture the dynamic behaviour. In order to capture this, I used Visual Paradigm to model the system of our application. This use case diagram consists of the actors, use cases and their relationships.

The diagram below (**See next page**) is used to gather the functional and non-functional requirements of the Diabetes Management System.



***Use Case Diagram for Diabetes Management System***

* 1. **PRODUCT USE CASE LIST**

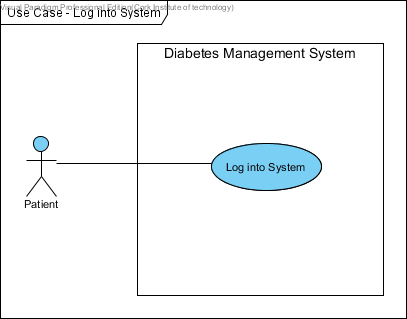
|  |  |
| --- | --- |
| **Use-Case Name** | **Brief Summary of Each Use-Case** |
| Log into System | This use case describes the Login Process. |
| Update Personal Information | This use case describes the event of updating the Patient and Doctor Profile Information. |
| View Personal Information | This use case describes the event of viewing Patients complete Profile Information. |
| Record Blood Glucose Level | This use case describes the event of documenting Patient’s Blood Glucose Level. |
| Record Insulin Doses | This use case describes the event of documenting Patient’s Insulin Doses. |
| View Glucose Test Result | The use case describes the event of viewing blood glucose level tests. |
| Record Blood Pressure | This use case describes the event of documenting Patients’ blood pressure. |
| Make an Appointment | This use case describes the event of making an appointment to the Doctor/Nurse and keeping a record of any appointments with the Doctor. |
| Leave Personal Notes | This describes the process of leaving personal drafts, appointment dates, emergency contacts, etc. |
| Leave Comments | This use case describes the process of leaving special events/comments based on the Patient’s Blood Glucose Level measurement. |
| View Patient Blood Glucose Level Measurements | This use case describes the event of viewing Patient’s Blood Glucose Tests. |
| Register Patients | This use case describes the process of registering Patients into the system. |
| Make Adjustments | This use case describes the event of making adjustments to the Insulin Doses. |
| Notify Patients | This use case describes the event of notifying Patients to come in for Check-ups. |
| Maintain Reports | This use case describes the event of recording and maintaining any information such as Patient’s record of blood sugar readings, record of Insulin adjustments, record of blood pressure, list of Insulin doses, etc. Also, it includes adding and deleting reports. |

* 1. **PRODUCT USE CASES NARRATIVE**

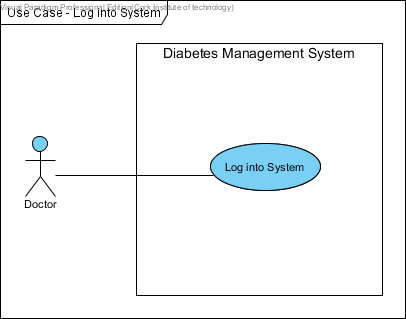
**Use-Case Narrative (Login)**

|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **1** | **Log into System** | This use-case describes the Log in process |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Log into System | |
| **Scenario:** | User wants to log into the Diabetes Management System with user credentials (e.g. username and password). | |
| **Event (Trigger):** | User wants to enter into the Diabetes Management System to view or update their profile. | |
| **Brief Description:** | User provides required credentials. System validates credentials and logs user into the Diabetes Management System. | |
| **Actors:** | Patient, Doctor, Nurse. | |
| **Type:** | Essential | |
| **Preconditions:** | Patient, Doctor and Nurse are known by the System | |
| **Post conditions:** | Patient, Doctor and Nurse logged into the system. | |
| **Flow of Events**  **(Steps):** | **Actor Action** | **System Response** |
| 1. This use case begins when a user desires to log into the system. |  |
| 1. The user provides the username and password. | * 1. System validates the entered username and password and logs the user into the system. |
| **Alternative Flow of Events** | | |
| **Line 2.1:** Invalid username and password entered. Indicate error. Return to Step 2. | | |



