**A**

**Project Report**

**on**

**“Diabetes Monitoring Application (Dia-Mon)”**

Developed by:

**Anuj Gajjar:166120316013**

**Moksh Modi:166120316033**

Guided by:

**Ms. Suhani Chauhan**

A Project report submitted to Gujarat Technological University

In completion of the requirements of Diploma Engineering

In Information Technology

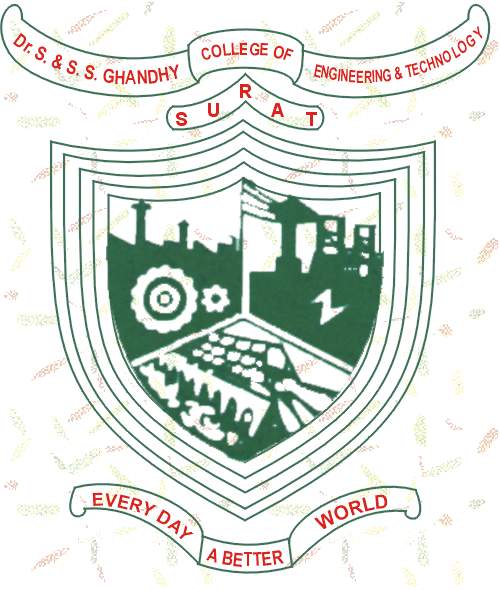
## **April 2019**

## 

## **Dr. S & S.S. GHANDHY COLLEGE OF**

## **ENGINEERING & TECHNOLOGY**

## **SURAT**



## **CERTIFICATE**

This is to certify that project work embodied in this report entitled **Diabetes Monitoring Application** was carried out by **Mr. Anuj Gajjar Himanshu (166120316013) and Mr. Moksh Modi Chirag (166120316033)** at **Dr. S. & S.S. Ghandhy College of Engineering and Technology, Surat.**

For partial successful completion of **Project-I (3351605)** of **Semester-6 Diploma engineering in Information Technology** to be awarded by Gujarat Technological University. This project work has been carried out under my supervision and is to the satisfaction of the department.

**Date:**

**Place:** Surat

|  |  |  |  |
| --- | --- | --- | --- |
| **GUIDE** | **H.O.D.** | **PRINCIPAL** | **EXTERNAL EXAMINER** |

## **DECLARATION OF ORIGINALITY**

We hereby certify that we are sole author of this project and that neither any part of the work nor the whole work has been submitted for degree to any other University or Institution.

We certify to the best of our knowledge, our work does not infringe upon anyone’s copyright nor violates any proprietary right and that ideas, techniques, quotation, or any other material from the work of other people included in our report or otherwise, are fully acknowledge in the accordance with the standard reference practices.

We declared that is a true copy of our report, included any final revision, as approved by our supervisor.

**Date:**

**Place:** Surat

Your Sincerely,

**Anuj Gajjar Himanshu (166120316013)**

**Moksh Modi Chirag (166120316033)**

## **ACKNOWLEDGEMENT**

The final outcome of this project needs lot of guidance and assistance from many people and we are extremely privileged to have this all along the completion of the project. All that we have done is been only possible due to such supervision and assistance and we would not forget to thanks them.

We heartily thank our internal project guide**. Ms. Suhani Chauhan (Lecturer in I.T. Department),** for his guidance and suggestions during this period. We would not forget to remember **Mr. J.R. Mahida** for this encouragement and more over for his timely support and guidance till the completion of our project work. the project would not have been completed without continuous support, motivation extended by our colleague and friends who were always with us whenever required.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all teaching staff of **I.T. Department** which helped us in successfully completing our project work.

Last but not least, thanks to **Dr. S & S.S. Ghandhy College of Engineering & Technology** for providing us the platform to represent the project

Your Sincerely,

**Anuj Gajjar Himanshu (166120316013)**

**Moksh Modi Chirag (166120316033)**

## **TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Title Page |  |
| Certificate Page |  |
| Declaration of Originality page |  |
| Acknowledgement |  |
| Table of Contents |  |
| Table of Contents |  |
| List of tables |  |
| List of figures |  |
| Abstract |  |

|  |  |
| --- | --- |
| **Chapter 1 Introduction** | 1 |
| * 1. Specification | 2 |
| * 1. Overview of Language | 3 |
| * 1. Scope of Project | 4 |
| **Chapter 2 Requirement Analysis** | 5 |
| 2.1. Requirement Analysis | 6 |
| 2.2. Problem Solving Technology | 8 |
| 2.3. Project Life Cycle Model | 8 |
| 2.4. Software Requirement Specification | 9 |
| **Chapter 3 System design** | 10 |
| 3.1. Organization Chart | 11 |
| 3.2. E-R Diagram | 12 |
| 3.3. Data Flow Diagram | 13 |
| 3.4. Data Dictionary | 18 |
| **Chapter 4 UML** | 23 |
| 4.1. Use Case Diagram | 24 |
| 4.2. Sequence Diagram | 25 |
| 4.3. Class Diagram | 26 |
| 4.4. Activity Diagram | 27 |
| **Chapter 5 GUI / Application Interface** | 31 |
| **Chapter 6 Testing** | 37 |
| **Chapter 7 Advantages and Disadvantages** | 42 |
| **Chapter 8 Conclusion** | 43 |
| **Chapter 9 Future Enhancement** | 44 |
| **Bibliography** | 45 |

## **List of Tables**

|  |  |
| --- | --- |
| * + 1. Admin\_Info | 18 |
| * + 1. Doctor\_login | 18 |
| * + 1. Doctor\_lnfo | 19 |
| * + 1. Doctor\_Clinic\_lnfo | 19 |
| * + 1. Doctor\_Persnal\_login | 20 |
| * + 1. Medical\_Store | 20 |
| * + 1. patient\_login | 21 |
| * + 1. Patient\_lnfo | 21 |
| * + 1. Doctor\_Payment\_Info | 22 |

## **List of Figures**

|  |  |
| --- | --- |
| Figure 2.3 Incremental Model | 8 |
| Figure 3.1 Organization chart | 11 |
| Figure 3.2 ER Diagram | 12 |
| Figure 3.3.1 DFD Level-0 | 13 |
| Figure 3.3.2 DFD Level-1 | 14 |
| Figure 4.1 Use Case Diagram | 24 |
| Figure 4.2 Sequence Diagram | 25 |
| Figure 4.3 Class Diagram | 26 |
| Figure 4.4 Activity Diagram | 27 |
| Figure 5.1 Splash Screen Activity | 32 |
| Figure 5.2 Login Page | 33 |
| Figure 5.3 Sign-Up Page | 34 |
| Figure 5.4 Glucose Monitoring | 35 |
| Figure 5.5 Glucose Monitoring History | 36 |

# **Abstract**

Diabetes Monitoring Application Will Allow the User or Patients to Monitor Their Blood Sugar Level Daily. the Application Will Then Produce Charts Based On Their Daily Blood Sugar Level. the Application Will also Allow the Doctor to Monitor the Patient’s Daily Blood Sugar Level. Moreover, the Application Will Feature Support From the Patient’s Dietician and also Feature A Medical Store for the Patient’s Medical Supplies. the Application Will also Have A “**Doctor Alert Feature**” to Alert the Doctor About the Patient’s Increasing Blood Sugar Level.

# Project Purpose:

The Purpose of the Application is to Provide Support for the Diabetic Patients and Help Them Monitor Their Diabetes Better. the Focus of the Application is to Help Patients with Little or No Knowledge About the Diabetes Controlling, with Help From Their Doctor and Dietician.

# Project Objectives:

* Daily Blood Sugar Level Monitoring
* Rich UI and Charts From Blood Sugar Level Inputs
* Support From the Patient's Doctor
* Support From the Patient’s Dietician
* Online Medical Store for Patient’s Supplies
* Doctor Alert System

## **Chapter - 1**

Introduction

|  |  |
| --- | --- |
| **Sub-Topics** |  |
| 1.1 Specification of Project | 2 |
| 1.2 Overview of Language | 3 |
| 1.3 Scope of project | 4 |

# 1.1 Project Specifications

The Required Hardware and Software Specifications are

1. Server PC:-

* Core i5 or Xeon Processor
* 16 GB RAM
* 1 TB HDD
* Windows Server 2010 or Linux Server OS

1. Phone or Handset:-

* Snapdragon 425 or Mediatek Processor
* 1 GB RAM
* 300 MB for Storage
* Android OS 6 or Above

1. Software Requirements:-

* Android Studio 3.0
* WAMP Server
* Android Phone with Version 6 or Above

# 1.2 Overview of Language

The Languages Used In the Project are

* for Backend:-

1. **PHP**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

What is PHP Used for?

