

# **MINI PROJECT**

**(2021-22)**

## **“SPOT FITNESS”**

### **Project Report**



**Institute of Engineering & Technology**

**Submitted By -**

Aditya Raj Singh (191500061)

Mayank Verma (191500449)

**Under the Supervision Of**

**Mr. Akash Kumar Choudhary**

**Technical Trainer**

**Department of Computer Engineering & Applications**



**Department of Computer Engineering and Applications**

**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**

**Chaumuha, Mathura – 281406 U.P (India)**

### **Declaration**

I/we hereby declare that the work which is being presented in the Bachelor of technology. Project “**Spot Fitness** in partial fulfilment of the requirements for the award of the **Bachelor of Technology** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mr. Akash Kumar Choudhary, Technical Trainer, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

**Sign:** *Aditya Raj Singh*

**Name of Candidate:** Aditya Raj Singh

**University Roll No.:**191500061

**Sign:** Mayank Verma

**Name of Candidate:** Mayank Verma

**University Roll No.:**191500449



**Department of Computer Engineering and Applications**

**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**

**Chaumuha, Mathura – 281406 U.P (India)**

**Certificate**

This is to certify that the project entitled “Spot Fitness Application”, carried out in Mini Project – I Lab, is a bonfire work by Aditya Raj Singh, Mayank Verma and is submitted in partial fulfilment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

**Signature of Supervisor:**

**Name of Supervisor:**

Mr. Akash Kumar Choudhary

**Date:**

## ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Mr Akash Kumar Choudhary, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

**Sign:** *Aditya Raj Singh*

**Name of Candidate:** Aditya Raj Singh

**University Roll No.:** 191500061

**Sign:** *Mayank Verma*

**Name of Candidate:** Mayank Verma

**University Roll No.:** 191500449

## **ABSTRACT**

In this project, we are creating an android application, basically a multitasking app which we have named “SPOT FITNESS”. This application will provide us a platform to perform the exercise in online mode. All the users will be having their separate accounts on this app which they can access by entering their specific id and password. Users can also use other sign In options like using email (provided by institution), using otp (using the mobile number registered in their institution). The app is completely efficient and transparent. The app also has a complete User Interface attached to the firebase a perfect login system with email id and password and a forget password too. Android App ecosystem is diverse and is changing people’s life all over the world. Android users are expected to increase because of the advance changes of the operating system and the way it deals with issues and compatibility with other mobile devices. Furthermore, designing solutions for the problems that we may face in future is essential. Like this application definitely stands the need of users to make home workout procedure easy and much efficient.

# CONTENTS

Cover Page

Declaration

Certificate

Training Certificate

Acknowledgement

Abstract

Content

List Of figures

List Of tables

Chapter 1 Introduction

- 1.1 Context
- 1.2 Motivation
- 1.3 Objective
- 1.4 Existing System
- 1.4 Sources

Chapter 2 Software Requirement Analysis

- 2.1 Impact Of Spot Fitness App
- 2.2 Problem Statement
- 2.3 Hardware and Software Requirements
- 2.4 Modules and Functionalities
- 2.5 Spot Fitness App

Chapter 3 Software Design

- 3.1 Use Case Diagram
- 3.2 Data Flow Diagram
- 3.3 Sequence Diagram

Chapter 4 Technology Used

- The Difference between Eclipse and Android Studio...

## Chapter 5 Implementation and User Interface

- 5.1 Implementation of Spot Fitness App
- 5.2 User Interface

## Chapter 6 Testing

- 6.1 Installation Testing
- 6.2 Unit Testing
- 6.3 User Testing
- 6.4 Performance Testing
- 6.5 Compatibility Testing

## Chapter 7 Conclusion

## References

# LIST OF FIGURES

1. Existing System
2. Use Case Diagram
3. Data Flow Diagram
4. Sequence Diagram
5. Android Kit-Kat
6. Flow Chart for User
7. Navigation Drawer
8. Dashboard

# CHAPTER-1

## INTRODUCTION

### 1.1 CONTEXT

This Android Application “SPOT FITNESS” has been submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University

### 1.2 MOTIVATION

In the recent years, we have realized the importance of digital world and how important it is for us to have our resources online. Mobile applications have been the greatest source to reach different resources which we need digitally and having them at the reach of our fingertips would be an opportunity.