***Patient Log into System***

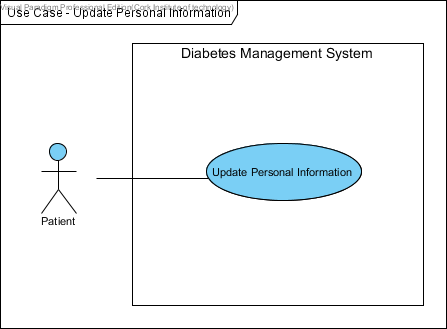


***Doctor Log into System***

**Use-Case Narrative (Update Personal Information)**

|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **2** | **Update Personal Information** | This use-case describes the event of updating profile information. |

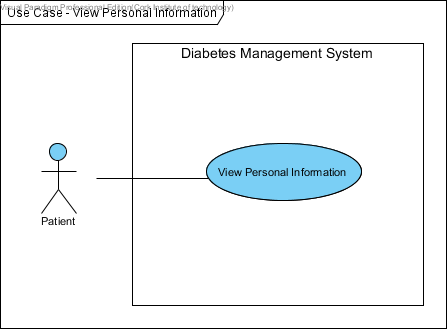
|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Update Personal Information | |
| **Scenario:** | User wants to update his/her profile in the system. | |
| **Event (Trigger):** | Update personal information. | |
| **Brief Description:** | User provides information (e.g. name, address, telephone number). System updates the personal information. | |
| **Actors:** | Patient, Doctor and Nurse. | |
| **Type:** | Essential | |
| **Preconditions:** | Patient, Doctor and Nurse are known by the System  Patient, Doctor and Nurse information must exist. | |
| **Post conditions:** | Patient, Doctor and Nurse information updated in the system. | |
| **Flow of Events**  **(Steps):** | **Actor Action** | **System Response** |
| 1. This use case begins when a user desires to update their personal information in the system. |  |
| 1. The user provides the username and password. | 2.1 System validates username and password and logs the user into the system.  2.2 System displays user’s personal information. |
| 1. The user modifies their personal information. (E.g. address, name, telephone number). | 3.1 System updates and displays updated user’s personal information. |
| **Alternative Flow of Events** | | |
| **Line 2.1:** Invalid username and password entered. Indicate error. Return to Step 2. | | |



***Patient Update Personal Information***

**Use-Case Narrative (View Personal Information)**

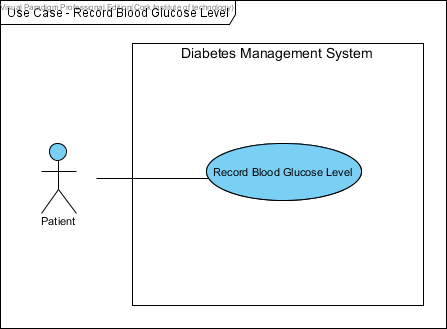
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **3** | **View Personal Information** | This use case describes the event of patients viewing their profile information. |



***Patient views Personal Information***

**Use-Case Narrative (Record Blood Glucose Level)**

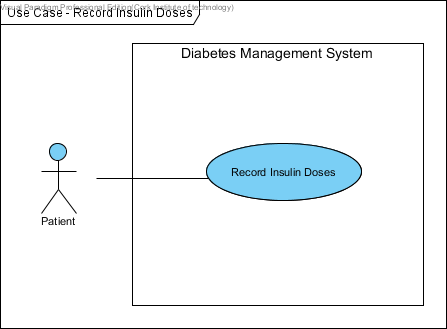
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **4** | **Record Blood Glucose Level** | This use case describes the event of patients tracking their blood glucose level in the system. |



***Patient records blood glucose level***

**Use-Case Narrative (Record Insulin Doses)**

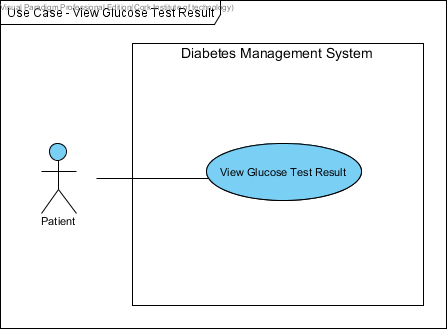
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **5** | **Record Insulin Doses** | This use case describes the process of keeping track of patient’s Insulin Doses. |