* for Making Backend APIs
* for Database Connectivity
* for Admin Panel
* for Frontend :-

1. **Android**

Android OS is the One of the Most Popular Mobile OS, It is Developed by Google Under OHA(Other Handset Alliance).Nearly,70% of the World’s Phone Uses the Android OS, and the Rate is Increasingly Day by Day.

What is Android Used for?

* for Making Rich UI
* Easy Implementation
* Support for Large Amount of Devices
* Secure
* for Database:-

1. **MySQL**

MySQL is a powerful Relational Database Management System (RDBMS) which we will use to learn the basic principles of database and data manipulation using Structured Query Language (SQL) statements. SQL is a database language that is used to retrieve, insert, delete and update stored data. This is achieved by constructing conditional statements that conform to a specific syntax

What is MySQL Used for?

* Used for Creating RDBMS Table
* has Strict Query and Syntax
* Secure and Easy to Use
* Support A Wide Range of Servers

# 1.3 Scope of Project

The Project is Targeted for the Patients with Type 1 Diabetes, and also for Patients Diagnosed Recently with Diabetes. the Application also has A Doctor Module, by Which the Doctor can Monitor the Blood Sugar Level Along with Patient. the Project also has Pharmacy Store for Buying Medical Supplies for Patient

## **Chapter - 2**

Requirement Analysis

|  |  |
| --- | --- |
| **Sub-Topic** |  |
| 2.1 Requirement Analysis | 6 |
| 2.2 Problem Solving Techniques | 8 |
| 2.3 Project Life Cycle Model | 8 |
| 2.4 Software Requirement Specification | 9 |

# 2.1 Requirement Analysis

# 1.Functional requirement

# Client side:

* Login:

There will be a page in which user can access his/her account. It requires email id/user id and password. This page will shows Home page.

* Registration:

At this page user can sign up for account which requires user’s basic information.

* Health details:

Health information technology (HIT) is rapidly changing how health care providers communicate, exchange information, and interact with each other in an organization through the implementations of electronic health records (EHRs).

Diet details means what she/he eat daily? and suger level details requied for first time health checking.

* Log out:

It can be use to logout the client-panel.

# Admin:

* Log In:

It requires user-name and password to log-into his/her account where it should allow admin to manage the user and their activity.

* Employee Registration:

If any other person wants to be admin,then employee Registration is helpful.it is requires all the information of the person.

* Registration request:

It approves all the requests of registration that are applied by user.

* Log out:

It can be use to logout the admin-panel.

# Doctor side:-

**What type of diabetes you have?**

* **There are Following Types of Diabetes:**
* Type 1 Diabetes-Diabetes Mellitus,
* Type 2 Diabetes- Gestational Diabetes,
* Pre Diabetic Conditions,
* Hypertensive Diabetes,
* Diabetic Retinopathy,
* Diabetic Neuropathy,
* Diabetes Nephropathy (Kidney Failure),
* Diabetes with heart Disease.

The results varies on patient condition and cannot be guaranteed.

# 2.Non- Functional requirement

* **Database:-**

the database system SQL should store all the data and activity of the user and admin.

* **Platform:-**

the web-app should run on the different plateform. E.g –windows,linux.

* **Security:-**

This should provide security to hide all user’s data and activity.

* **Reliability:-**

Web-application must reliable for use that can user believe On the software.

# 2.2 Problem Solving Techniques

* Some of the Problems While Developing Diabetes Monitoring Application are

1. **Providing an Easy Interface to Users**

the Application is Targeted for Users with Recently Diagnosed Diabetes As Well As Users with Years of Diabetes. Thus, Providing An Easy Interface is A Tough Task.

**Solution:**

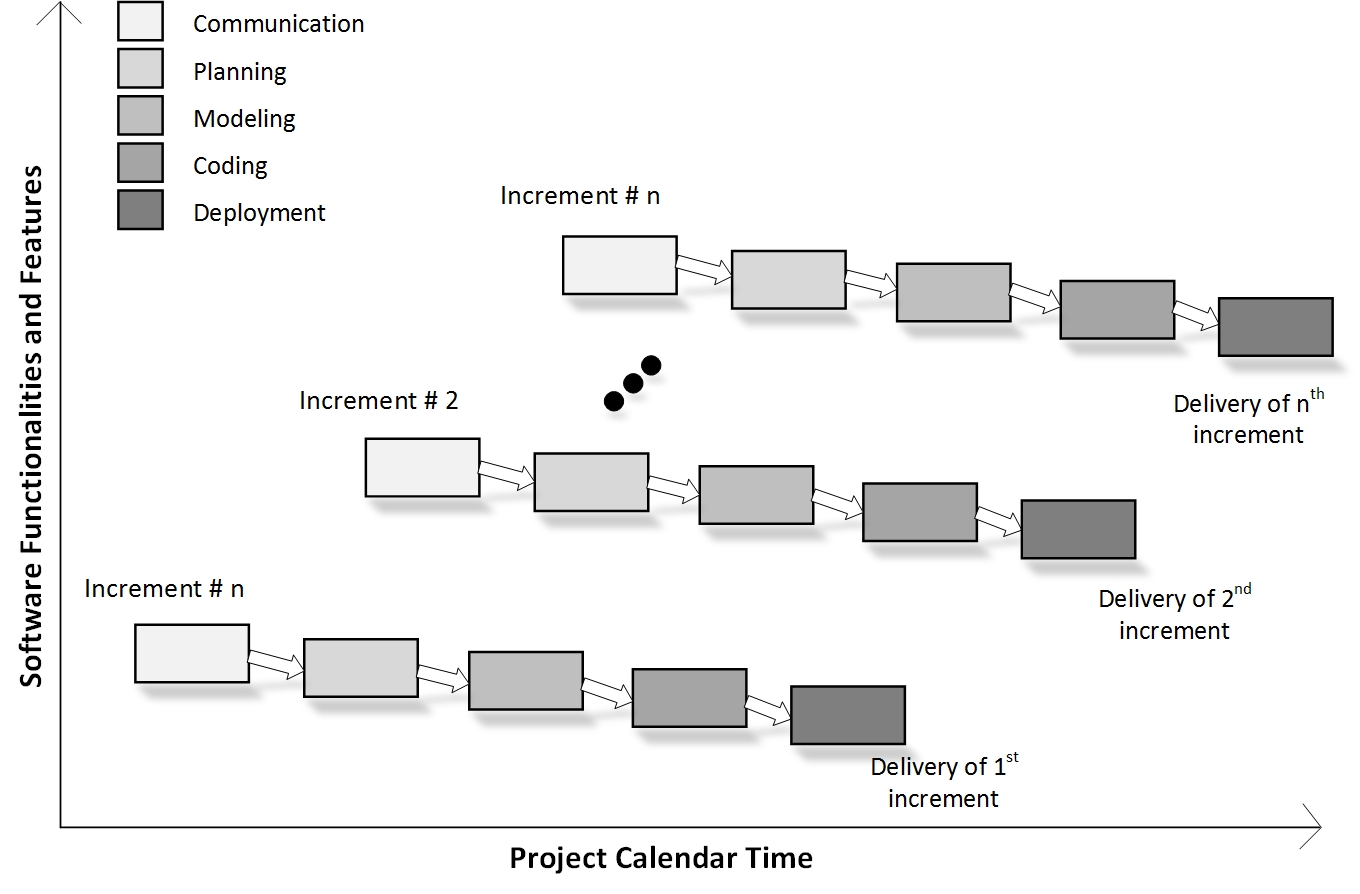
In Order to Ensure That User can Enjoy the Interface We Will Provide A Basic Guide or Tutorial to the New Users.

