# DIABETES FIT

**BY**

**HEMADDRY SARKER RAJON**
**ID: 173-15-10425**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

**Fahad Faisal**
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised By

**Md. Sadekur Rahman**
Assistant Professor
Department of CSE
Daffodil International University

**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**DECEMBER 2021**

# APPROVAL

This Project titled "**Diabetes Fit**", submitted by HEMADDRY SARKER RAJON, ID No: 173-15-10425 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 06 January, 2022.

## BOARD OF EXAMINERS

**Chairman**

________________________

**Dr. Sheak Rashed Haider Noori**
**Associate Professor and Associate Head**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

**Internal Examiner**

____________________________

**Subhenur Latif (SL)**
**Assistant Professor**
Department of Computer Science and engineering
Faculty of Science & Information Technology
Daffodil International University

**Internal Examiner**

______________________________

**Md. Azizul Hakim (MAH)**
**Senior Lecturer**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

**External Examiner**

______________________________

**Dr. Shamim H Ripon**
**Professor**
Department of Computer Science and Engineering
East West University

# DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Fahad Faisal, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**

Faisal

**Fahad Faisal**
Assistant Professor
Department of CSE
Daffodil International University

**Co-Supervised by:**

Sadik

**Mr. Md. Sadekur Rahman**
Assistant Professor
Department of CSE
Daffodil International University

**Submitted by:**

Hemaddry Sarker Rajon

**Hemaddry Sarker Rajon**
ID: 173-15-10425
Department of CSE
Daffodil International University

# ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Fahad Faisal**, **Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Mobile application development*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan, Professor & Head,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

# ABSTRACT

High blood sugar which is also known as Diabetes is a very common disease now-a-days. The purpose of making is this android application is to help the diabetes affect people to maintain and track their blood sugar, medicine and food habit. This application is built in Flutter with some exciting features. Flutter is a free mobile application user interface. It helps to create applications for both iOS and android with only one code. As a result, developers have to write code once for both Android and iOS. This application will help people to keep fit as it has a build in feature of exercise suggestion and health blog about diabetes. It has features to record daily blood sugar level with specific time and date. Moreover, it has a field to display the statistics of blood sugar measurements. This application will help diabetes people to take medicine in time. As it has a notification system, which will remind the patient to take medicine in time. It also have a standard diet chat for the patient, health blood about diabetes to create more awareness among the users. This application is not only for diabetes patient, normal people can also use this application fluently for tracking their blood sugar and can take tips to keep fit and avoid high blood sugar or diabetes.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# Introduction

## 1.1 Introduction

High blood sugar interrupt the human body's ability to convert glucose into glycolysis. Human bodies use glucose to make glycolysis [8]. Human body releases an insulin (hormone) when glucose is in the blood. Insulin takes glucose into cells for used as glycolysis. Diabetes affect people’s body can’t produce enough insulin and huge glucose remains in the blood. Kinds of diabetes are – type 1, type 2 and gestational [6]. Among these three diabetes variant type 2 diabetes is more dangerous. Type 2 diabetes causes fatal diseases like Heart disease, Stroke and so on [7]. There are different kinds of applications are available in the market place for manage high blood sugar and help patient to maintain it. But, this application which is named as “Diabetes Fit” is not similar to other applications. It has some extra features from other ordinary diabetes management application which are available in the market. Like it has a standard diet chat features, health blog features, exercise features and so one. The prime motive to build this app is to help the diabetes patient to maintain their blood sugar and life-style in a dynamic way. No diabetes people can also use this application to keep fit and maintain their regular blood sugar.

## 1.2 Motivation

Diseases are now is a very common things in human’s day to day life. Diabetes is one of them. Today’s era is based on technologies. Developers are building self-caring android applications to help people maintain a healthy life style. “Diabetes Fit” is also a self-caring android application, which has some efficient features to help both diabetes and non-diabetes people to keep track of their blood sugar. Moreover, to build this application has also some reasons. There are numerous diabetes applications are available in play store. But, after a research as a result some drawbacks came out. Most of the application doesn’t contain some necessary Features like food suggestion, standard diet chat, exercise suggestions and so on. After finding out those limitations, we took a decision to build this application with those necessary features.

## 1.3 Objectives

Self-care is a factor of knowledge, individual and community motivation as a motivator. Diabetes self-care skills such as behavioral skills and blood sugar levels as a result.

- Self-tracking
- Self-conciseness
- Food habit
- Workout habit
- A good life

## 1.4 Expected Outcomes

Our awaited outcome from this application is to help people maintain their life with a routine bound way and a digital record of blood sugar in hand. From the health blog patient will be conscious about what to do and what not to do. Moreover, our plan is to change people's life with this small idea in a vast way.

## 1.5 Report Layout

- Background
- Requirement Specification
- Design Specification
- Implementation and Testing
- Impact on Society, Environment and Sustainability
- Conclusion and Future Scope

# CHAPTER 2

# Background

## 2.1 Preliminaries/Terminologies

Application elements are the most important building material of an Android application. Every section is an entry path from where the in-fracture or user can install application. Different chunk hang on others chunk. Application elements are four types [9]:

- Activities
- Services
- Broadcast receivers
- Content providers

Every chunk performs for a different reason and has a different life cycle which explains the process of creation and destroy of a part. Four types of application elements are describe below-

**Activities:**

Activity is known as user interface. Activity shows a single screen with a UI. As a demonstration, a music application may have one function that displays a list of new music, another function for searching songs and another function for showing genera. Moreover, those functions can work along to create a promising user experience in the music application. Every function is independent for other. Although, an independent task can perform if another task permit other task to perform. As demonstration, the camera application can perform work on a music application that includes new images and permit the user to share an image. Specific function can provides critical communications between the system and the application:

- Record user's present concertation about to allow the system is always using a process that handles the task.
- Tracking the emendate task items that the actor may fall back (default functions), and keep attention those processes.
- Help the application to retrieve its execution work flow so that the user can return back to the functions when his or her previous state has been restored.

- Provides a path for applications to use actors flow among different things and for the infracture to integrate this flow.

**Services:**

The service is a general purpose login to keep the app running for all sorts of causes. It’s the part which works in the back-end to achieve term tasks or to achieve remote task flows. The service doesn’t provide a UI. For demonstration, a service may perform tasks in the background when the user is in a different application, or it could download material over the network without interfering with user screen with the function. The other party, such as an employee, may start a service and allow it to operate or be bound to communicate with it. There are types of providers, which instruct the hardware how to run an application [10]. Those are started assistance and bound assistance.

**Started assistance** instruct the infrastructures hold them performing until the task is finished. This can be concur certain material in the back end task or to play task even after the user has left the application backing up material or playing task also represents different types of activated utility that change the way the infrastructures behaves:

- Music player is something the actor knows directly, so the application order the infrastructures that it needs to be ahead with the notification to aware the actor about it; in this situation the infrastructures knows that it has to keep the process of that utility running, because the actors will not be happy when it crashed.
- Normal background assistance is not something the actor knows exactly as active, so the infrastructure has more access to manage its own work flow. It may permit it to be executed if it needs the random access memory of the user's immediate concern.

**Bound assistance** use it because another application (or application) said it wanted to use the app. This is a service that provides an application programming interface to another work flow. The infrastructure therefore recognizes that there is a dependence between these work flows, so if process X is tied to system Y utility, it sense that it needs to keep perform Y running on X. In addition, if permit X is specific. The actor cares about it, and knows how to treat permit Y as fact he also cares about. Due to their adjustability, utility have become a very essential building elements for all kinds of high-level infrastructure ideas. Always on display, android listeners, usability resources, and many other key options of the system are all built into the applications

That the applications use and the infrastructure that binds them where they need to work.

**Broadcast receivers:**

The streaming catch is the elements that permit the infrastructure to deliver events to the application without the natural actor flow, permitting the application to give feedback to broadcast message across the system. Because broadcast catchers are one of the well-defined inputs to the application, the structure can bring transmit even to applications that are not presently in use. So, for demonstration, an application can sets an alarm to send a message to the actor about an upcoming event and by bringing that alarm to the application's streaming receiver, there is no necessity for the application to continue performing until the alarm goes off turned off. Most streaming comes from a program for demonstration, a transmit announcing that the screen has been turned off, the charge is down, or a photo has been taken. Apps may re-start the broadcast for example, notifying other applications that certain material has been downloaded to the device and available for use. Although transmit receivers do not display the actor interface, they can create a status bar notification to notify the actor in the operation of a broadcast event. Most commonly, however, the streaming catcher is simply a path to other elements and is consider to perform a very small scale of task.

**Content supplier:**

The content supplier has a split app set of material that can stock on a document structure, on a website, on the web, or on any other storage path that application have entry permission. With the content supplier, some applications may query or make change the data if the content supplier approves it. For example, the Android structure provides a content supplier that manages actor contact info. Any application with the appropriate permissions can ask a content supplier to read and write information about a particular actor. It is attractive think of a content supplier as saying something on a website, cause there are too many APIs and support built into them in that usual case. They have a different primary motive in the design of the structure. In the system, the content supplier is the access point for the application to publish named data objects, identified by the URI system. That way the application can determine how it need to plot the material it contains in the material space, giving those materials to other organizations that they can use to permit the entry to get data. Some facts permits the infrastructure to do in operating the application:

- Giving the material does not need the application to stay active, so the material can continue after their applications are out. The structure's only necessity is to verify that the host application is still active when it has to restore application data to the corresponding information identifier.

- Information material are also serves an important fine-grained security model. For demonstration, an application may set the image identifier, but leave its content supplier closed so other applications can access it freely. If a second application attempts to access that information identifier on the browser, the application may permit that application to access the data with a temporary identifier to allow access to that data only after that identifier, but no access in the second application.

Content supplier are also helpful in reading and writing private information in app and shared. An isolated feature structure design so that any application can start as part of alternative application. For demonstration, if I want the actor to take a picture with the mobile's camera, it is possible that there is another app that does that and my application can use it instead of doing the photography task myself. We do not need to enter or connect a link from the application. Rather, I can start work on the application that takes pictures. After finishing, the image is restored to my application for use. To the actor, it looks like that is actually part of application. When the hardware starts a part, it begins the operation of that application if it is not earlier in use and specifies the required classes in part. For better understanding, when actor’s application begins work on a camera app that takes a picture that function continues through the process that is part of the camera application, not the process of your app. Moreover, not similar applications from many other infrastructure, apps do not have an individual login. Because the structure uses every single application with a discrete process that has file access that limit access to any other applications, one application is unable to use the component directly from other application. However, the Android operating system can. To unlock an element in other application, bring an info to the infrastructure that specifies actor’s intention to begin a segment.