***Patient records Insulin Doses***

**Use-Case Narrative (View Glucose Test Result)**

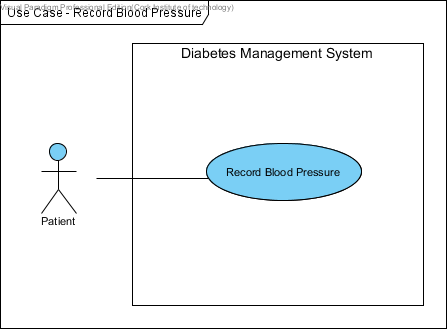
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **6** | **View Glucose Test Result** | The use case describes the event of viewing blood glucose level tests. |



***Patient views Glucose Test Result***

**Use-Case Narrative (Record Blood Pressure)**

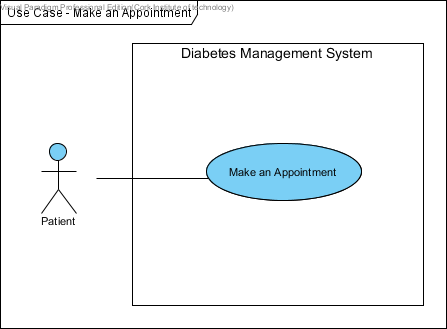
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **7** | **Record Blood Pressure** | The use case describes the event of keeping track of blood pressure. |



***Patient records Blood Pressure***

**Use-Case Narrative (Make an Appointment)**

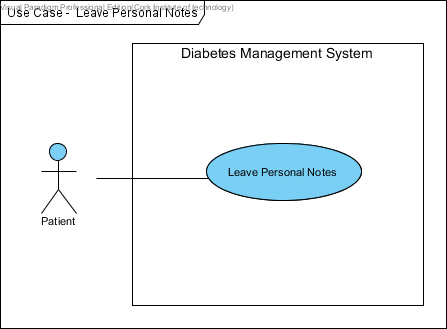
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **8** | **Make an Appointment** | The use case describes the event of making an appointment. |



***Patient makes an Appointment***

**Use-Case Narrative (Leave Personal Notes)**

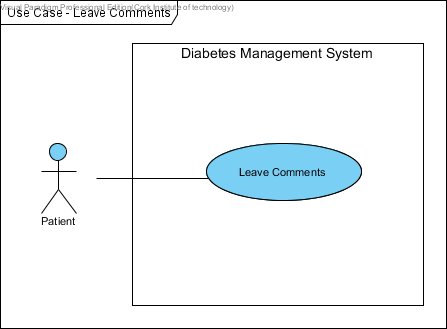
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **9** | **Leave Personal Notes** | This describes the event of leaving personal drafts, appointment dates, emergency contacts, etc. |



***Patient leaves Personal Notes***

**Use-Case Narrative (Leave Comments)**

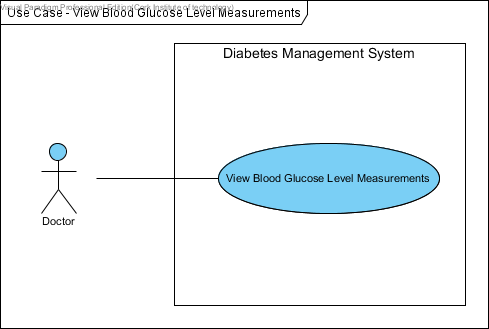
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **10** | **Leave Comments** | This describes the event of leaving special events or comments. |



***Patient leaves Comments***

**Use-Case Narrative (View Blood Glucose Level Measurements)**

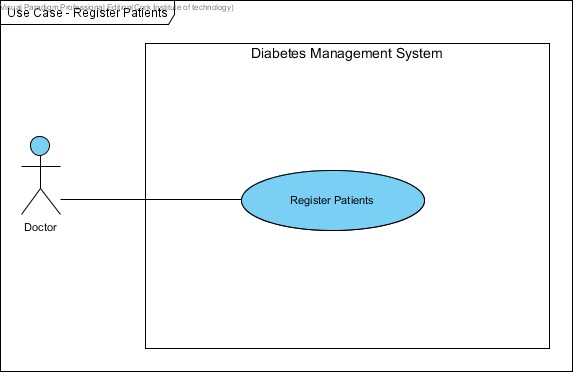
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **11** | **View Blood Glucose Level Measurements** | The use case describes the event of viewing Patient’s Blood Glucose Level Tests. |