# 2.3 Project Life Cycle Model

* **The Project Life Cycle Model:-**

We are Using Incremental Model for Our Application Because We A Going to Provide A Basic Functional Application and Later Provide Major Updates. This is Because by This Approach We can First Take the Users Review According to First Release and Then Provide a Better From Their Reviews

* **Incremental Model:**



**Figure 2.3 Incremental Model**

# 2.4 Project Specifications:

**The Required Hardware and Software Specifications are**

1. **Server PC**

* Core i5 or Xeon Processor
* 16 GB RAM
* 1 TB HDD
* Windows Server 2010 or Linux Server OS

1. **Phone OR Handset**

* Snapdragon 425 or Mediatek Processor
* 1 GB RAM
* 300 MB for Storage
* Android OS 6 or Above

1. **Software Requirements**

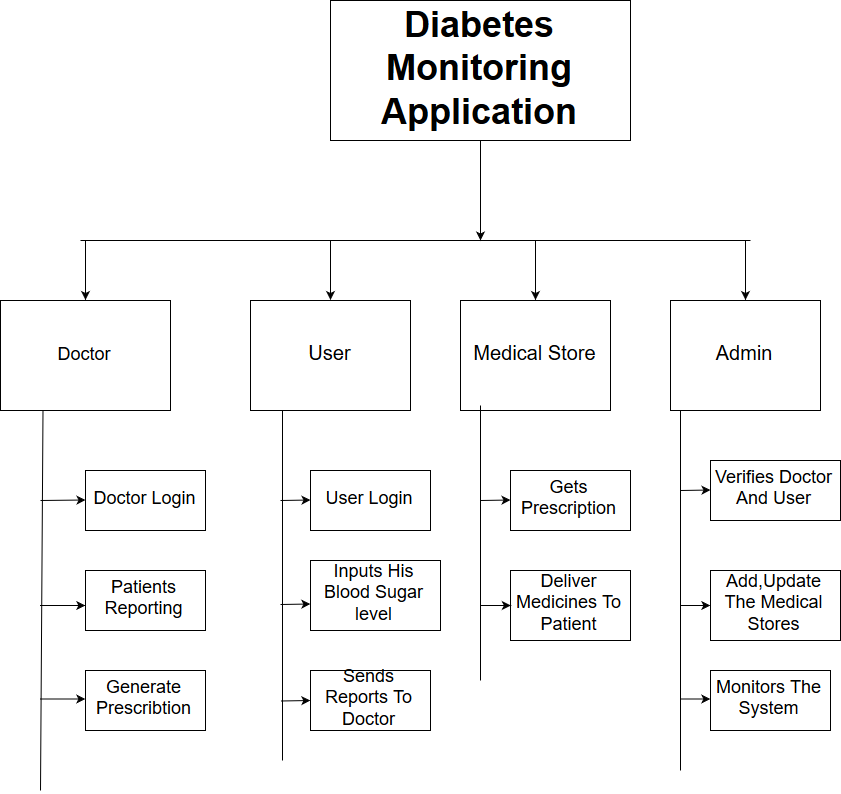
* Android Studio 3.0
* WAMP Server

## Chapter - 3

System Design

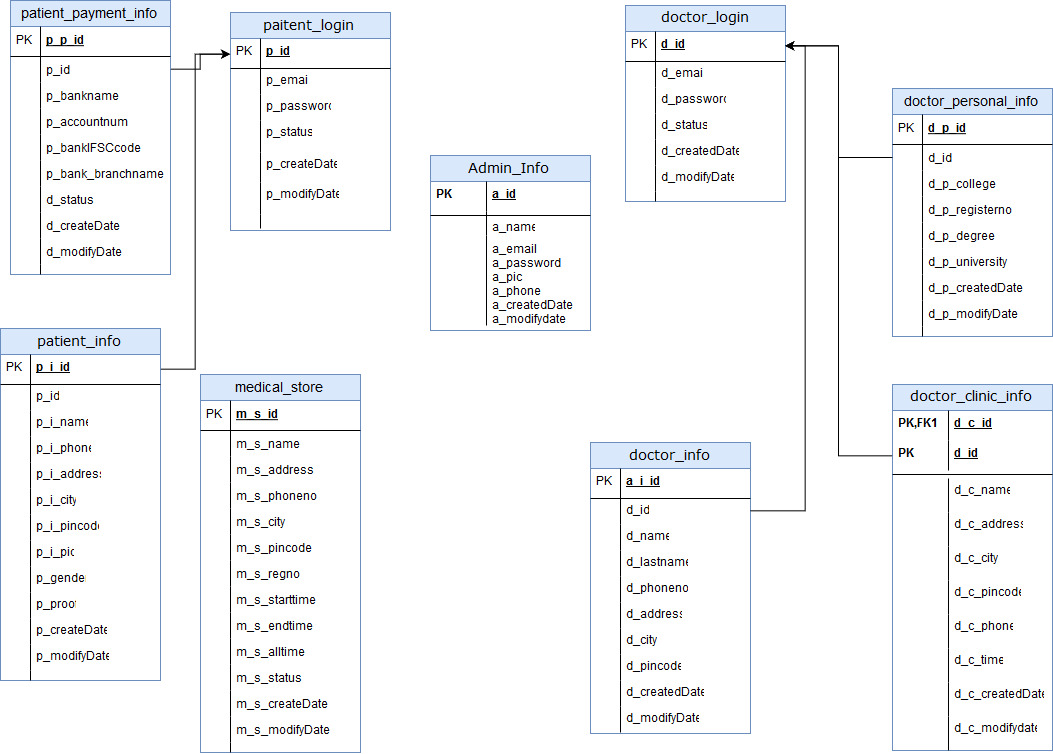
|  |  |
| --- | --- |
| **Sub Topics** |  |
| 3.1 Organization Charts | 11 |
| 3.2 ER Diagram | 12 |
| 3.3 Data Flow Diagram | 13 |
| 3.4 Data Dictionary | 18 |

# 3.1 Organization Chart:



**Figure 3.1 Organization Chart**

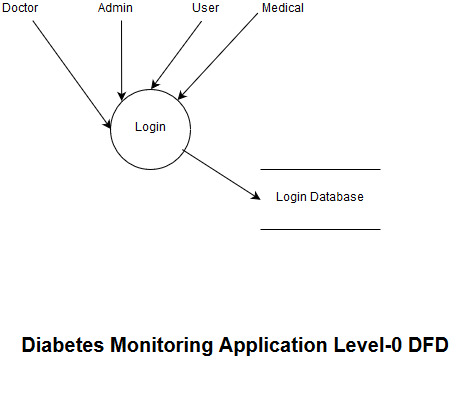
# 3.2 ER Diagram:



**Figure 3.2 ER Diagram**

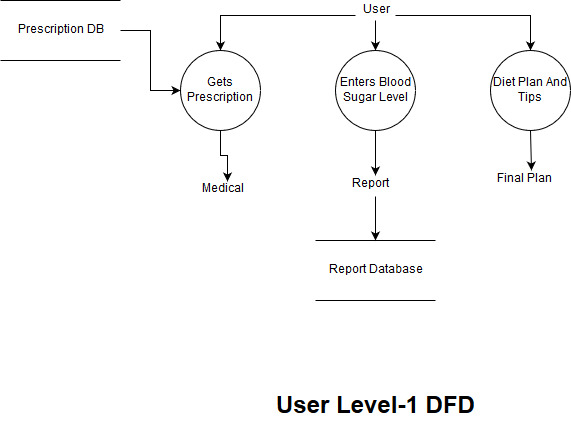
# 3.3 Data Flow Diagram:

* **Level-0 DFD:**



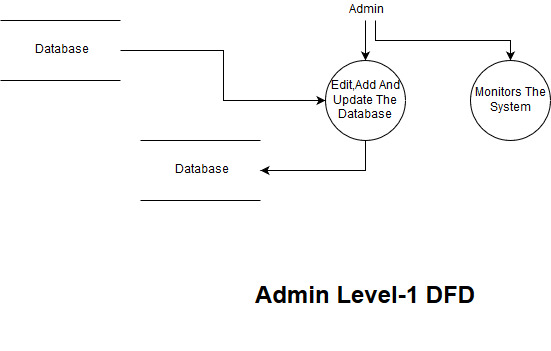
**Figure 3.3 Level-0 DFD**

* **User Level-1 DFD:**



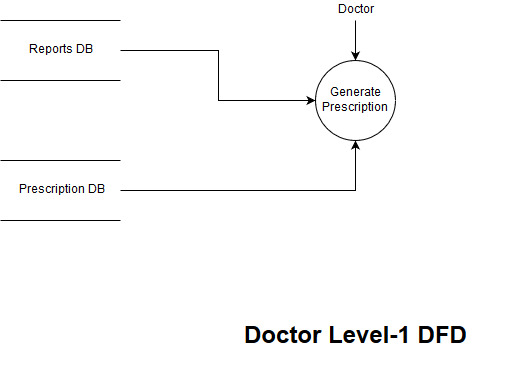
**Figure 3.3.1 User Level-1 DFD**

* **Admin Level-1 DFD:**



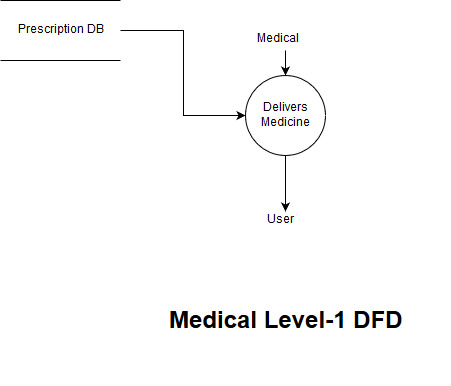
**Figure 3.3.2 Admin Level-1 DFD**

* **Doctor Level-1 DFD:**



**Figure 3.3.3 Doctor Level-1 DFD**

* **Medical Level-1 DFD:**



**Figure 3.3.4 Medical Level-1 DFD**

# 3.4 Data dictionary:

**3.4.1 Admin\_info:-**

**Table 3.4.1 Admin\_info**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| A\_id | Int | Primary | Store id of |
| A\_email | varchar(20) | unique | Store email of |
| A\_password | varchar(20) | - | Store password of |
| A\_mobile no | varchar(20) | unique | Store mobile no of |
| A\_created date | Date | - | Date of Creation |
| A\_modify date | Date | - | Date of Last Modification |