## 2.2 Related Works

There are different kinds of diabetes management application are available in the google play store. Which do also most the similar work, like record blood sugar, statistics, and medicine record and so on. Some application's interface are shown below-

**Diabetes Tracker:**

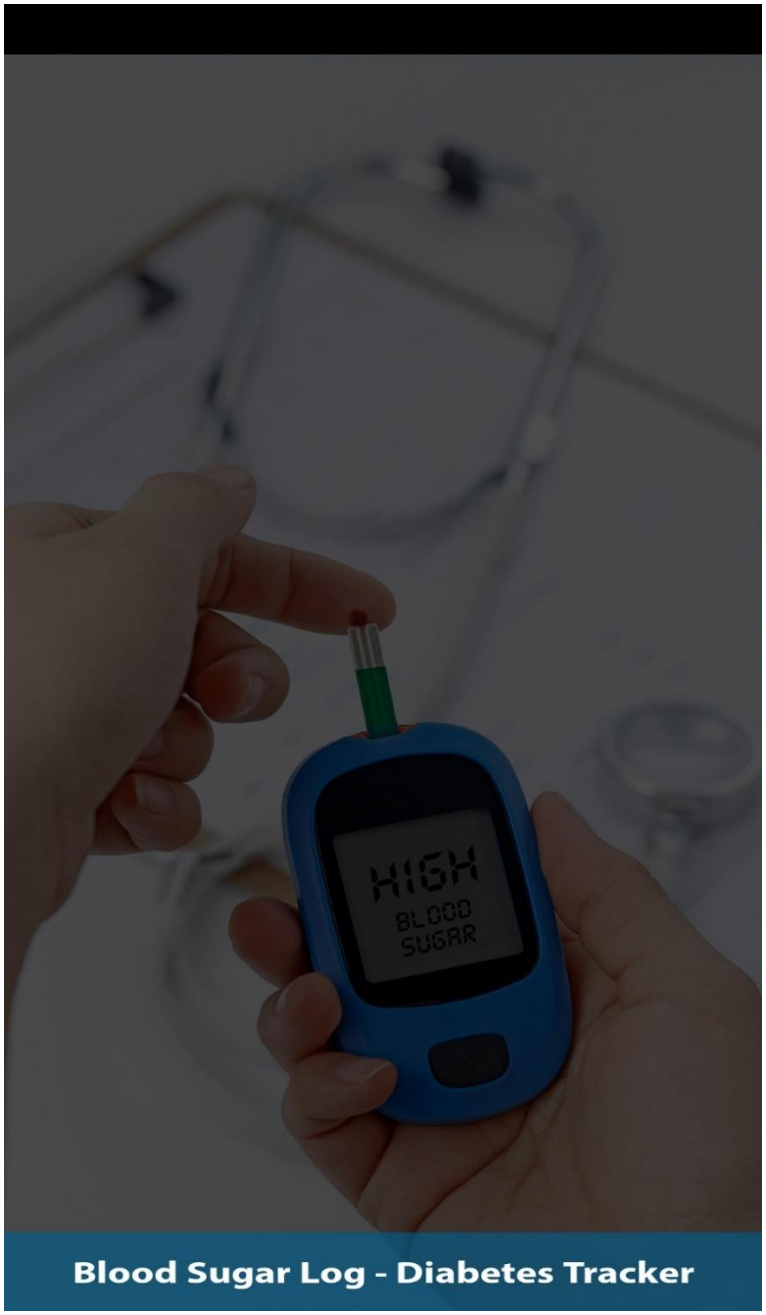
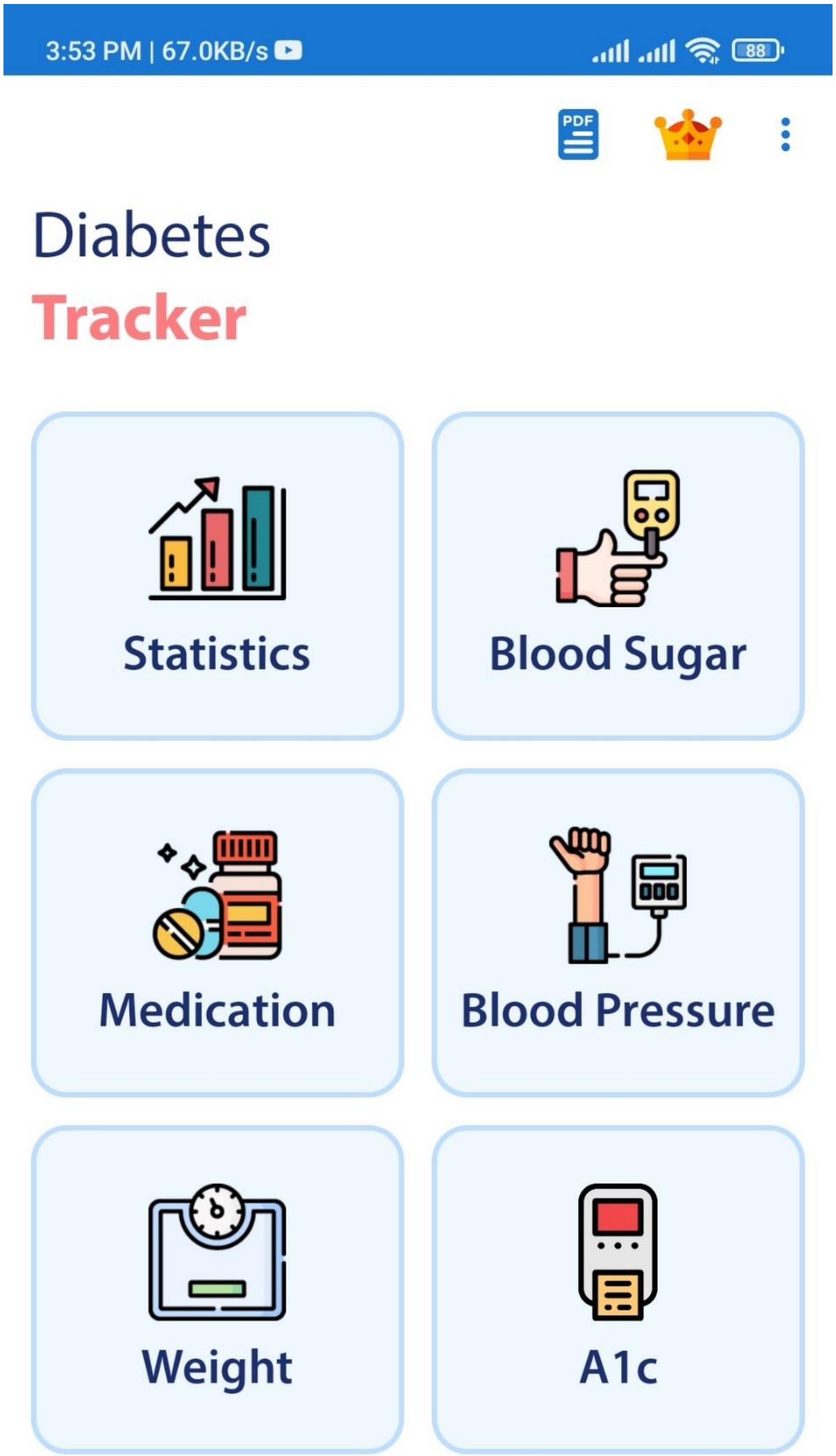


Figure 1: Diabetes Tracker application's interface no. 1 [1]

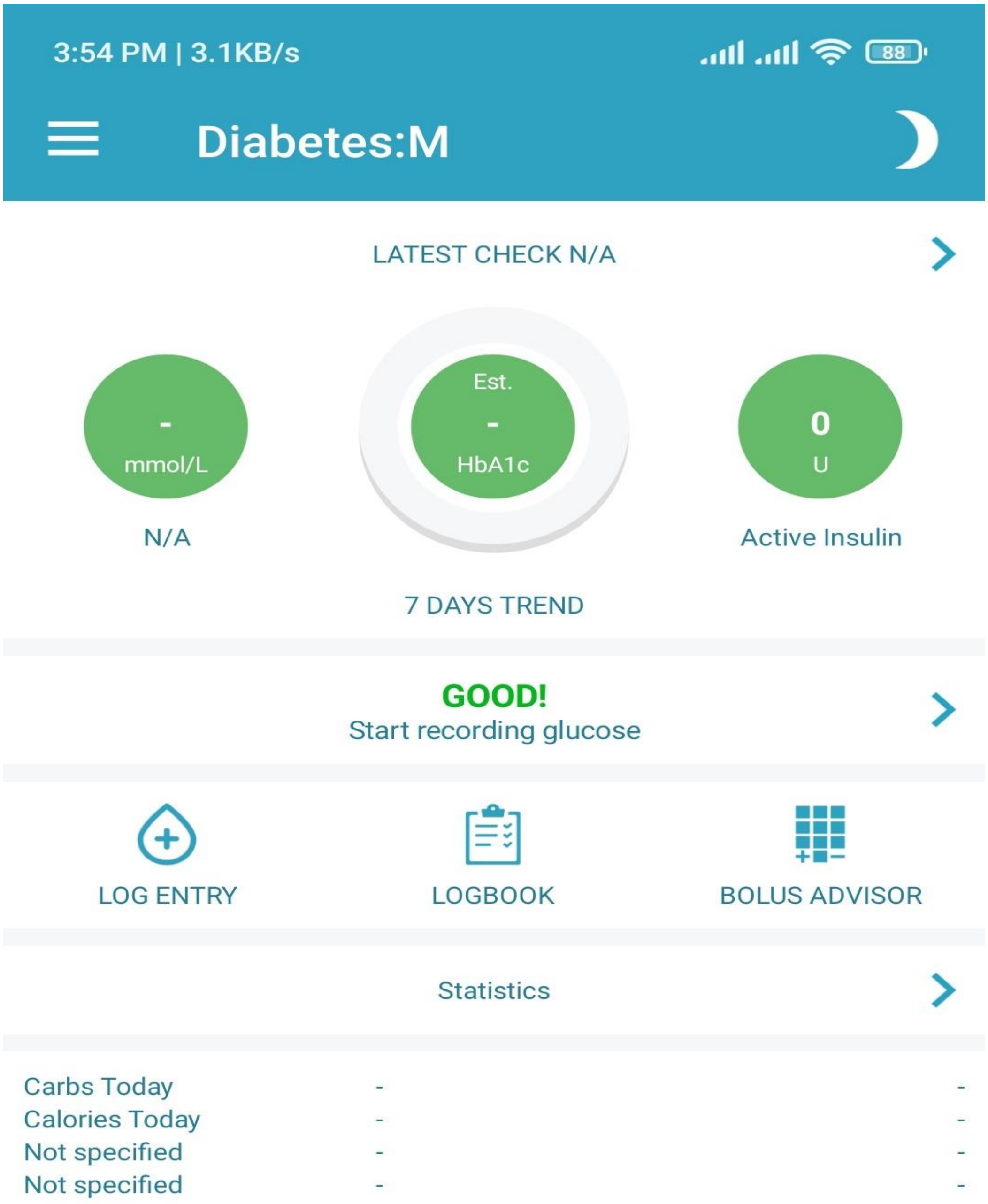

3:53 PM | 67.0KB/s

Diabetes Tracker

Figure 2: Diabetes Tracker application's interface no. 2 [1]

**Diabetes: Manager-**

Figure 3: Diabetes Manager application’s interface no. 1 [2]



GOOD!
Start recording glucose

| | | |
|---|---|---|
| Carbs Today | - | - |
| Calories Today | - | - |
| Not specified | - | - |
| Not specified | - | - |

Figure 4: Diabetes Manager application's interface no. 2 [2]