***Doctor views Blood Glucose Level Measurements***

**Use-Case Narrative (Register Patients)**

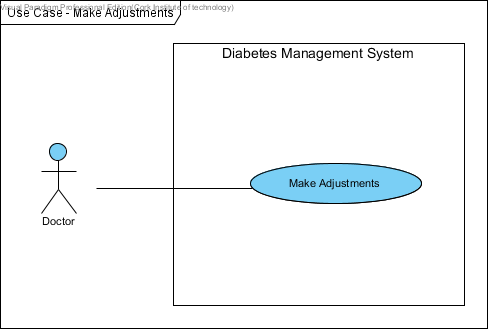
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **12** | **Register Patients** | This describes the process of registering Patients into the Diabetes Management System |



***Doctor registers Patients***

**Use-Case Narrative (Make Adjustments)**

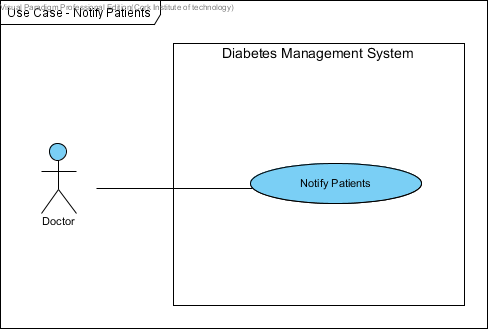
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **13** | **Make Adjustments** | The use case describes the event of making an adjustment to the Insulin Doses. |



***Doctor makes Adjustments***

**Use-Case Narrative (Notify Patients)**

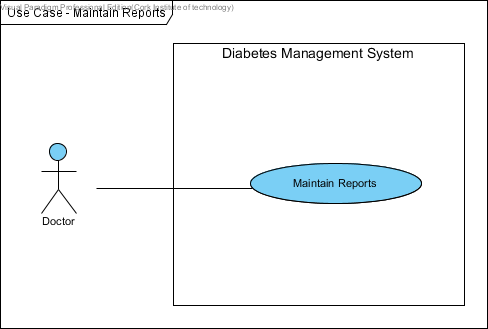
|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **14** | **Notify Patients** | The use case describes the event of notifying Patients to come in for Check-ups. |



***Doctor notifies Patients***

**Use-Case Narrative (Maintain Reports)**

|  |  |  |
| --- | --- | --- |
| **Use-Case #** | **Use-Case Name** | **Full Description of Use-Case** |
| **15** | **Maintain Reports** | This use case describes the event of recording and maintaining any information such as Patient’s record of blood sugar readings, record of Insulin adjustments, record of blood pressure, list of Insulin doses, etc. Also, it includes adding and deleting reports. |



***Doctor maintains Reports***

* 1. Summary of Prototyping
     1. **Functional Requirements** 
        1. The system shall allow the Patient and Doctor to log in with their username and password.
        2. The system shall enable the Patient to view their Personal Information.
        3. The system shall allow the Patient to update their Personal Information.
        4. The system shall enable the Patient to view their Glucose Test Results.
        5. The system shall record Patient Insulin Doses.
        6. The system shall record Patient Blood Glucose Level.
        7. The system shall record Patient Blood Pressure.
        8. The system shall allow the Patient to make an appointment.
        9. The system shall allow the Patient to leave Personal Notes.
        10. The system shall allow the Patient to leave Comments.
        11. The system shall enable the Doctor to view the Patient Blood Glucose Level Measurements.
        12. The system shall allow the Doctor to register Patients into the system.
        13. The system shall allow the Doctor to make adjustments.
        14. The system shall allow the Doctor to notify Patients.
        15. The system shall enable the Doctor to maintain Reports.
        16. The system shall enable copies of Patient Information, Blood Glucose Level tests to be printed.
     2. **Non-Functional Requirements** 
        1. **Operational Requirements**
           1. The system shall run on web browsers, desktop PCs and mobile devices, to be used by stakeholders.
           2. The system shall interface with the business management system.
           3. The system shall be flexible and easily adaptable to changes, saving up to 50 percent of its operating cost.
           4. The system shall be a platform and accessible from any web browsers, mobiles and from any remote areas.
        2. **Performance Requirements**
           1. The system shall support management staff of approximately 150 management people.
           2. The system shall enable patients to update their personal information and blood glucose level results within 2 minutes.
           3. The user shall get a system response within 10 seconds.
           4. The Doctor shall be able to register up to 100 diabetic patients in the system.
           5. Responses to report queries shall take less than two seconds.
        3. **Security Requirements**
           1. No patient can access any other patient’s personal information or blood glucose results.
           2. The system shall be secured to prevent unauthorized access of patient data.
        4. **Availability Requirements**
           1. Information about blood glucose results shall be available at any time to the patients.
           2. A list of diabetic patients shall be made available at any time to the Doctor.
  2. **DATA DICTIONARY**