**3.4.2 Doctor\_login:-**

**Table 3.4.2 Doctor\_login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| D\_id | Int | primary | Store id of |
| D\_email | varchar(20) | unique | Store email of |
| D\_password | varchar(20) | - | Store password of |
| D\_status | varchar(20) | - | Store status of |
| D\_created date | Date | - | Date of Creation |
| D\_modify date | Date | - | Date of Last Modification |

**3.4.3 Doctor\_lnfo:-**

**Table 3.4.3 Doctor\_lnfo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| D\_i\_id | Int | primary | Store id of |
| D\_first\_name | varchar(20) | - | Store first name of |
| D\_last\_name | varchar(20) | - | Store last name of |
| D\_id | Int | Foreign | Store id of |
| D\_contect no | varchar(20) | - | Store mobile no of |
| D\_address | varchar(50) | - | Store address of |
| D\_city | varchar(20) | - | Check city name |
| D\_pincode | varchar(10) | - | check location |
| D\_created date | Date | - | Date of Creation |
| D\_modify date | Date | - | Date of Last Modification |

* + 1. **Doctor\_clinic\_lnfo:-**

**Table 3.4.4 Doctor\_clinic\_lnfo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| D\_c\_id | Int | primary | Store id of clinic |
| D\_c\_first\_name | varchar(20) | - | Store first name of clinic |
| D\_c\_last\_name | varchar(20) | - | Store last name of clinic |
| D\_id | Int | Foreign | Store id of |
| D\_c\_contect no | varchar(20) | Unique | Store mobile no of clinic |
| D\_c\_address | varchar(50) | - | Store address of clinic |
| D\_c\_city | varchar(20) | - | Check city name of clinic |
| D\_c\_pincode | varchar(10) | - | check location of clinic |
| D\_c\_time | Date |  | Time of Opening |
| D\_created date | Date |  | Date of Creation |
| D\_modify date | Date |  | Date of Last Modification |

* + 1. **Doctor\_persnal\_login:**

**Table 3.4.5 Doctor\_persnal\_login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| D\_p\_id | Int | primary | Store id of doctor |
| D\_p\_register no | varchar(20) | Unique | Store register no of doctor |
| D\_id | Int | Foreign | Store |
| D\_p\_degree | BLOB | Unique | Degree of Doctor |
| D\_p\_college | varchar(20) |  | College of Doctor |
| D\_p\_university | varchar(20) |  | University of Doctor |
| D\_created date | Date |  | Date of Creation |
| D\_modify date | Date |  | Date of Last Modification |

* + 1. **Medical\_store:**

**Table 3.4.6 Medical\_store**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| m\_s\_id | Int | primary | Store id of |
| M\_s\_name | varchar(20) |  | Store name of |
| M\_s\_reg\_no | varchar(20) | Unique | Store register no of |
| M\_s\_status | varchar(20) |  | Store status of |
| M\_s\_contect no | varchar(20) | Unique | Store mobile no of clinic |
| M\_s\_address | varchar(20) | - | Store address of clinic |
| M\_s\_city | varchar(20) | - | Check city name of clinic |
| M\_s\_pincode | varchar(20) | - | check location of clinic |
| M\_s\_start\_time | Date | - | Working Hours of Shop |
| M\_s\_end\_time | Date | - | Working Hours of Shop |
| M\_s\_all\_time | Date | - | Working Hours of Shop |
| M\_s\_created date | Date | - | Date of Creation |
| M\_s\_modify date | Date | - | Date of Last Modification |

* + 1. **Patient\_login:**

**Table 3.4.7 Patient\_login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| P\_id | Int | primary | Store id of |
| P\_email | varchar(20) | Unique | Store email of |
| P\_password | varchar(20) | - | Store password of |
| P\_status | varchar(20) | - | Store status of |
| P\_created date | Date | - | Date of Creation |
| P\_modify date | Date | - | Date of Last Modification |

* + 1. **Patient\_lnfo:**

**Table 3.4.8 Patient\_lnfo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| P\_i\_id | Int | primary | Store id of |
| P\_i\_first\_name | varchar(20) |  | Store first name of |
| P\_i\_last\_name | varchar(20) |  | Store last name of |
| P\_id | Int | Foreign | Store id of |
| P\_i\_contect no | varchar(20) | Unique | Store mobile no of |
| P\_i\_address | varchar(20) |  | Store address of |
| P\_i\_city | varchar(20) |  | Check city name |
| P\_i\_pincode | varchar(20) |  | check location |
| P\_i\_pic | BLOB |  | Profile Pic of Patient |
| P\_gender | varchar(20) |  |  |
| P\_proof | BLOB |  | Proof of Patient |
| P\_created date | Date |  | Date of Creation |
| P\_modify date | Date |  | Date of Last Modification |

* + 1. **Doctor\_payment\_info:**

**Table 3.4.9 Doctor\_payment\_info**

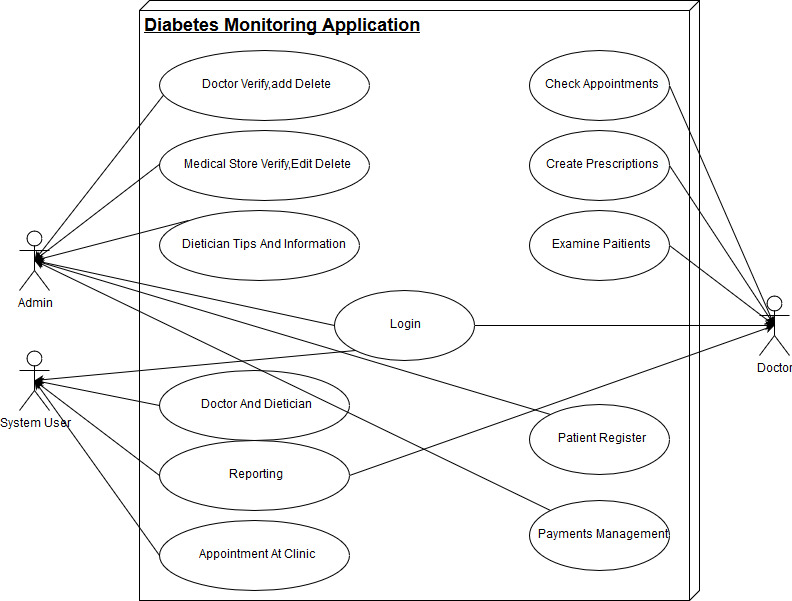
|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Key** | **Description** |
| D\_p\_id | Int | primary | Store id of |
| D\_p\_bankname | varchar(20) |  | Store bank name of |
| D\_id | Int | Foreign | Store |
| D\_p\_bank  account\_no | varchar(20) | Unique | Account no. of Doctor |
| D\_p\_bank | varchar(20) |  | Account no. of Doctor |
| D\_p\_bank  branch\_name | varchar(20) |  | Account no. of Doctor |
| D\_p\_status | varchar(20) |  | Account no. of Doctor |
| D\_created date | Date |  | Date of Creation |