**Diabetes Diary:**

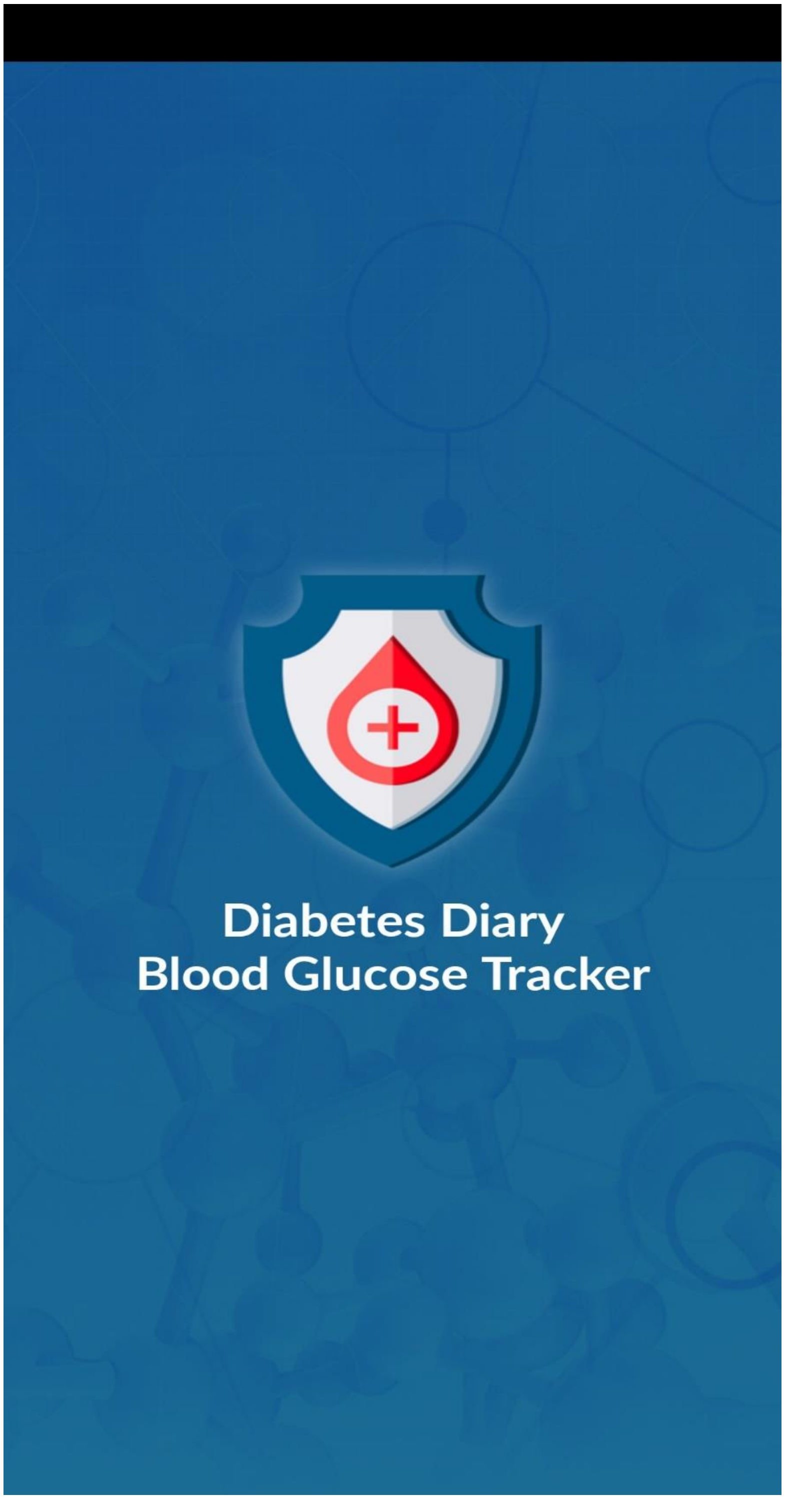


Figure 5: Diabetes Diary application’s interface no. 1 [3]

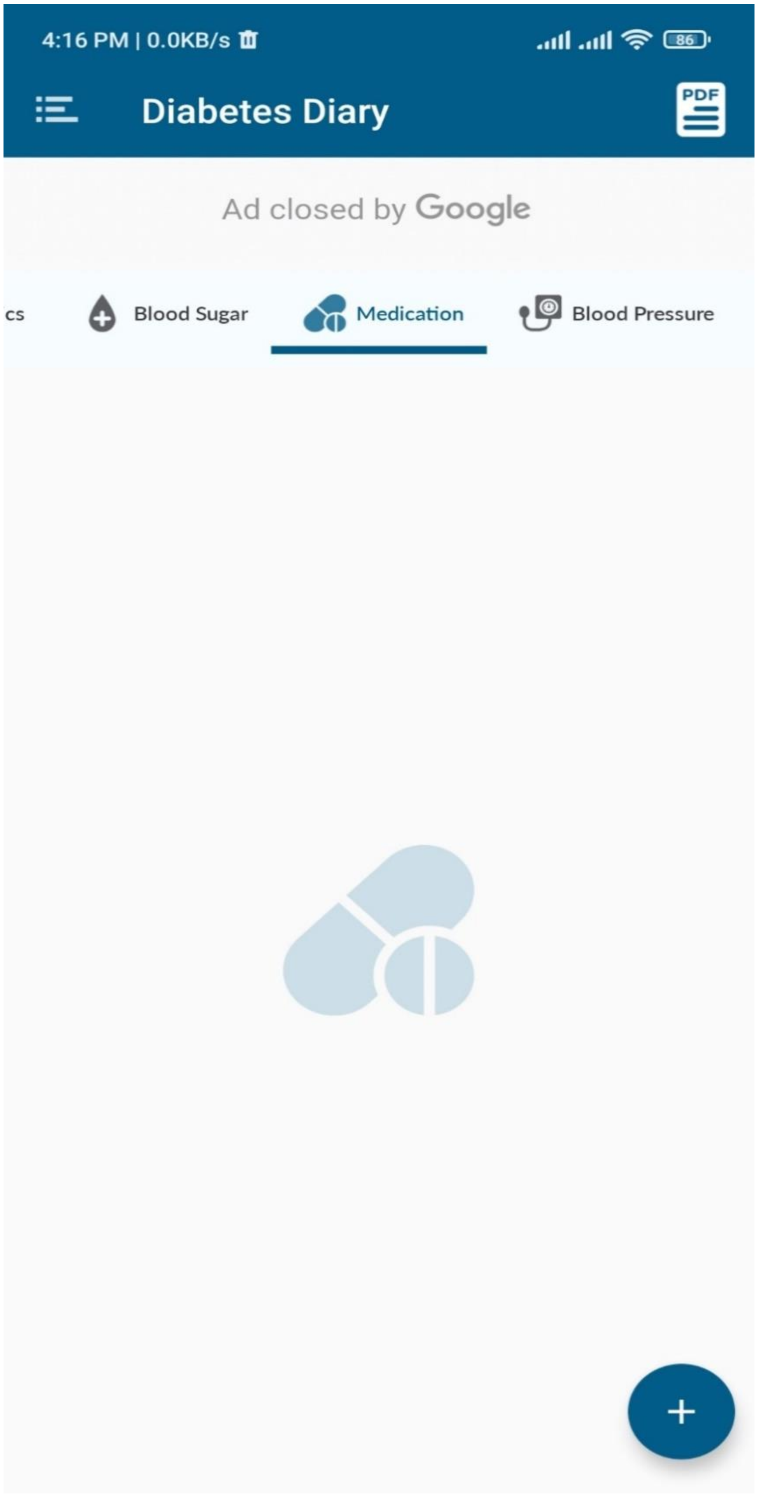


Figure 6: Diabetes Diary application's interface no. 2 [3]

### 2.3 Comparative Analysis

Firstly, we have to know about the difference between mobile application and websites. Then we will discuss about difference between types of mobile applications. So, now let's talk about mobile apps and websites. There has been a long debate about what mobile applications are. Whether software or something else. However, a mobile application is a computer software or computer program used for a specific purpose and works on a variety of mobile devices such as a smartphone, tablet, or smart watch, which ends up running on the Operating Platform for Android, iOS or windows. On the other hand, a business website is a digital address database that provides complete information about what the business is about, the products and services they have been offering, and the business performance. It also proves to be an excellent platform for showcasing all business offers and allowing them to present the business product and services they have been providing in a better way. In addition to this, the business website serves as a platform between business and customers and that is why there is always a need for web development services. Now, talk about application variant [11]:

- Native Applications
- Web Applications
- Hybrid Applications
- Interpreted Applications
- Cross-platform Applications

**Native applications:**

Indigenous applications were made to be processed locally, presumably device type and OS and its narration. Code included to get usable, likely to the operation used for a normal computer applications. When the application is appropriate for diffusion, it is released to the application keep track of every platform. Then comes the process to check the specifications whether it meets or not with the phase in which we will operate. After that, the application is ready for download. One of the key features of the traditional app is, it permits limitless interact with all options provided by the machine. Also, Internet access is not necessary to use this kinds of application. This kinds of app are fast to response and can be driven in the background operation, too they shows a warning if there is an operation that requires actor permission. Native method is costly to develop.

### Web applications:

These applications are developed for using in browser. Web applications are developed using standard technologies. No platform need to install to use it and no interrupted permission is needed. Just need to access to the Internet. In short, fast and user friendly to use. Web applications are fully independent of the website. Web browser is need to run web applications. There is no requirements for operation system weather it is mac or windows.

### Hybrid Applications:

Integrated applications use web strategies, but are not operated by a browser. Instead, they are used for the device's web browser which provides permission to device-specific features via the application programming interface. Integrated apps offer great benefits, as example re-use of code for multiple environment and have permission on hardware. Mixed programs have disadvantages collate to native applications. Actor’s information has a problem with not using native components on social media. The second one is applications could be slow due to the additional traffic associated with the web handler.

### Interpreted Applications:

Translated apps are made from a one project that is widely changed as specific code, and another is interpreted during operation. Their implementation is not based on the environment and uses technology and a few languages. Unlike web development and hybrid multi-platform solutions, and Translated applications closer to specific interfaces are available. We can define it as a disadvantage.

### Cross-platform Applications:

This types of applications are made for both iOS and android. That means one codebase is used for both platform to develop applications. Flutter is the platform to make those kind of applications. Flutter is based on dart language. It’s a platform where applications are built in one codebase for two different kinds of operating system. Among those types of mobile applications cross-platform Applications are more convenient to build. Because, it gives us the privilege to write code one time and use it in both iOS and android operating system.

## 2.4 Challenges

To achieve something innovative challenges must have to accept. Challenges are like benchmark to make something more appropriate. In most cases, five challenges arrives to build a mobile application [12]. Those are-

- Development Approach
- Attention
- Device Screen Size and Compatibility
- Performance
- Customer Reviews and experience

So, in time of build a mobile application these points have to keep in mind for better outcome. These are the key note challenges behind building a mobile application.

# CHAPTER 3
# Requirement Specification

## 3.1 Business Process Modeling

BPM is known as graphic visual of a business process and sub related processes [13]. Process modeling produces broad, multidisciplinary diagrams and flow charts that contain important details in the performance of a given work flow along with the following:

- Events and activities take place within the course of the work
- Events and activities owner
- Decision and action can be taken based on results
- Total process times and each step in the work flow
- Success and failure levels of the work flow

**Key notes of BPM:**

- Process models are not designed in person. Instead, it generated by mining algorithms with event record to build workflow.
- Process models are based on quantitative information material. By creating a model for its process of creating a new account, a software company may find that a large number of customers leave the registration process because it's time consuming.
- Process models are usually provided using two standard business process notification styles: Business Process Modeling and Notation or Integrated Modeling Language. Within these commentary systems, certain visual elements have meanings that are universally recognized when applied to a process model. UML or BPMN are standard writing methods permits operational models to be easily shared:
    1. The bolt represent the successive flow
    2. The obovate represent the starlings and endings of work processes
    3. Rectangles represent certain functions within the process
    4. Swimming methods are used to find which parts of the procedure
- BPM should not be puzzled with process maps or diagram. Process maps are based on staff records are created in person and gives a high-level view of workflow. Process models are deep material driven that generate more accurate ideas for workflow.