This data dictionary is a textual list of all concepts that are defined during the analysis. The aim of the data dictionary is to define a vocabulary that will be common to all the users of the Diabetes Management System.

|  |  |
| --- | --- |
| **Name** | **Description** |
| **Patient ID** | This is a unique automated Identification Number, which is assigned to each patient when registered by the system. This will enable them to log in into the system. |
| **Username** | This is an Identification Name of the user account name. This can be combination of the user names and any other names or special characters the account will know. |
| **Password** | The password enforces the privacy’s of the users account. It is used by the user to log into the System. |
| **First Name** | The first name of the users as known to the System. |
| **Last Name** | The user’s last name as known to the System. |
| **Address** | This is the current postal address of the users, These are mostly the home or office address of the user. |
| **Phone Number** | This is the contact phone number of the user. |
| **Email Address** | This is active and current email address of the user. The system uses automate email to send password to the user on registration. |
| **DOB** | DOB is the date of birth of the users. |
| **Gender** | It is the sex of the user of the application. |
| **Diabetes** | This is a disorder of the metabolism causing excessive thirst and the production of large amounts of urine. |
| **Height** | This is the height of the patient. |
| **Kgs** | This measure the unit of the weight. |
| **BMI**  **(Body Mass Index)** | This is to calculate the body mass of the patient. It is calculated by the formula: ((weight \* 703)/ (height \* height)). The gauge is used to determine whether or not a person is overweight or obese. |
| **Medical History** | This is the medical history of the patient such as: Allergies, heart condition and general patient’s health status. |
| **Blood Pressure** | This is the pressure of the blood in the circulatory system. Also, it is measured for diagnosis since it is closely related to the force and rate of the heartbeat and the diameter and elasticity of the arterial walls. |
| **Blood Glucose Level** | This is the amount of glucose in the blood. Glucose is a sugar that comes from the foods we eat and it’s also formed and stored inside the body. |
| **Insulin Doses** | This is the number of Units given in each injection (Pre-Breakfast, Pre-Lunch, Pre-Dinner, and Before Bed). |
| **Comments** | These are special events that happen when the patient takes the insulin injections. The patient can leave comments such as: “hype during the night”. |
| **Personal Notes** | This is some text written by the patient that can contain information such as appointment dates, checklist of for diabetes day care unit. It is updated frequently. |
| **Image** | This is a picture of the user. |
| **Aerobic Exercise** | Activity involving large muscles, done for an extended period of time. Aerobic exercise can be used for weight loss. Examples of aerobic exercise include walking, biking, jogging, swimming, aerobic classes and cross country skiing. This information may be used via ‘Tips’ that are shown on the form when a member logs on. These are randomly generated from a list of tips on the server. |
| **Database Security** | Connecting directly to a database leaves the system vulnerable to hackers. We reference ‘Database Security’ as connecting to our database by first sending our information through a script and then to the database. |
| **Database** | This is where data is stored in the system such as patients’ data. |
| **Doctor** | This is a user that is qualified to treat people who are ill. Doctor can notify patient when their blood sugar level readings are low. |
| **Nurse** | This is someone trained to take care of the sick or infirm especially in a hospital. |
| **Patient** | A user of the system that can view and update their statistics. |
| **User ID** | Special number to identify the different users of the system. |
| **Log on** | An interface where the user must input their username and password before they can use the application. It is the bridge between the system and the users. |
| **Performance Graph / Light** | A graph representation to illustrate patients’ blood sugar level performance. |
| **Register Screen** | When the patient details are entered into the system, they are then registered in the system. |
| **Diabetic Patient** | A person that has diabetes. |