## **Chapter - 4**

**UML**

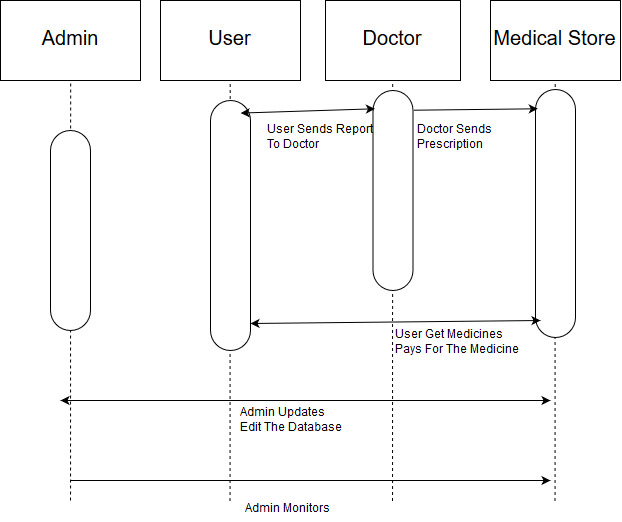
|  |  |
| --- | --- |
| **Sub-Topic** |  |
| 4.1 Use Case Diagram | 24 |
| 4.2 Sequence Diagram | 25 |
| 4.3 Class Diagram | 26 |
| 4.4 Activity Diagram | 27 |

# 4.1 Use Case Diagram:

****

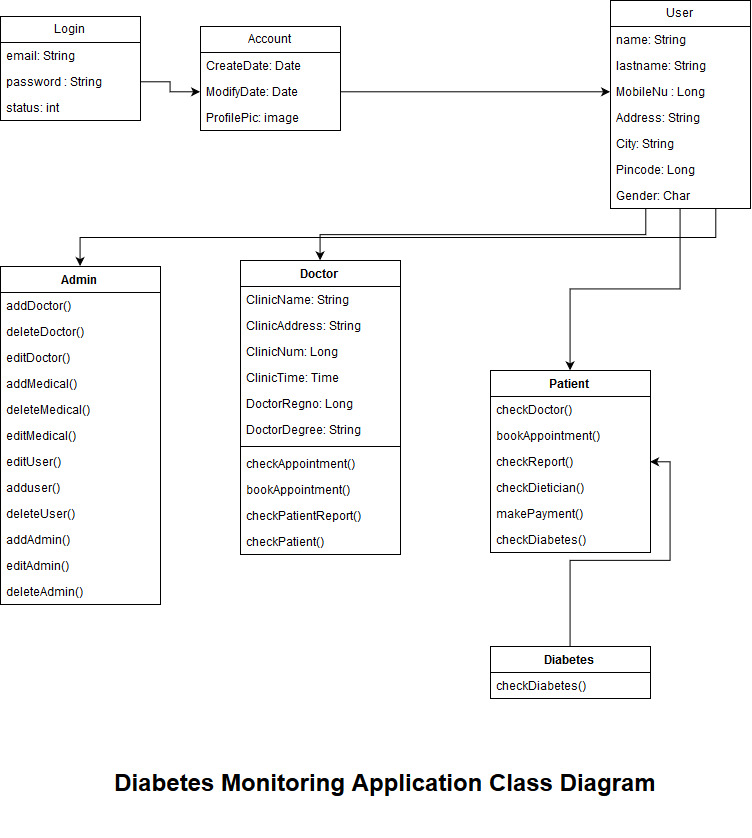
**Figure 4.1 Use Case Diagram**

# 4.2 Sequence Diagram:



**Figure 4.2 Sequence Diagram**

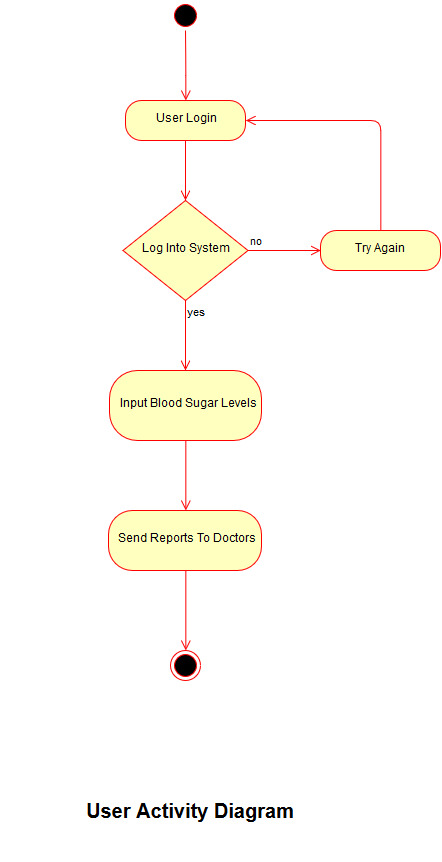
# 4.3 Class Diagram:

****

**Figure 4.3 Class Diagram**

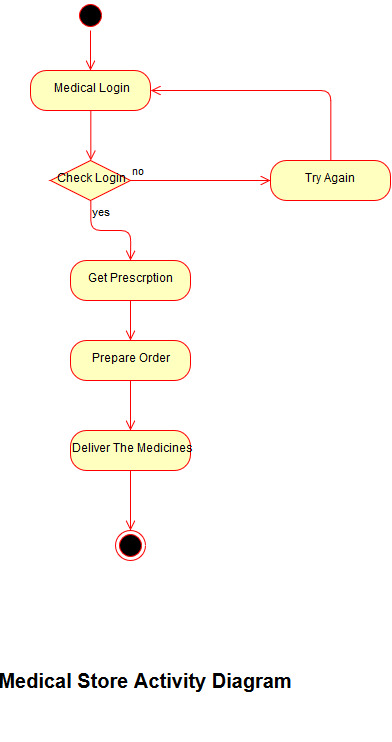
# 4.4 Activity Diagram:

* **User Activity Diagram:-**

****

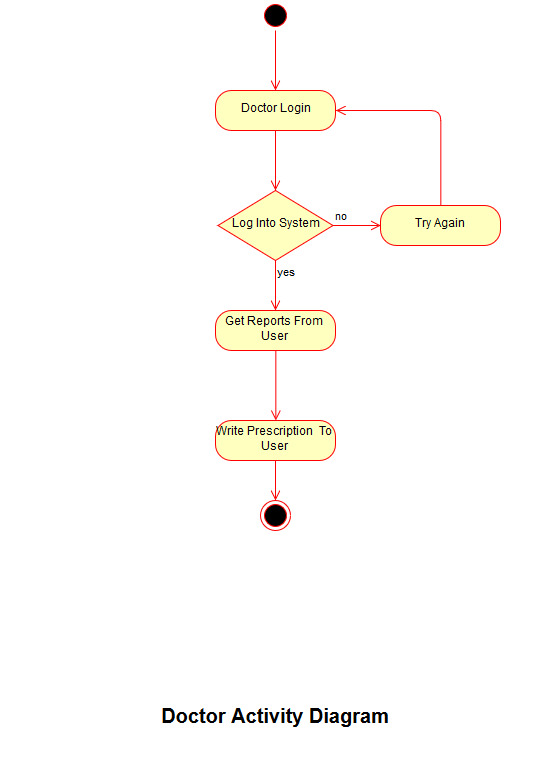
**Figure 4.4 User Activity Diagram**

* **Medical Activity Diagram:**

****

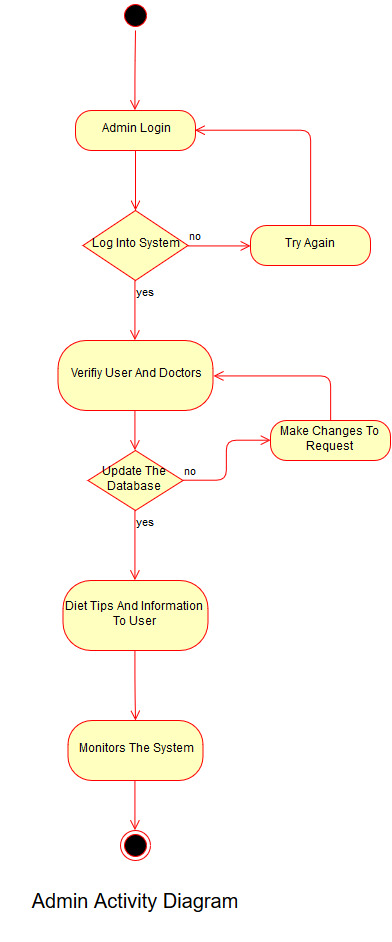
**Figure 4.4.1 Medical Activity Diagram**