**How BPM are made:**

One must first understand the business process modeling tools, tricks and requirements which is suitable for event logs and mining. Most IT systems maintain logs. These digital logs automatically and track status to changes in the infrastructure. Error that occurs with the infrastructure can be an operation changing. General occurrences are:

- Actor logged in
- Updates the record
- Actor submits form
- Information shifting between system structure

Event logs track both event events and the information surrounding these events, such as the device performing the task and time to perform tasks. Event records used as inputs during the production of process models. Mining is the use across all event log data. The algorithm recognize trends in material and use resultative reports for analysis to produce a clear visual of the process work flow within the infrastructure. This visual presentation is a working model of process. Depending on the working process model, mining algorithms can be used to an isolated infrastructure structure, multiple infrastructure structure or the whole technology, ecosystem and sections.

## 3.2 Requirement Collection and Analysis

Analysis and data collection are main challenges behind building an application. There are many ways to find needs for users, customers, sponsors and other people who are interested in development. One of the following methods or usually a combination, is used:

- Mocking
- Literature Review
- Interface Analysis
- User monitoring
- Prototyping
- Needs workshop
- Reverse Engineering
- Research

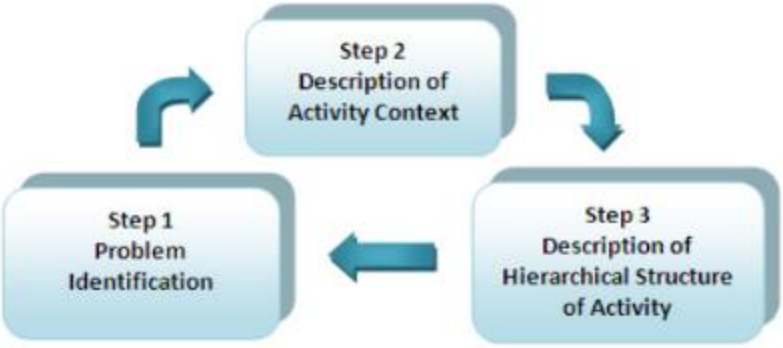


Figure 7: The Requirements Analysis Approach [5]

## 3.3 Use Case Modeling and Description

The usage working model shows interaction between actor and infrastructure. Thus, it defines user objectives, interaction between users and the infrastructure, and the infrastructure's necessary conduct in fulfilling principles. This model contains a number of items. Must needed elements are: usage case, players and linkage between them. The diagram of the case used is used to clearly show the bottom set of the model to make communication easier. There will usually be drawings of a few characters used in a particular model, subset of model elements suitable for a specific reason. Part of the identical model may be shown in most usage diagrams, but each model should fit. When the contraption are being used to store the model, this sync principle is automatic so that any twist in the demo element will automatically appear in every used case diagram showing that element [14]. The application case demo could hold packages used for model editing to facilitate analysis, communication, navigation, development, editing and editing. Most of the case models used are actually text, with the text noted in the Application Case Specification affiliated with every single aspect of the usage model. These factors explain the flow of cases of use case. The utility case model serves as an integral thread throughout the system development. It is used as a key determinant of system performance requirements, as a basis for analysis and design, inputs to duplicate planning, as a basis for defining test cases and as a basis for user notation.

**Basic model elements:**

**Actor:** Part of the model presenting every single character. Characteristics include player name and brief description.

**Use the Case:** Part of the model that represents each use. Characteristics include the use case identity and the requirements of the use case.

**Associations:** Associations build linkage between actors and the use cases in case they take part in.

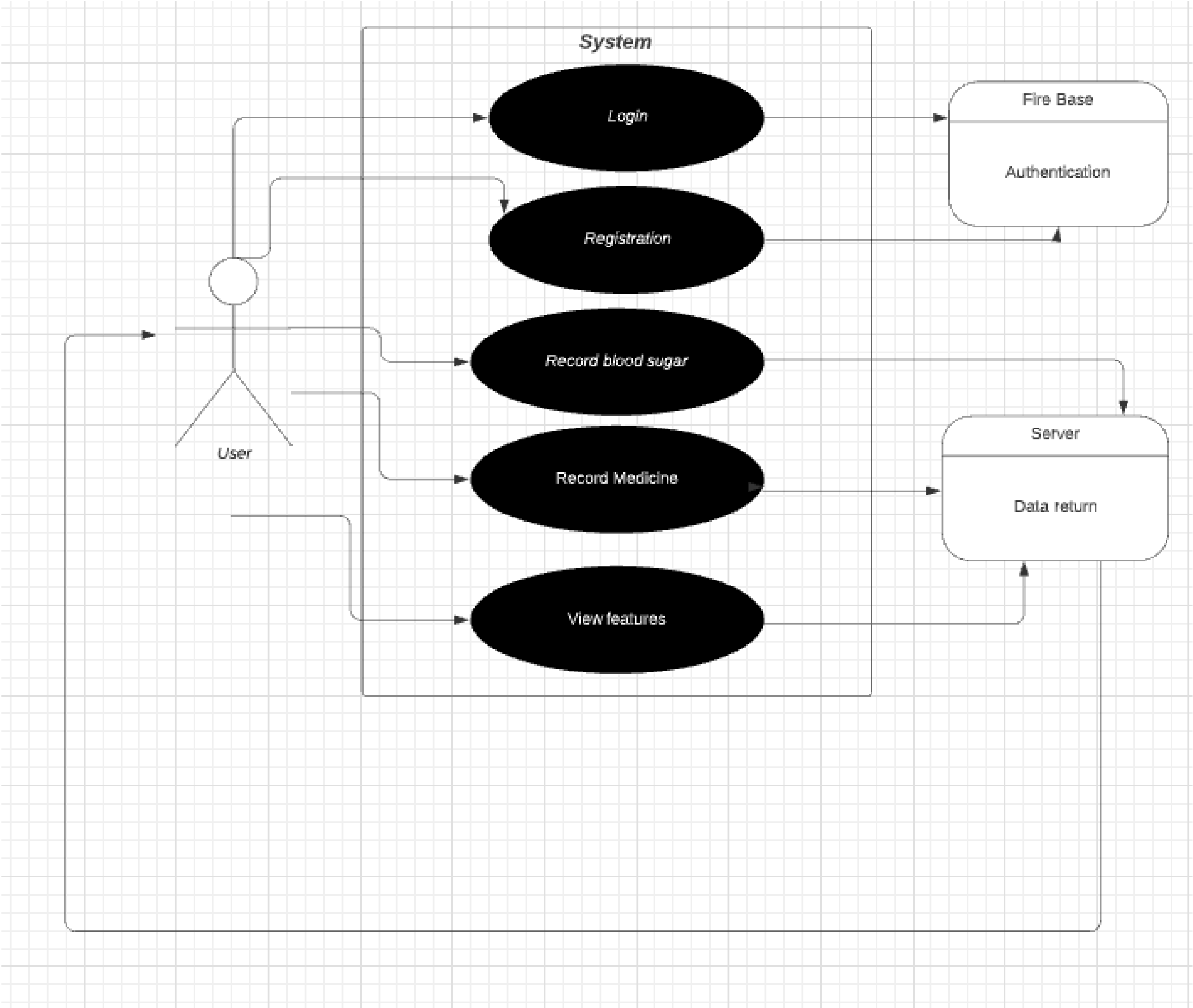


Figure 8: Use case model for Diabetes Fit

This use case model express here the working process of the application. The login and registration authentication will be checked by the admin. User can easily view the others features and record blood sugar and medicine.

# CHAPTER 4

## Design Specification

### 4.1 Front-end Design

Front End is everything a user sees in a mobile app including app design. Simply put, the front end of the mobile app is what the user experiences. You may be familiar with the term “User Experience” or “UX”; the conclusion is where the user encounters and shares these things. The advanced engineer will focus on what is happening with the app, rather than on the background, which is often described as the background [15]. The process of developing a mobile application involves a collaboration between front and back developers to complete as a complete mobile application. There are usually two types of pre-development: Indigenous and hybrid. The original native development is usually designed for iOS or Android platforms, while hybrids use both and can be compatible with all types of applications.

### 4.2 Back-end Design

Background Upgrade to Mobile Application means upgrades that occur on the server side. Backend is an essential part of a mobile app that stores, protects, and processes data. The background app is like a server for users to organize the required information. Precisely, background apps are used by previous applications to send a piece of information [16]. An advanced application sends information to the backend online using protocols. These protocols are designed specifically for applications only. Codes written by backend developers help generate website Information in a previous application. So, all the required output that we need is generated by the background app. The backend developer focuses on the following features of the application:

- Database
- Scripting
- The architecture

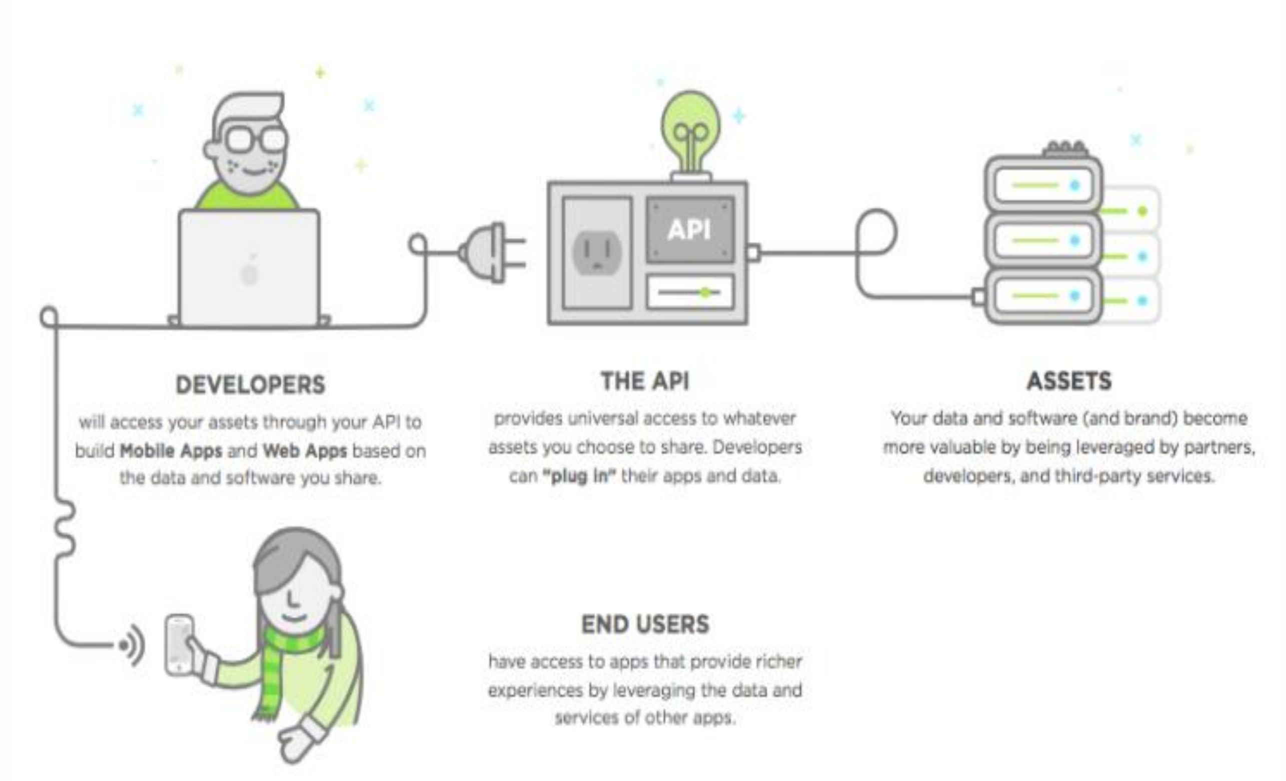

API
DEVELOPERS
will access your assets through your API to build Mobile Apps and Web Apps based on the data and software you share.
THE API
provides universal access to whatever assets you choose to share. Developers can "plug in" their apps and data.
ASSETS
Your data and software (and brand) become more valuable by being leveraged by partners, developers, and third-party services.
END USERS
have access to apps that provide richer experiences by leveraging the data and services of other apps.