Moreover, this kind of application can be used by the users to make the online guided exercise and efficient than the existing system. This would be an excellent effort to provide a simple alternate way for offline gym exercises procedure which is complex and time taking.

### 1.3 OBJECTIVE

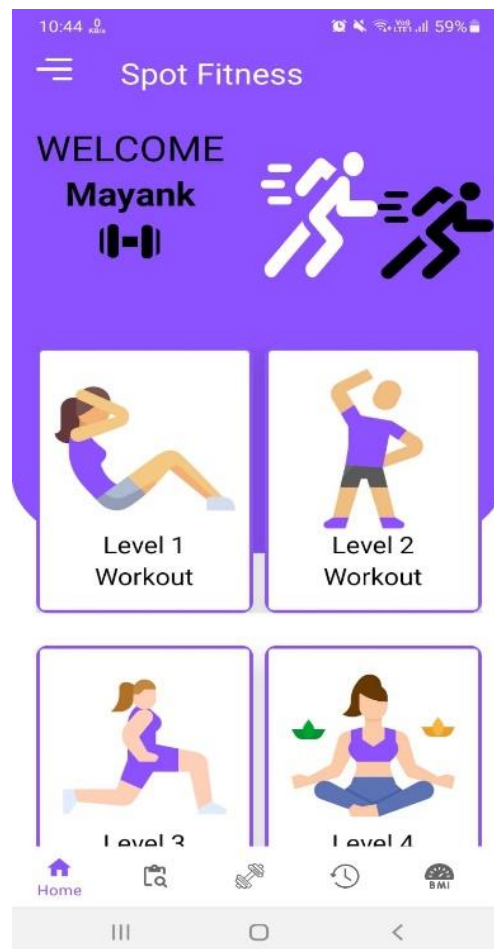
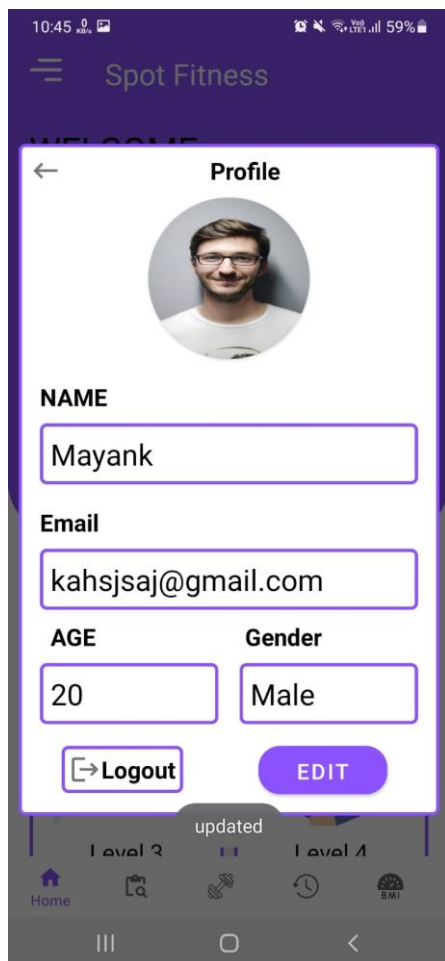
The main objective of this application is to create an app named “SPOT FITNESS” which will provide an option to perform online guided exercise and efficient than the existing system. This would be an excellent effort to provide a simple alternate way for offline gym exercises procedure which is complex and time taking.

The goal of the app was to provide a platform for online guided exercise



## 1.4 EXISTING SYSTEM

In the present scenario, we are having offline or we can say manual gym system for exercise. With the help of this application, we are able to find a platform where we can perform in gym exercise at home. Here some of the snap how our application will look.



## CHAPTER -2

# SOFTWARE REQUIREMENT ANALYSIS

### 2.1 IMPACT OF Fitness ON DAILY LIFE

The “SPOT FITNESS” is an Android Application which will allow the users to perform the home workout as it is much easy and efficient way.