* **Doctor Activity Diagram:**

****

**Figure 4.4.2 Doctor Activity Diagram**

* **Admin Activity Diagram:**

****

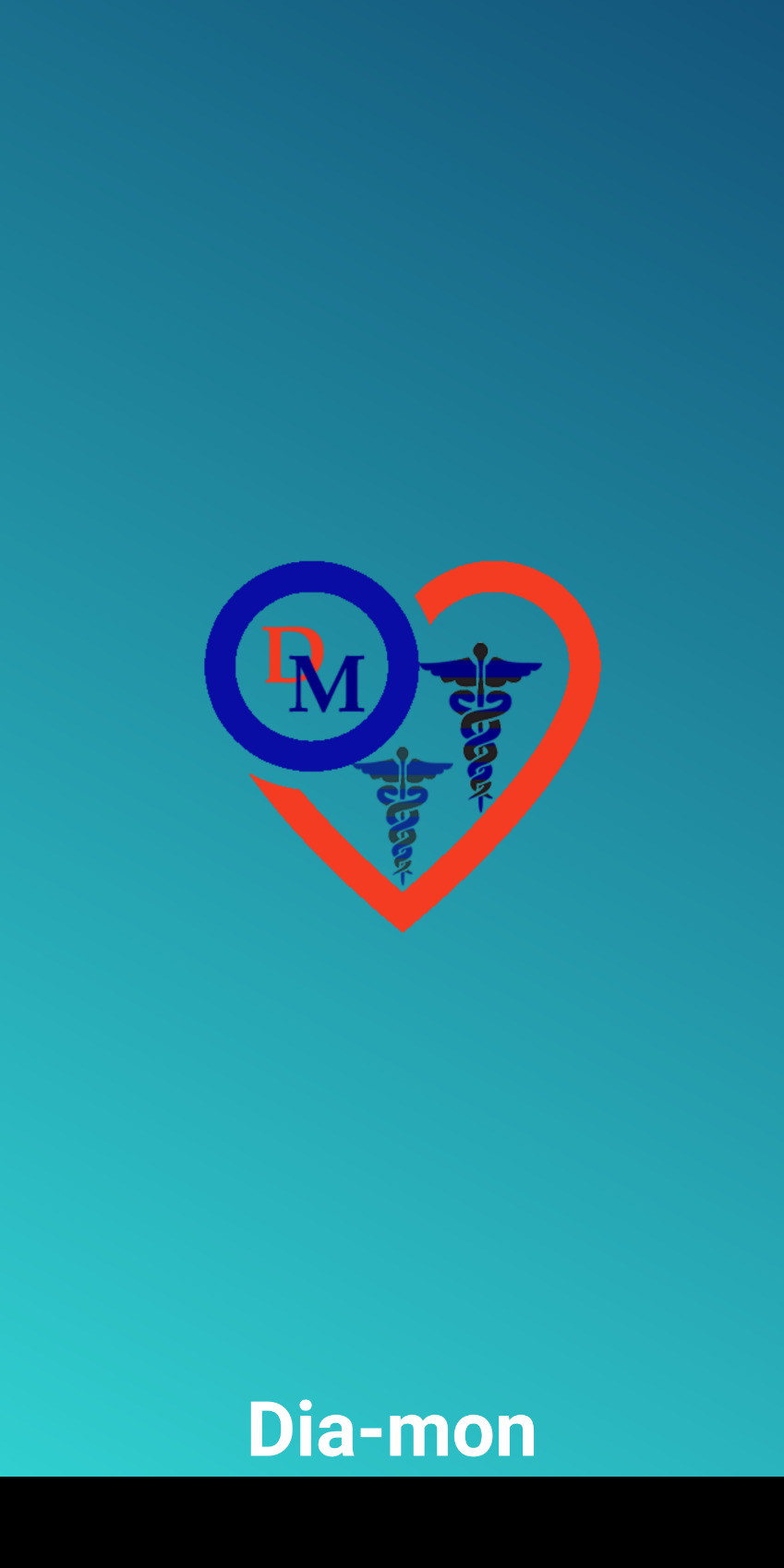
**Figure 4.4.3 Admin Activity Diagram**

## **Chapter - 5**

GUI / Application Interface

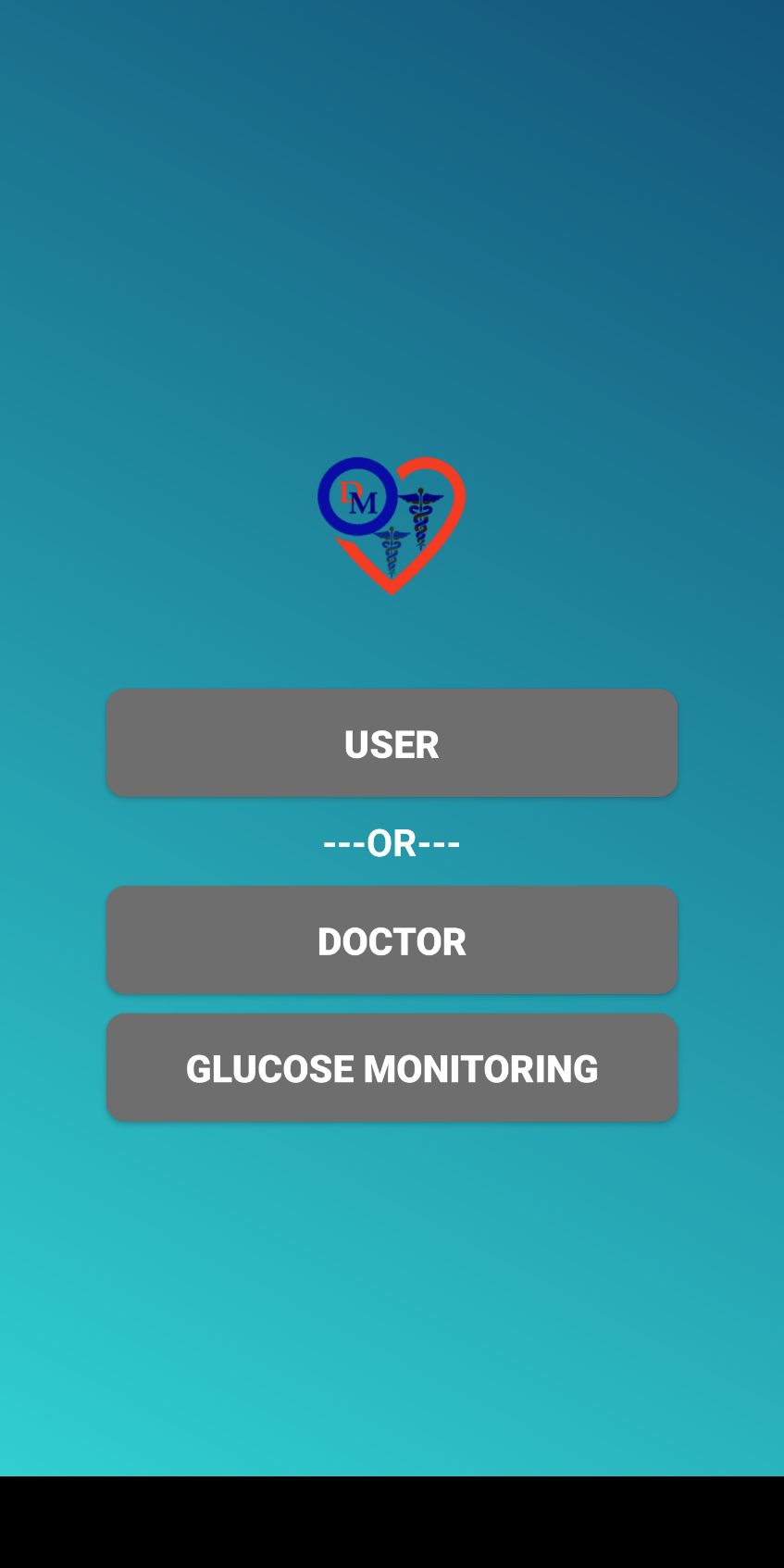
|  |  |
| --- | --- |
| **Sub-Topics** |  |
| Figure 5.1 Splash Screen | 32 |
| Figure 5.2 Login Page | 33 |
| Figure 5.3 Sign-Up Page | 34 |
| Figure 5.4 Glucose Monitoring | 34 |
| Figure 5.5 Glucose Monitoring History | 36 |
|  |  |