Figure 9: Back-end work process [4]

## 4.3 Interaction Design and User Experience (UX)

User interface (UI) is a process that designers use to create links to software or using devices, focusing on aesthetic or style. The developer’s aim is to create links that user’s find easy to use and enjoy [17]. UI design refers to the visual interface of the user and other forms such as voice-controlled areas. The user interface is the connecting place where users interact with the projects. UI formats are [18]:

**Graphical user interfaces (GUI):** Users link with visual presentations on digital control panels. Desktop’s visual is GUI.

**Voice-controlled interfaces (VUI):** Users deal with this in their own words. Many assistants like Siri, Alexa and android assistant are good example of VUI.

**Gesture-based interfaces:** Users here occupy with 3D design spaces through physical activity in virtual reality (VR).

In time of making user interface (UI), we have to keep some key notes on our mind. Those are:

- Usability and likeability
- Enjoyable UI
- UI should communicate brand values.

These key points are very important in perspective of user experience in using applications. Easy and interactive UIs attracts the user most.

# CHAPTER 5

# Implementation

## 5.1 Implementation of User interface

The first impression of an application is user interface. Easy and more user friendly user interfaces attracts the user most. So, we tried to build a clean UI with modern design and material design pattern with fire base authentication. Our prime goal is to satisfy the user with a simple and minimalistic design pattern. So, as per our plan we implemented the user interfaces. We use abstract model to implement the application. Now we will see some preview of the user interface:

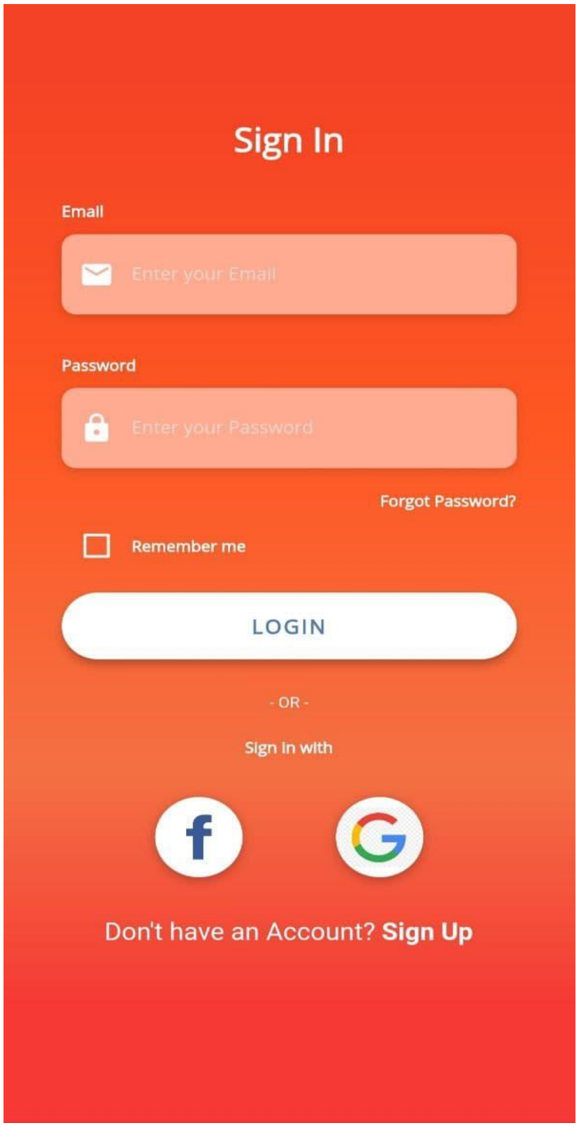


Figure 10: Sign in page of Diabetes Fit

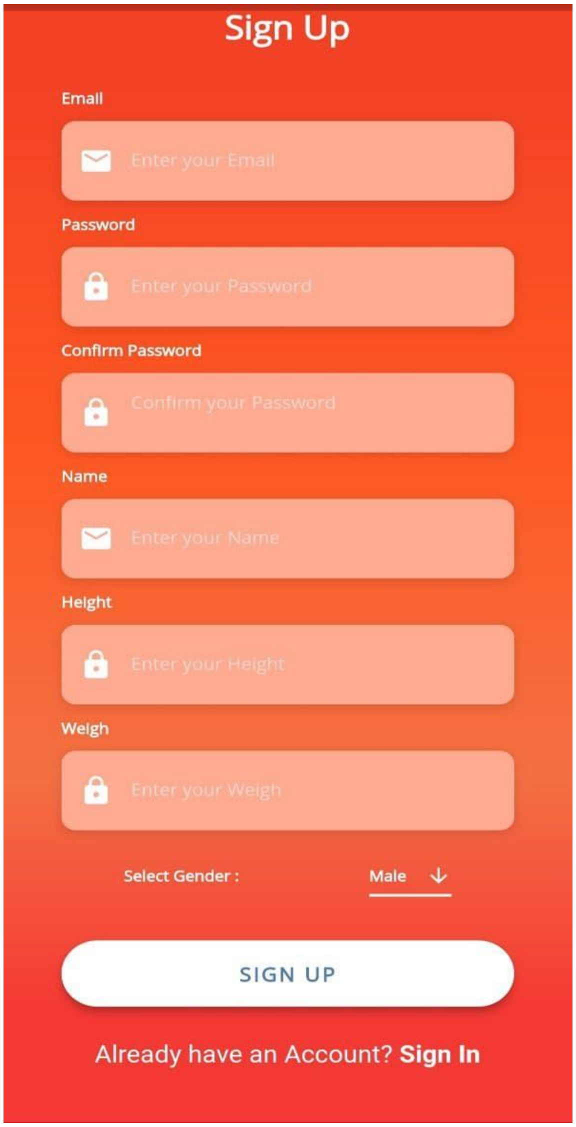


Figure 11: Sign up page of Diabetes Fit

So, here in the preview we can see the interface of sign in and sign up or registration view. Now we will see the preview of inner interface and some preview of user features-

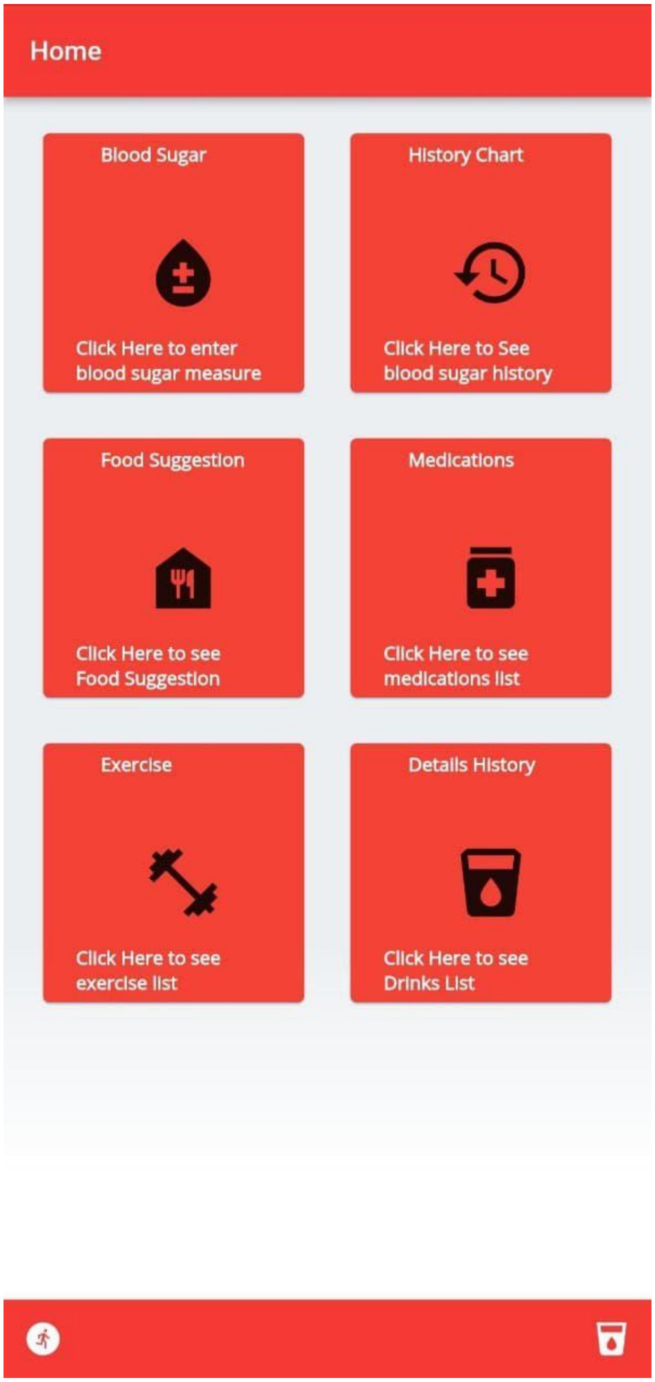

Home
Blood Sugar
Click Here to enter blood sugar measure
History Chart
Click Here to See blood sugar history
Food Suggestion
Click Here to see Food Suggestion
Medications
Click Here to see medications list
Exercise
Click Here to see exercise list
Details History
Click Here to see Drinks List