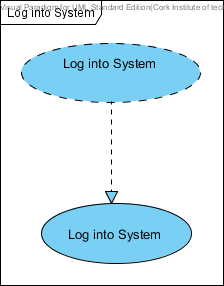
* 1. **COLLABORATION DIAGRAM**

**Collaboration Diagrams**

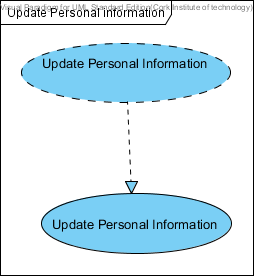
Collaboration Diagrams are used to model an alternate view of the Sequence Diagram. It shows how objects involved in a situation interact with each other, instantiating a particular class in the Diabetes Management System. Also, it gives the modeller the ability to show detail, such as visibility. Collaboration diagrams are used to understand all of the effects on any given object and are also used for procedural design.

Collaboration diagrams are used to model every scenario found in the Diabetes Management System.

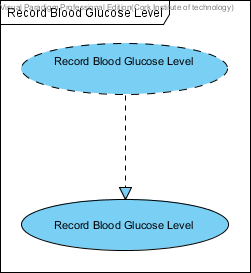
The use case diagram **Log into System** is collaborating with the Use case Log into System.



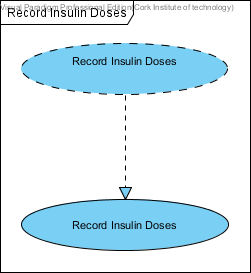
The use case diagram **Update Personal Information** is collaborating with the Use case Update Personal Information.



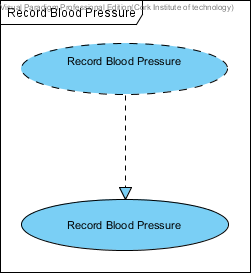
The use case diagram **Record Blood Glucose Level** is collaborating with the Use case Record Blood Glucose Level.



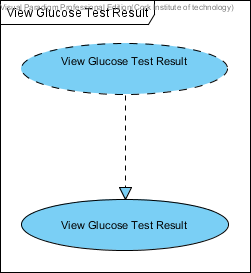
The use case diagram **Record Insulin Doses** is collaborating with the Use case Record Insulin Doses.



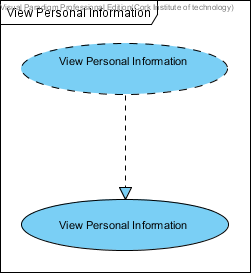
The use case diagram **Record Blood Pressure** is collaborating with the Use case Record Blood Pressure.



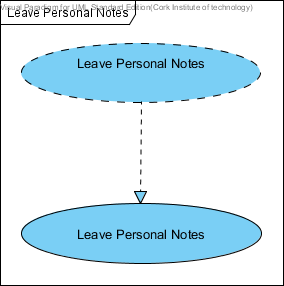
The use case diagram **View Glucose Test Result** is collaborating with the Use case View Glucose Test Result.



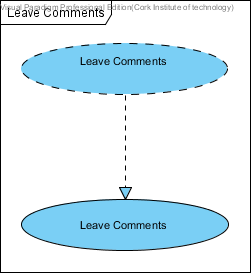
The use case diagram **View Personal Information** is collaborating with the Use case View Personal Information.



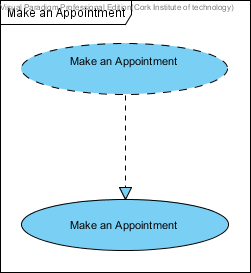
The use case diagram **Leave Personal Notes** is collaborating with the Use case Leave Personal Notes.



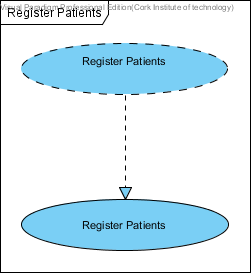
The use case diagram **Leave Comments** is collaborating with the Use case Leave Comments.



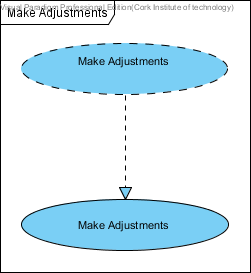
The use case diagram **Make an Appointment** is collaborating with the Use case Make an Appointment.



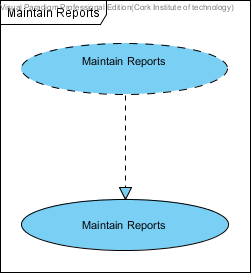
The use case diagram **Register Patients** is collaborating with the Use case Register Patients.