# 5.1 Splash Screen of Application:



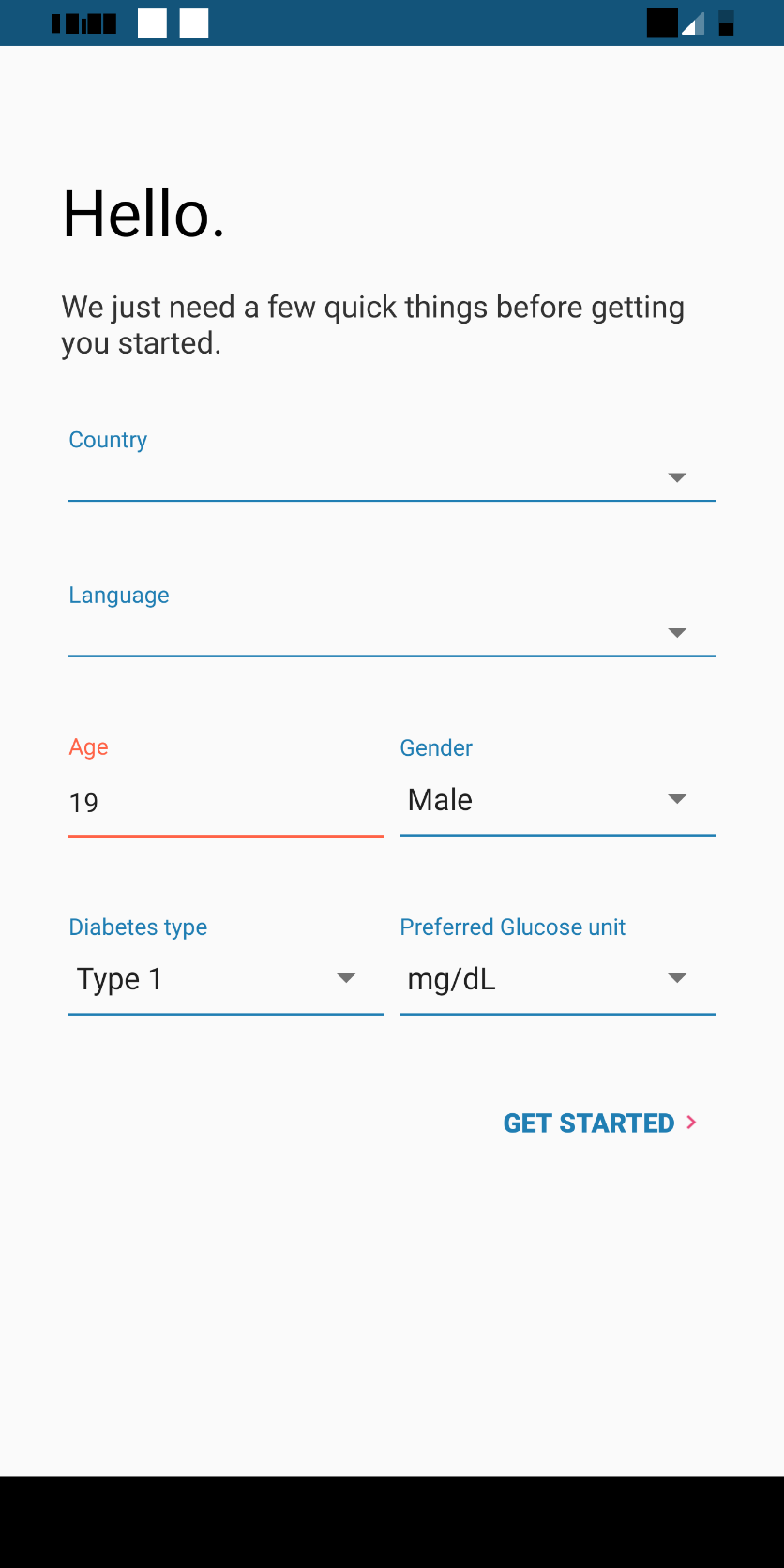
**Figure 5.1 Splash Screen**

# 5.2 Login Page of Application:



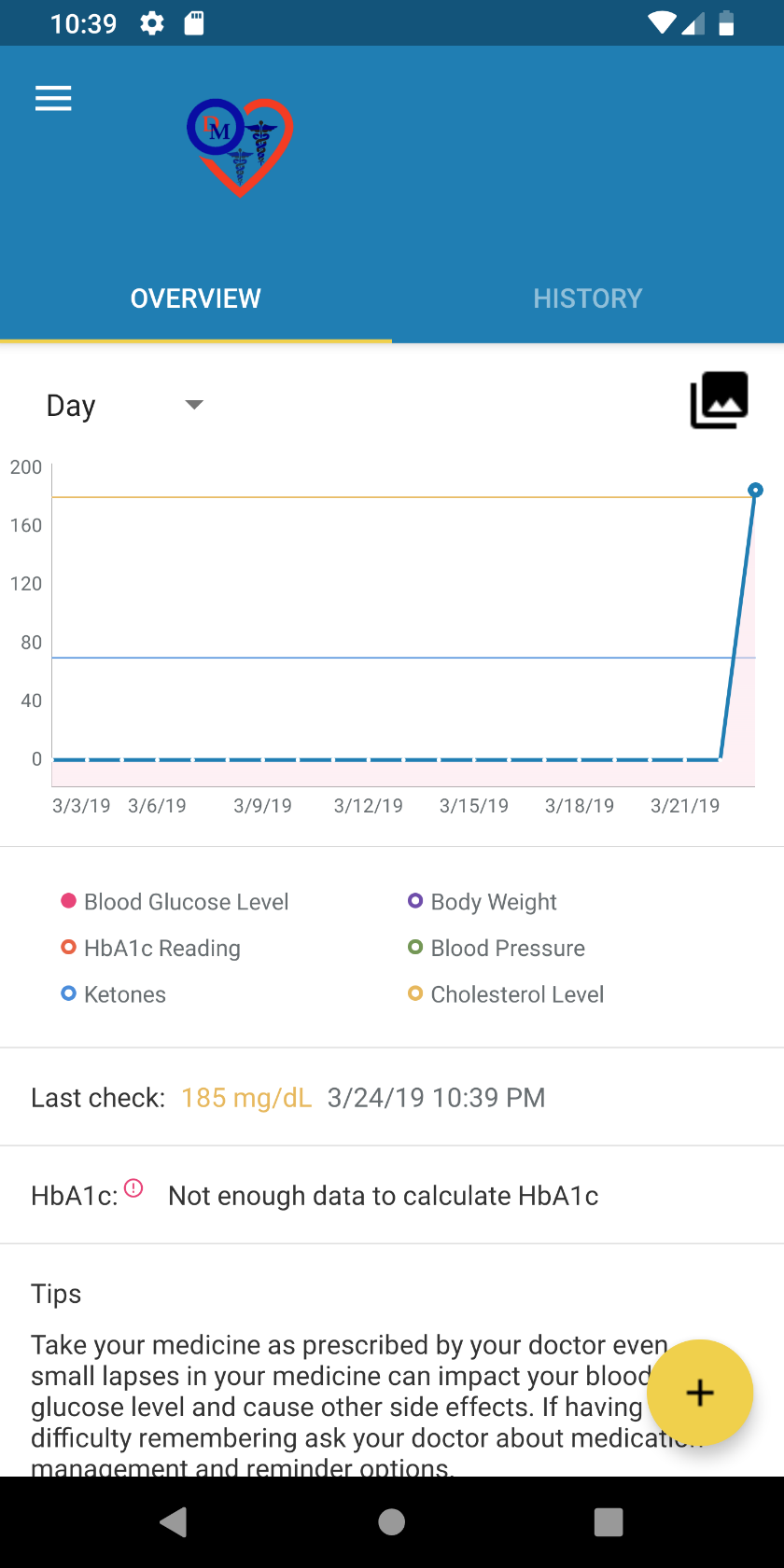
**Figure 5.2 Login Page**

# 5.3 Sign Up Page of Application:



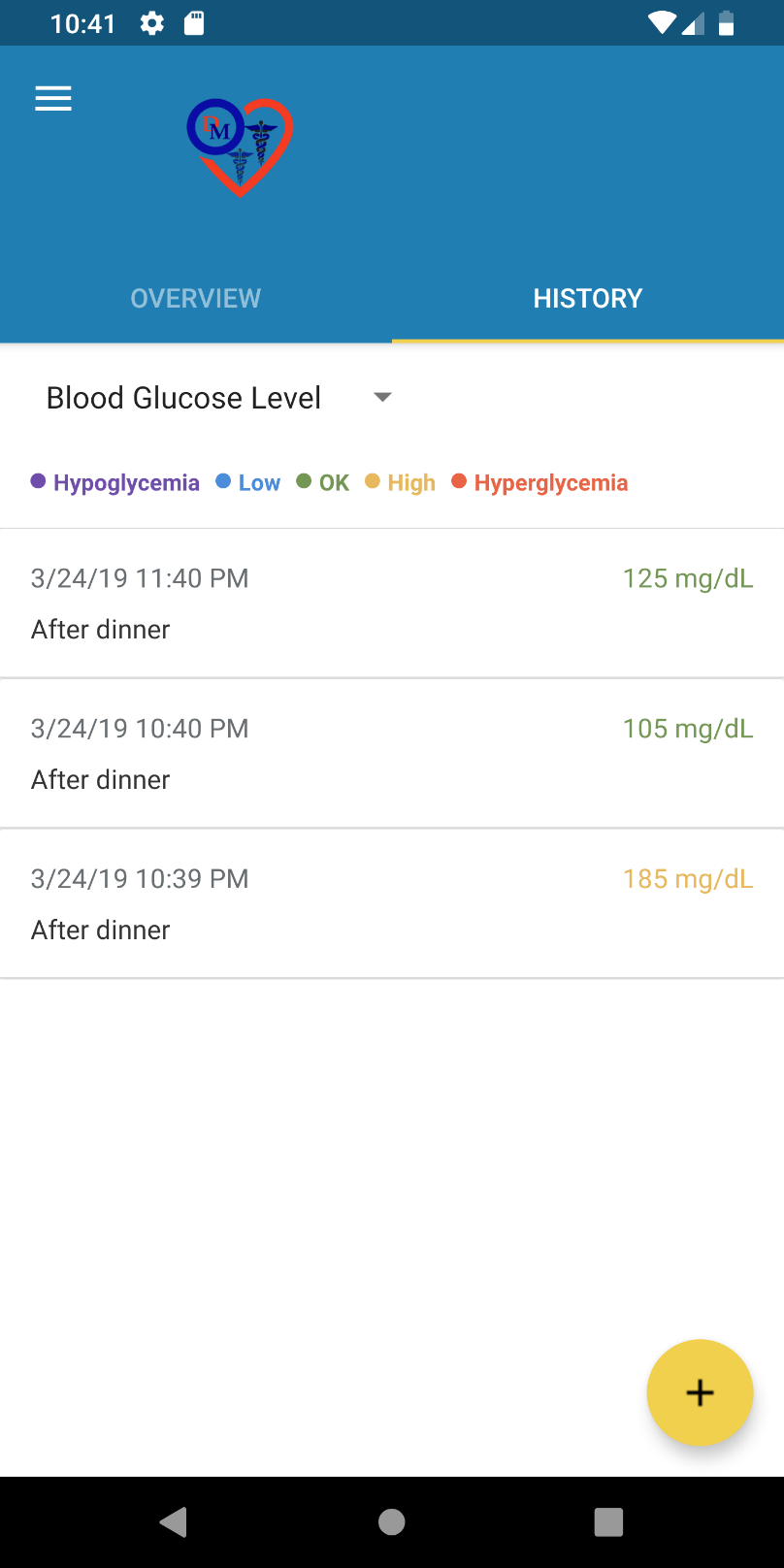
**Figure 5.3 Sign Up Page**

# 5.4 Glucose Monitoring of Application:



**Figure 5.4 Glucose Monitoring**

# 5.5 Glucose Monitoring History:



**Figure 5.5 Glucose Monitoring History**

## **Chapter - 6**

Testing

|  |  |
| --- | --- |
| **Sub-Topics** |  |
| 6.1 Unit Testing | 38 |
| 6.2 Integration Testing | 38 |
| 6.3 System Testing | 39 |
| 6.4 Black Box Testing | 39 |
| 6.5 White Box Testing | 40 |
|  |  |

# 6.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. The unit test is white-box oriented, and the step can be conducted in parallel for multiple components.

We performed Unit testing in our system for test of small modules like Login Module, Registration Module, Report Generation, etc.

Unit testing can be performed from the bottom-up, starting with smallest and lower level modules and proceeding one at a time, for each module in bottom-up testing a short program is used to execute the module and provide the needed data, so that module is asked to performed the way it will when embedded within the larger system.

With unit testing, we generally performed the white box testing in our all modules (Login Module, Registration Module, Report generation, etc). In these all modules, we check all coding and syntax of modules. So that unit testing is useful in our system.

# 6.2 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting the tests to uncover errors associated with interfacing. In these, we are combining all the unit modules and test these all at the same time.

In our system, we integrate all the modules (Login Module, Registration Module, Report Module, Report Generation, etc.) are performed the integration testing on all those.

In our system, we used two techniques in Integration Testing:

* **Top Down Integration:**

Combine the test and debug top level routines that become the integration test harness for low level units.

* **Bottom-Up Integration:**

Bottom-Up Integration testing begins construction and testing with atomic modules. Because components are integrated from the bottom up, processing required for components subordinate to a given level is always available and the need for stubs is eliminated.

# 6.3 System Testing

In this testing, we take the requirements document as a reference to see whether the software meets all its requirements, goal or not.

The objective of the system testing is to promote software functionality, reliability and maintainability.

System testing is used where the software and the other system elements are tested as a whole. Its comes at the top level of the testing which is done after completion of different sub testing such as unit testing, integration testing, validation testing.

System testing is actually a series of different tests whose primary purpose is to fully exercise computer based system.

In these, we use sub testing which verify that system elements have been integrated and performed allocated function or not.

We used some testing in our system such as:

**Recovery Testing:**

In this system testing we verify that if any failure occurred in our system then its recover proper or not.