Figure 12: Inner interface of Diabetes Fit

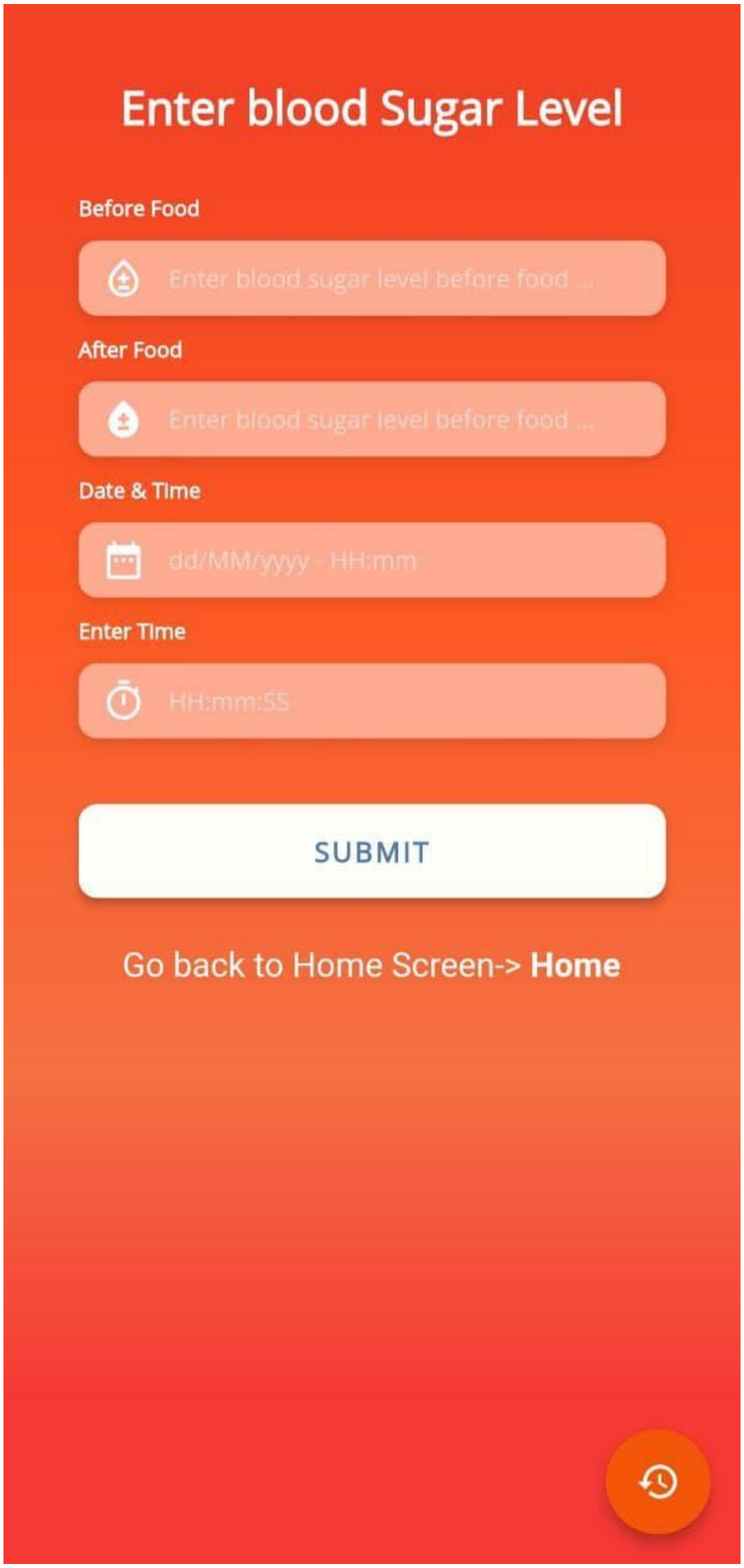
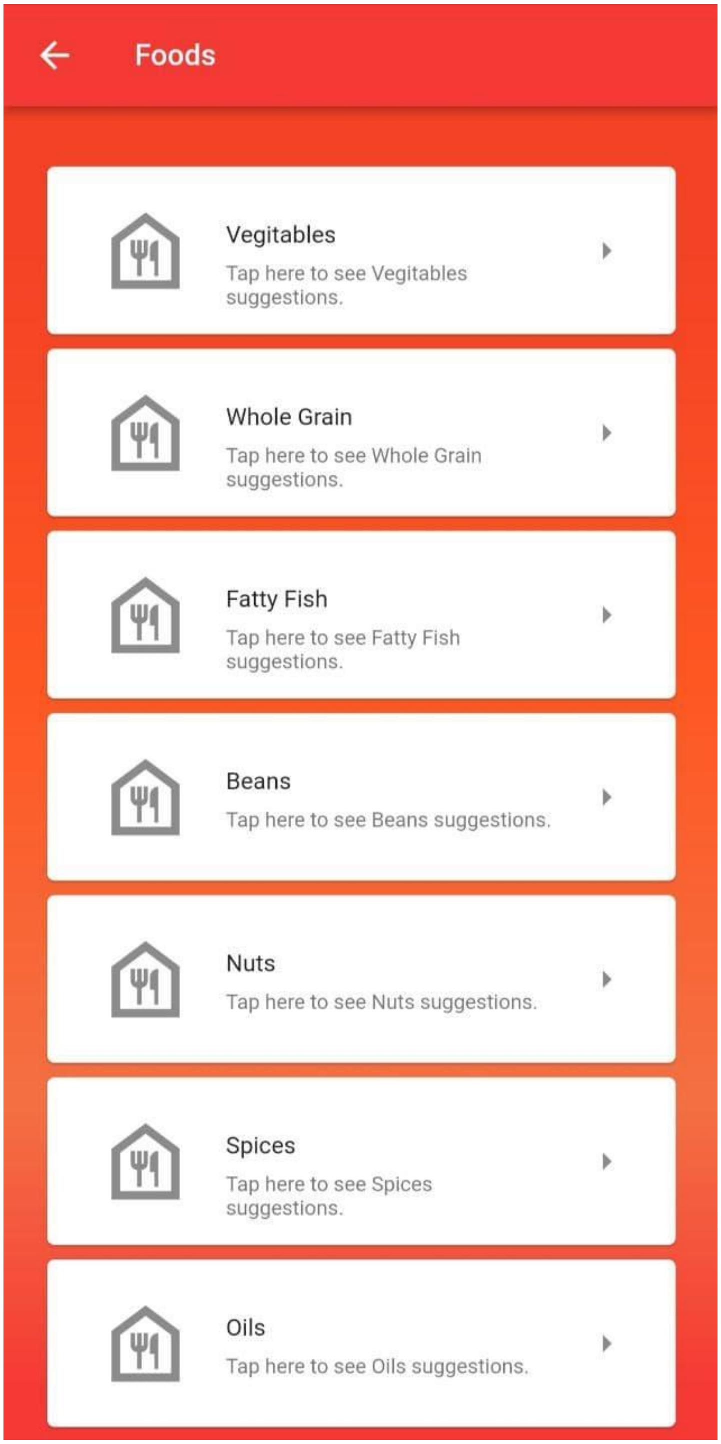

Enter blood Sugar Level
Before Food
Enter blood sugar level before food ...
After Food
Enter blood sugar level before food ...
Date & Time
dd/MM/yyyy - HH:mm
Enter Time
HH:mm:SS
SUBMIT
Go back to Home Screen-> **Home**

Figure 13: Interface of blood sugar record



Foods

Vegitables

Tap here to see Vegitables suggestions.

Whole Grain

Tap here to see Whole Grain suggestions.

Fatty Fish

Tap here to see Fatty Fish suggestions.

Beans

Tap here to see Beans suggestions.

Nuts

Tap here to see Nuts suggestions.

Spices

Tap here to see Spices suggestions.

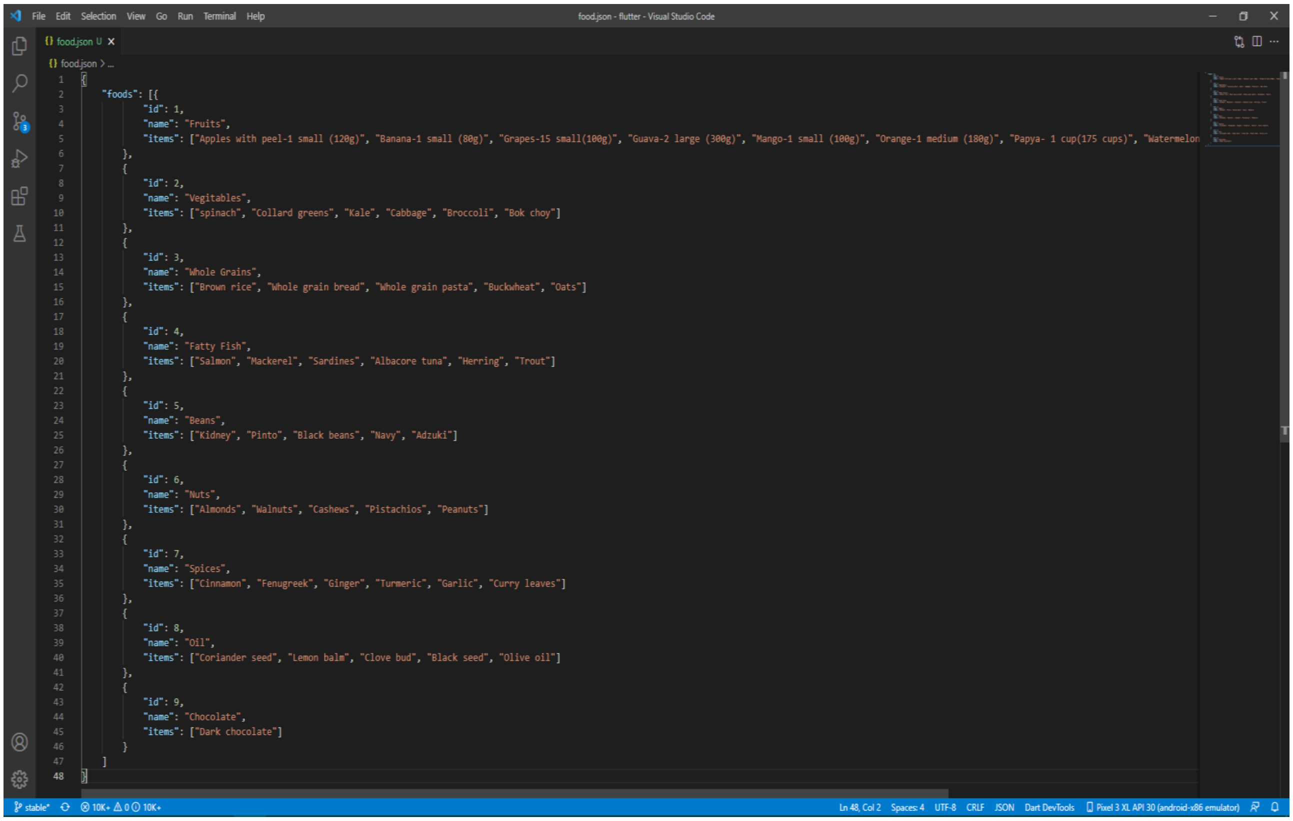
Oils

Tap here to see Oils suggestions.