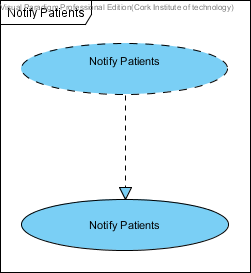
The use case diagram **Make Adjustments** is collaborating with the Use case Make Adjustments.



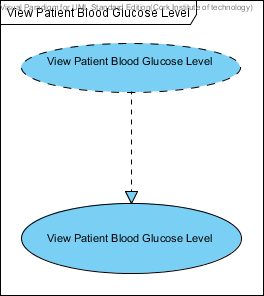
The use case diagram **Maintain Reports** is collaborating with the Use case Maintain Reports.



The use case diagram **Notify Patients** is collaborating with the Use case Notify Patients.



The use case diagram **View Patient Blood Glucose Level** is collaborating with the Use case View Patient Blood Glucose Level.

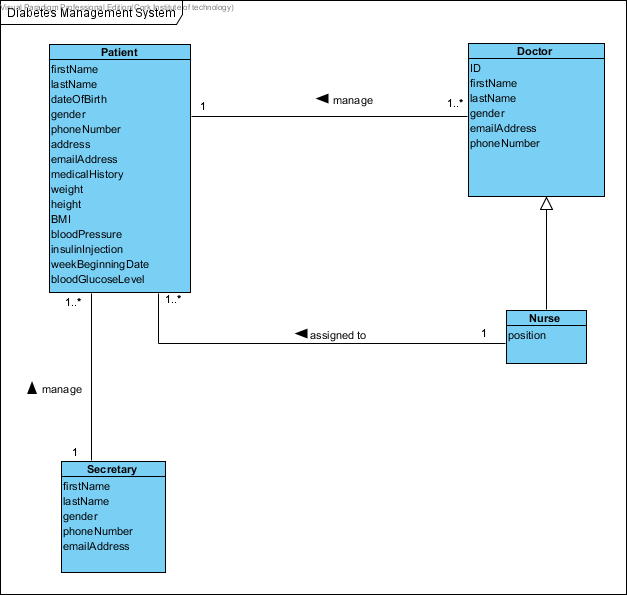


* 1. **BUSINESS DATA MODEL**

**Class Diagrams**

Class Diagrams in UML (Unified Modelling Language) is a static diagram that describes the structure of the Diabetes Management System being modelled. It shows a collection of the System’s Classes (i.e. Class Names), their Attributes, Operations, and the Relationships among Objects. Moreover, it is the main building block of object-oriented modelling. Class diagrams can be used to illustrate data model of the Diabetes Management System.

Class Diagrams will be used to model the general static view of the Diabetes Management System and detailed modelling that translate the models into programming codes.



***Domain/Conceptual Model for Diabetes Management System***

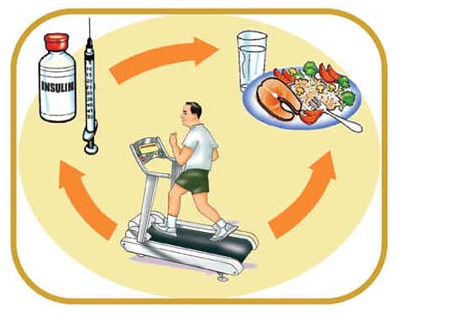
1. **CONCLUSIONS**

In conclusion, diabetes is a very serious disease. The reason is that, the body loses its ability to control the level of glucose (sugar) in the blood. Patients with diabetes control their diabetes by controlling what they eat, checking their blood glucose (sugar) several times a day, and most of all by exercising. However, if diabetes is not controlled, diabetes can lead to serious complications such as blindness, kidney failure, amputations, and can even lead to death. Additionally, it is easier to prevent diabetes than to treat it, as it is said; “**Prevention is better than cure”**.

Diabetes can be prevented or delayed by:

* Losing excess weight and belly fat, if the individual is overweight.
* Becoming more active.
* Eating healthy.
* Reducing your blood pressure if it is high.
* Lowering your blood sugar level if you have pre-diabetes.

Therefore, by adopting to these lifestyle changes, people with diabetes will be able to lower their risk of developing diabetes and its serious complications. Moreover, a prevention plan can help in reducing the risks for diabetes and heart disease while helping the individual to enjoy a healthier lifestyle.