**Security Testing:**

In this testing, we attempt to verify that protection mechanism of our system. In our system we check the security such as login and password security. If any user want to access our deep information and want to use our services then he/she must have to be login and other user can’t access or see the information of the another user.

# 6.4 Black Box Testing

BLACK BOX TESTING, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access
* Behavior or performance errors
* Initialization and termination errors

**Techniques:**

Following are some techniques that can be used for designing black box tests.

* **Equivalence Partitioning** - It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.
* **Boundary Value Analysis** - It is a software test design technique that involves the determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.
* **Cause-Effect Graphing** - It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.

# 6.4 White Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.

It is one of two parts of the "Box Testing" approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.

**Techniques:**

Major White box testing technique is Code Coverage analysis. Code Coverage analysis eliminates gaps in a Test Case suite. It identifies areas of a program that are not exercised by a set of test cases. Once gaps are identified, you create test cases to verify untested parts of the code, thereby increasing the quality of the software product

There are automated tools available to perform Code coverage analysis. Below are a few coverage analysis techniques

* **Statement Coverage** - This technique requires every possible statement in the code to be tested at least once during the testing process of software engineering.
* **Branch Coverage** - This technique checks every possible path (if-else and other conditional loops) of a software application.

## **Chapter - 7**

Advantages and Disadvantages

* **Advantages:**
* Diabetic People Can Monitor Their Blood Sugar Easily
* Provides GUI/Interface Form Data Entered Daily
* Customizable Charts
* Doctor Appointment System
* Doctor Prescription System
* Medical Store Locator
* **Disadvantages:**
* Requires Internet for Usage
* Can Be Gimmick for Old Users
* Patient Still Need to Input Blood Sugar Level Manually

## **Chapter - 8**

Conclusion

# 8.1 Conclusion of Diabetes Monitor Application:-

The Application When Launched Could Change Life the People, This is Main Goal of the Application. the Application Will Be Beneficial to Both the Patient and Doctor. the Application Will Allow the User to Monitor Their Diabetes on Daily Basis and Will Guide Them to Live A Better Life.

* With Dietician Module, the Patients Will Be Able to Eat Healthy by Following Recipes and Tips of Dietician.
* The Friendly UI of the Application Providing A Detailed Information of Blood Sugar Level of Weeks, Months, Days, etc.

## **Chapter - 9**

Future Reference

# 9.1 Future Plans for Diabetes Monitoring Application:-

With the Change of Technology in World, People Want to Keep Up with Their Health In This Busy Lifestyle. the Overall Impression of Diabetes Monitoring Application is That This Prototype can Extended Almost to An Infinite.

* With Success of Diabetes Monitoring Application In Near Future, We Will Add A Yoga and Personal Fitness Module to Help People with Diabetes Live to Their Fullest.
* We also Hope to Pair Up with A Manufacturer and Launch A Fitness Band or Tracker, Along with Our Application to Provide A Seamless Experience.

Bibliography

# References: -

* <https://developer.android.com/>
* <http://php.net/>
* <http://www.wampserver.com/en/>
* <https://www.mysql.com/>