Figure 14: Interface of food suggestion

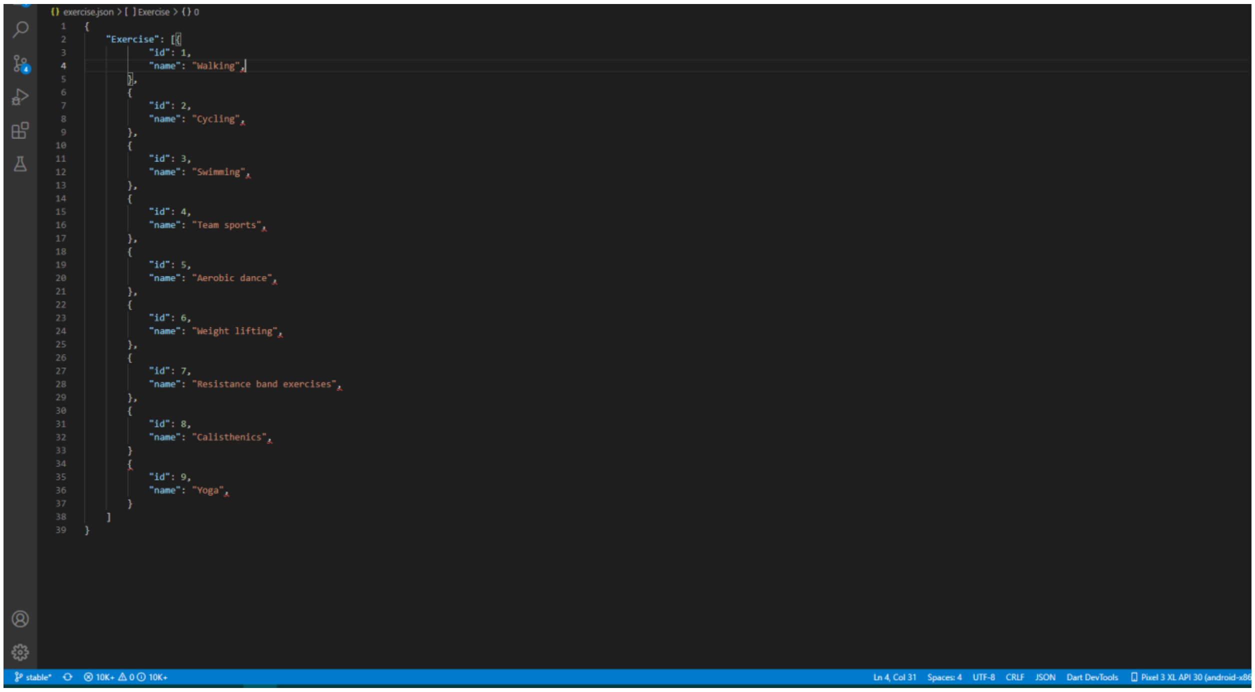
## 5.2 Implementation of JSON list

JSON is stands for java script object notation. Here, in to key features we use JSON list for inputting the value of the field food suggestion and exercise suggestion. For a clear vison we are adding the previews of JSON list here-


File Edit Selection View Go Run Terminal Help
food.json - flutter - Visual Studio Code
food.json U
food.json > ...
stable*
10K+ 0 10K+
Ln 48, Col 2 Spaces: 4 UTF-8 CRLF JSON Dart DevTools Pixel 3 XL API 30 (android-x86 emulator)

```json
{
    "foods": [{
            "id": 1,
            "name": "Fruits",
            "items": ["Apples with peel-1 small (120g)", "Banana-1 small (80g)", "Grapes-15 small(100g)", "Guava-2 large (300g)", "Mango-1 small (100g)", "Orange-1 medium (180g)", "Papya- 1 cup(175 cups)", "Watermelon
        },
        {
            "id": 2,
            "name": "Vegitables",
            "items": ["spinach", "Collard greens", "Kale", "Cabbage", "Broccoli", "Bok choy"]
        },
        {
            "id": 3,
            "name": "Whole Grains",
            "items": ["Brown rice", "Whole grain bread", "Whole grain pasta", "Buckwheat", "Oats"]
        },
        {
            "id": 4,
            "name": "Fatty Fish",
            "items": ["Salmon", "Mackerel", "Sardines", "Albacore tuna", "Herring", "Trout"]
        },
        {
            "id": 5,
            "name": "Beans",
            "items": ["Kidney", "Pinto", "Black beans", "Navy", "Adzuki"]
        },
        {
            "id": 6,
            "name": "Nuts",
            "items": ["Almonds", "Walnuts", "Cashews", "Pistachios", "Peanuts"]
        },
        {
            "id": 7,
            "name": "Spices",
            "items": ["Cinnamon", "Fenugreek", "Ginger", "Turmeric", "Garlic", "Curry leaves"]
        },
        {
            "id": 8,
            "name": "Oil",
            "items": ["Coriander seed", "Lemon balm", "Clove bud", "Black seed", "Olive oil"]
        },
        {
            "id": 9,
            "name": "Chocolate",
            "items": ["Dark chocolate"]
        }
    ]
}
```

Figure 15: Food suggestion JSON list


{} exercise.json > [ ] Exercise > {} 0
stable* 10K+ 0 10K+
Ln 4, Col 31 Spaces: 4 UTF-8 CRLF JSON Dart DevTools Pixel 3 XL API 30 (android-x86

```json
{
    "Exercise": [{
            "id": 1,
            "name": "Walking",
        },
        {
            "id": 2,
            "name": "Cycling",
        },
        {
            "id": 3,
            "name": "Swimming",
        },
        {
            "id": 4,
            "name": "Team sports",
        },
        {
            "id": 5,
            "name": "Aerobic dance",
        },
        {
            "id": 6,
            "name": "Weight lifting",
        },
        {
            "id": 7,
            "name": "Resistance band exercises",
        },
        {
            "id": 8,
            "name": "Calisthenics",
        }
        {
            "id": 9,
            "name": "Yoga",
        }
    ]
}
```

Figure 16: Exercise suggestion JSON list

## 5.3 Implementation of Database

We use Firebase for data record and authentication. Firebase is a Service, which works in the backend. Give privilege to the developer’s huge tools and services to help them in developing better applications, increase their user base. Firebase is classified database system, which stores data in notation such as JavaScript object notation. Key features are:

**Authentication:** This database Supports authentication in various way. Like using passwords, phone numbers, Google, Facebook, Twitter, and more.

**Real time database:** Data is concurred to all clients in real time stamp and is always available even when the app is offline.

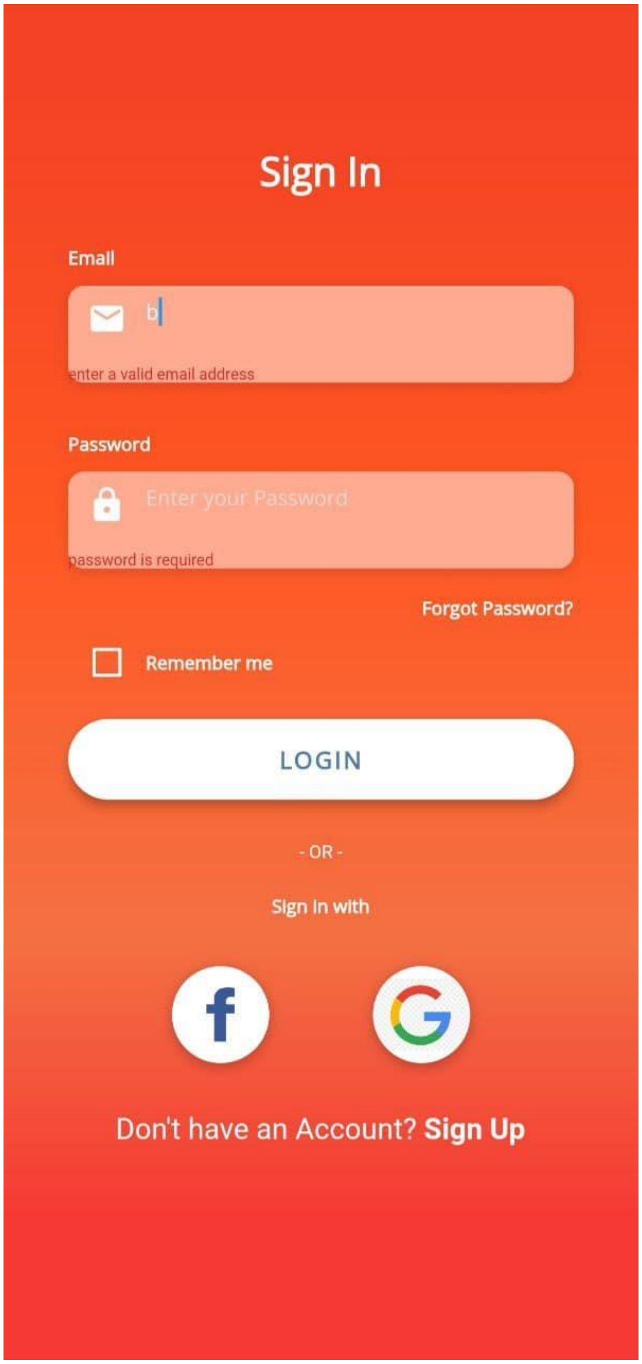
**Hosting:** It provides faster web application hosting; Content is stored in cached memory and delivery by networks around the world.

**Test lab:** Applications could be test virtually and virtual devices can found in data centers.

**Notifications:** Without coding notification can be send.

This database system gives us the privilege to both record data and data authentication on a single platform.

Now, we will show some preview of sign up and sign in page, where it will check valid information's like email, password and so on. Because, in the back end fire base is used for authentication. If we input invalid email or password it will return a warring in red line. So, here is the preview-



Email: b

enter a valid email address

password is required

Figure 17: Check validation

# CHAPTER 6

# Impact on Society & Sustainability

## 6.1 Impact on Society

In this technological era, people are more likely to use computer programs. But now the rage is changing as mobile users become more numerous compared to computer users. With the number of mobile users increasing in number, the market came up with a mobile application for mobile users. It is reported that the use of mobile applications and upgrades is growing rapidly. So there is a positive consequences of mobile apps around the world on users. The use of mobile applications in the developed world has become a high priority for the people, thus the community of developing countries is updating and using a new type of technological infrastructure, to access mobile application services. These mobile apps work on a hand-held mobile phone, which is easy to use and can be found everywhere. Nowadays, many people use mobile apps to communicate with their friends, browse the internet, file content management, create and manage documents, entertainment etc. This application is like those other mobile applications. It will help people to keep track of their blood sugar. It also have some extra features which will help people to keep aware about food habit and exercise.

## 6.2 Ethical Aspects

Like all other mobile applications this application also have some ethical aspects. Those are:

- No dark patterns
- Only needed permission request
- Empower users to protect themselves
- No promoting of quantity of engagement

## 6.3 Sustainability Plan

There are some key note which can be follow for sustainability plan. Those are:

- Main Emphasis Should Be On Solving Genuine Problems
- Establish Point of Difference
- Focus On Design Scalability
- Rapid Updates and Releases
- Focus on Users Feedback
- Work on App Efficiency
- Provide App Description

# CHAPTER 7

# Conclusion & Future Scope

## 7.1 Discussion and Conclusion

After analysis all other existing applications in the market we can say that, this Diabetes Fit application will be a suitable option for all types of user. This is not just an ordinary tracking application. It will gives the user more privilege to use. This application will perform like a personal assistant, which will remind about medicine, diet chart, food suggestion, and excises suggestion. At the end we just want to say that, In the long run it will be the next application of every blood sugar conscious user.

## 7.2 Scope for Further Developments

Our future agenda is to develop the application as per the user requirement and as per the perspective of other competitive applications. Moreover, we have a plan to use the user basic informatics and use those data for machine learning (ML). To find about that, which age people affects most in diabetes, which gender people affects most and about their blood sugar level statistics.

# References

[1] Google play store, available at <<Blood Sugar Log – Diabetes Tracker - Apps on Google Play>>, last accessed on 03-12-2021 at 6:00 PM.

[2] Google play store, available at << diabetes - Android Apps on Google Play>>, last accessed on 03-12-2021 at 6:15 PM.

[3] Google play store, available at << diabetes manager - Android Apps on Google Play>>, last accessed on 03-12-2021 at 6:35 PM.

[4] Mobile App Backend, available at << Mobile App Backend: How to Choose the Best One for your Application (os-system.com)>>, last accessed on 02-12-2021 at 10:35 PM.

[5] The requirement approach, available at <<https://www.javatpoint.com/software-engineering-requirement-analysis>>, last accessed on 05-12-2021 at 02:30 AM.