Furthermore, the Diabetes Management System (DMS) will assist in the management of diabetes. It will provide a means of tracking, organising, and managing all of the activities involved in the monitoring of diabetes. DMS will work in parallel with the existing paper-based system, which will make it possible to track patient’s data stored in the paper-based system i.e. the Diabetes Monitoring Diary. The diabetes management system will provide an easy to use GUI (Graphical User Interface) containing all the features of the system. Most importantly, provide a helpful **user-manual** for the end-users.

In summary, the Diabetes Management System (DMS) will help in the management of the following use cases:

* The creating/updating/deleting of patients in the system.
* The tracking of all patients’ information.
* Recording appointment date and time for the patients to come in for check-ups.
* Enabling doctors to print blood sugar level reports, insulin injection details, patients’ blood pressure records, changes to patient’s insulin doses, and the blood glucose testing statistics.
* Charts prepared to look for patterns and improve insulin delivery.
* Sending registration information to patients via email.
* Enabling patients to leave comments and personal notes.

# **BIBLIOGRAPHY**

American Diabetes Association, (2015). Complications. [online] Available at: <http://www.diabetes.org/living-with-diabetes/complications/> [Accessed 6 Jan. 2015].

American Diabetes Association, (2015). Statistics About Diabetes. [online] Available at: <http://www.diabetes.org/diabetes-basics/statistics/> [Accessed 6 Jan. 2015].

Diabetes Ireland, (2015). Managing Your Diabetes - Diabetes Ireland. [online] Available at: <http://www.diabetes.ie/living-with-diabetes/living-type-1/managing-diabetes/> [Accessed 6 Jan. 2015].

Diabetes.niddk.nih.gov, (2015). Causes of Diabetes. [online] Available at: <http://diabetes.niddk.nih.gov/dm/pubs/causes/#what> [Accessed 6 Jan. 2015].

Glooko.com, (2015). Glooko | Mobile Diabetes Management | Blood Glucose Log. [online] Available at: <https://glooko.com/> [Accessed 6 Jan. 2015].

Healthline.com, (2015). The 11 Best Diabetes iPhone & Android Apps of 2014. [online] Available at: <http://www.healthline.com/health-slideshow/top-iphone-android-apps-diabetes#2> [Accessed 6 Jan. 2015].

Healthline.com, (2015). The 11 Best Diabetes iPhone & Android Apps of 2014. [online] Available at: <http://www.healthline.com/health-slideshow/top-iphone-android-apps-diabetes#2> [Accessed 6 Jan. 2015].

Intelihealth.com, (2015). Health Problems That Diabetes Can Cause - Intelihealth. [online] Available at: <http://www.intelihealth.com/article/health-problems-that-diabetes-can-cause> [Accessed 6 Jan. 2015].

Mayoclinic.org, (2015). Blood glucose meter: How to choose - Mayo Clinic. [online] Available at: <http://www.mayoclinic.org/diseases-conditions/diabetes/in-depth/blood-glucose-meter/art-20046335?pg=1> [Accessed 6 Jan. 2015].

Melissa Conrad St&ouml;ppler, M. (2015). Diabetes (Type 1 and Type 2) Symptoms, Causes, Treatment - What is diabetes? - MedicineNet. [online] MedicineNet. Available at: <http://www.medicinenet.com/diabetes_mellitus/page2.htm#what_is_diabetes> [Accessed 6 Jan. 2015].

Nuigalway.ie, (2015). November 2012, New Diabetes Research Centre for the West : About Us, NUI Galway. [online] Available at: <http://www.nuigalway.ie/about-us/news-and-events/news-archive/2012/november2012/new-diabetes-research-centre-for-the-west-.html> [Accessed 6 Jan. 2015].

Roshdypharmacies.com, (2015). Preventing Diabetes. [online] Available at: <http://www.roshdypharmacies.com/healthcare-details.php?HC_ID=112> [Accessed 8 Jan. 2015].

Visual-paradigm.com, (2015). Creating class diagrams in Visual Paradigm. [online] Available at: <http://www.visual-paradigm.com/support/documents/vpuserguide/94/2576/7190_creatingclas.html> [Accessed 6 Jan. 2015].