[6] Types of diabetes, available at <<https://www.healthline.com/health/diabetes/types-of-diabetes>>, last accessed on 07-11-2021 at 10:30 PM.

[7] Health risk at diabetes, available at <<https://www.healthline.com/health/diabetes/effects-on-body>>, last accessed on 07-11-2021 at 10:12 PM.

[8] About diabetes, available at <<https://www.idf.org/aboutdiabetes/what-is-diabetes.html>>, last accessed on 07-11-2021 at 09:56 PM.

[9] Application elements, available at <<https://developer.android.com/components/-fundamentals>>, last accessed on 08-11-2021 at 09:15 PM.

[10] About application services, available at << https://searchapparchitecture.techtarget.com/-definition/application-service-provider-ASP>>, last accessed on 09-11-2021 at 5:30 AM.

[11] Types of mobile applications, available at << https://clevertap.com/blog/types-of-mobile-apps/>>, last accessed on 09-11-2021 at 5:45 AM.

[12] Challenges behind building a mobile application, available at <<https://www.apogaeis.com/blog/6-biggest-challenges-in-mobile-app-development/>>, last accessed on 12-11-2021 at 9:45 AM

[13] About business process model, available at <<https://bpm.com/what-is-bpm>>, last accessed on 12-11-2021 at 10:45 AM.

[14] About use case modelling, available at <<https://www.cs.uct.ac.za/mit_notes/software/htmls/ch03s05.html>>, last accessed on 18-11-2021 at 11:35 AM.

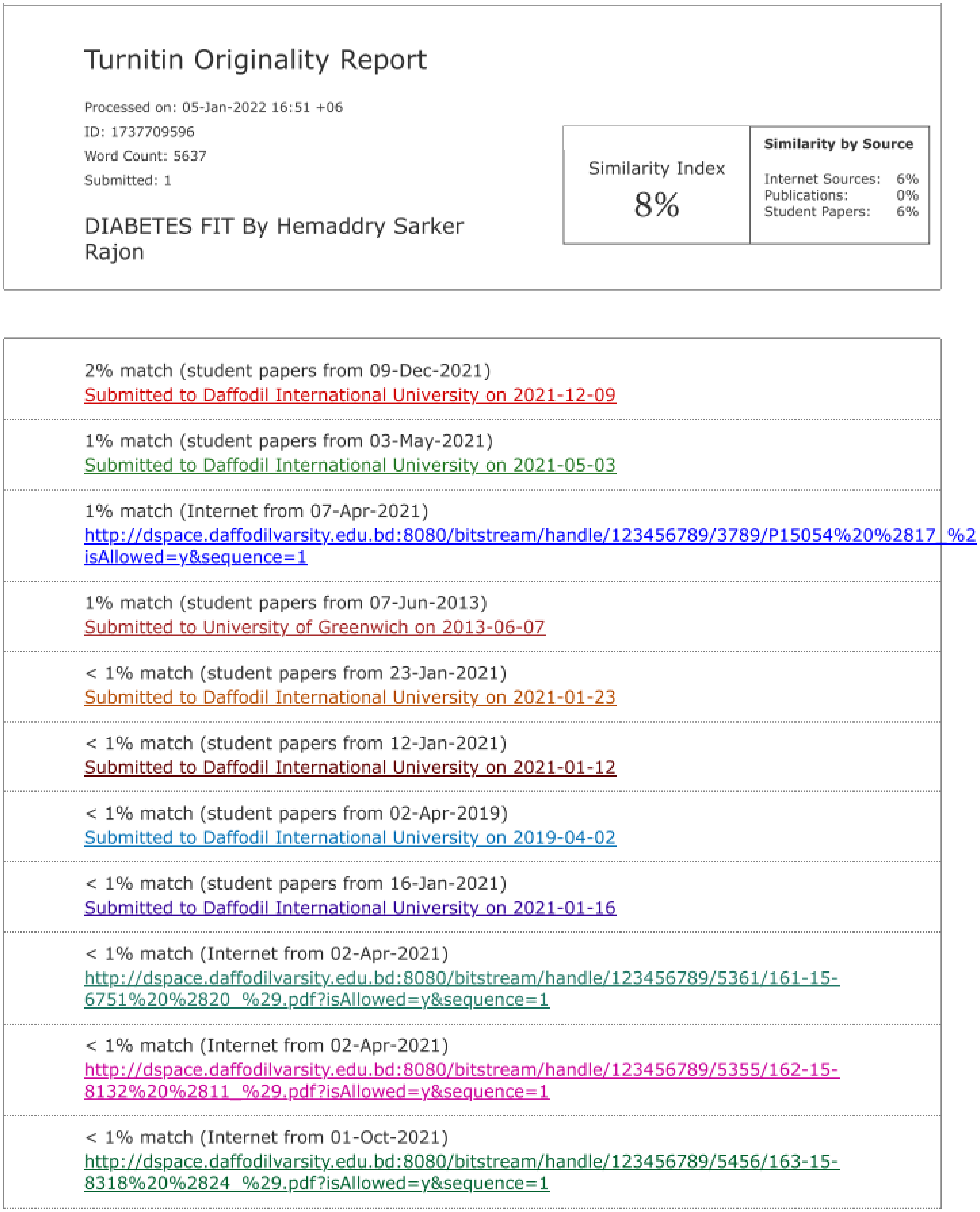
[15] About front end design, available at << https://www.goodfirms.co/glossary/front-end/>>, last accessed on 19-11-2021 at 10:41 AM.

[16] About back end design, available at << https://www.thedroidsonroids.com/blog/mobile-app-backend-development-guide-for-app-owners>>, last accessed on 19-11-2021 at 10:49 AM.

[17] About user interface, available at << https://www.interaction-design.org/literature/topics/ui-design>>, last accessed on 19-11-2021 at 01:49 PM.

[18] Types of user interface, available at << https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces>>, last accessed on 22-11-2021 at 01:55 PM.

# Plagiarism Report



Turnitin Originality Report

Processed on: 05-Jan-2022 16:51 +06
ID: 1737709596
Word Count: 5637
Submitted: 1

DIABETES FIT By Hemaddry Sarker Rajon

| Similarity Index | Similarity by Source | |
|---|---|---|
| 8% | Internet Sources: | 6% |
| | Publications: | 0% |
| | Student Papers: | 6% |

2% match (student papers from 09-Dec-2021)
Submitted to Daffodil International University on 2021-12-09

1% match (student papers from 03-May-2021)
Submitted to Daffodil International University on 2021-05-03

1% match (Internet from 07-Apr-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3789/P15054%20%2817_%2
isAllowed=y&sequence=1

1% match (student papers from 07-Jun-2013)
Submitted to University of Greenwich on 2013-06-07

< 1% match (student papers from 23-Jan-2021)
Submitted to Daffodil International University on 2021-01-23

< 1% match (student papers from 12-Jan-2021)
Submitted to Daffodil International University on 2021-01-12

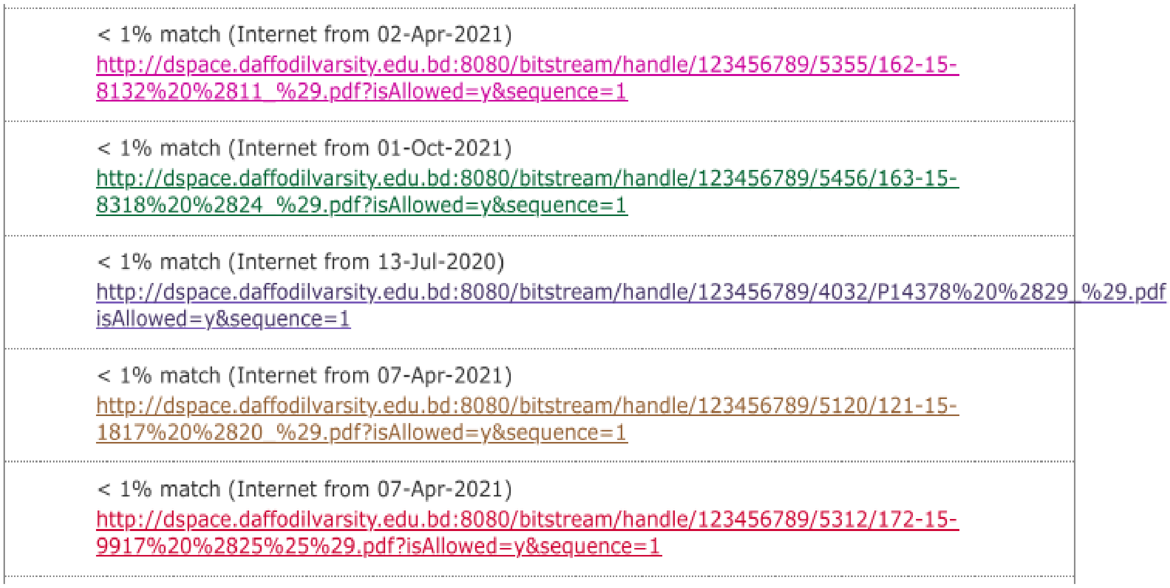
< 1% match (student papers from 02-Apr-2019)
Submitted to Daffodil International University on 2019-04-02

< 1% match (student papers from 16-Jan-2021)
Submitted to Daffodil International University on 2021-01-16

< 1% match (Internet from 02-Apr-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5361/161-15-6751%20%2820_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 02-Apr-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5355/162-15-8132%20%2811_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 01-Oct-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5456/163-15-8318%20%2824_%29.pdf?isAllowed=y&sequence=1



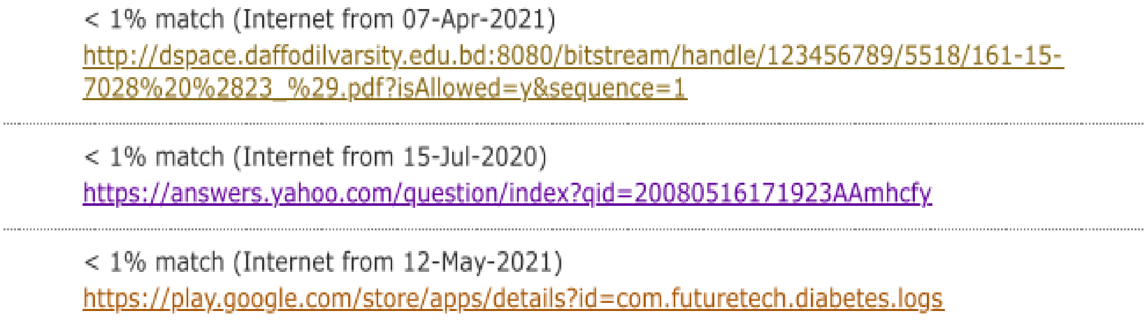
< 1% match (Internet from 02-Apr-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5355/162-15-8132%20%2811_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 01-Oct-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5456/163-15-8318%20%2824_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 13-Jul-2020)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4032/P14378%20%2829_%29.pdf isAllowed=y&sequence=1

< 1% match (Internet from 07-Apr-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5120/121-15-1817%20%2820_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 07-Apr-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5312/172-15-9917%20%2825%25%29.pdf?isAllowed=y&sequence=1



< 1% match (Internet from 07-Apr-2021)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5518/161-15-7028%20%2823_%29.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 15-Jul-2020)
https://answers.yahoo.com/question/index?qid=20080516171923AAmhcfy

< 1% match (Internet from 12-May-2021)
https://play.google.com/store/apps/details?id=com.futuretech.diabetes.logs