**Exercise strengthens your heart and improves your circulation.** The increased blood flow raises the oxygen levels in your body. This helps lower your risk of heart diseases such as high cholesterol, coronary artery disease, and heart attack. Regular exercise can also lower your blood pressure and triglyceride levels

]

### 2.2 PROBLEM STATEMENT

The “SPOT FITNESS” is an Android Application which will allow the users to perform the home workout as it is much easy and efficient way.

### 2.3 HARDWARE AND SOFTWARE REQUIREMENTS

#### Hardware Requirement

Processor: any smart phone processor

- RAM : 8 GB (or higher)
- Hard disk : 256GB

#### Software Requirement

- Software used: Android Studio
- Language used : Kotlin, XML
- Database: Firebase
- User Interface Design : Android Application

## 2.4 MODULES AND FUNCTIONALITIES

⑩ **Login:** This page is for those users who have already registered on the app and have an id and a password. There is also a way to login with Gmail and otp (with mobile number registered with institution).

⑩ **Start Exercise:** This is the page is to start exercise of level of your own type.

⑩ **Nutrition Search:** In this you can about nutrition of different type of foods

⑩ **Exercise page:** this will contain all the exercise

⑩ **History page:** this will contain the data of all exercises.

⑩ **BMI page:** in this page you can calculate your BMI index

- **Logout on profile:** Then is this last panel for the users to sign out from the account. As soon as the users sign out, they are brought back to the login page.

## TECHNOLOGY USED

### ANDROID

Android is a Linux -based operating system designed primarily for touch screen devices such as smart phone tablets and computers. Released in 2008, is now owned by Google. So android is an operating system like Windows, Ubuntu and Mac OS and a lot number of devices use Android these days like mobile phones, watches, laptop and television. So, we also created an android application “**Spot Fitness**”. An Android app is software running on an Android Platform. So, this can be concluded that like all the software it is a combination of Backend and Frontend. Backend is to design the logical parts of the app, for the functionality whereas Front End to develop the User Interface. And to implement the various parts of the android app, we require a number of tools and technologies which will come into picture. But first it would be great to see the three different types of Android Apps: -

- **Native Apps:** An executable program coded in the machine language of the hardware platform it is running in. **Native applications** are compiled into the machine language of that CPU. Native apps are the most common. They're coded in a specific language. A popular example is WhatsApp.
- **Web Apps:** are accessed via the internet browser and will adapt to whichever device you're viewing them on. They are not native to a particular system,

and don't need to be downloaded or installed. Due to their responsive nature, they do indeed look and function a lot like mobile apps — and this is where the confusion arises.

- **Hybrid Apps:** Hybrid apps are deployed in a native container that uses a mobile Web View object. When the app is used, this object displays web content thanks to the use of web technologies (CSS, JavaScript, HTML, HTML5). It is in fact displaying web pages from a desktop website that are adapted to a Web View display. The web content can either be displayed as soon as the app is opened or for certain parts of the app only i.e., for the purchase funnel. In order to access a device's hardware features (accelerometer, camera, contacts...) for which the native apps are installed, it is possible to include native elements of each platform's user interfaces (iOS, Android): native code will be used to access the specific features in order to create a seamless user experience. Hybrid apps can also rely on platforms that offer JavaScript

APIs if those functionalities are called within a Web View

## CHAPTER- 3

# SOFTWARE DESIGN

### 3.1 USE-CASE DIAGRAM:

#### **Figure-2: Use–Case Diagram**

So the above diagram represents the point of view of the new user, the registered user, and the developer and the arrows to each module show the interactivity of the person.

The New user will first be required to create up a new account so will interact with the “sign-up module” and fill up all the details that will be stored in the database. The next user will land into the dashboard where there will be the home page of application .

“Nutrition” where the user can enter the name of any food for which he wants to know its nutrition values likes protein , carbohydrate , fat , calcium , vitamin, etc.

For the registered user, the user will be having the credentials to login and will interact with the “login module” and then the user will enter into the dashboard

To remember what have done user can check the history where all the data has saved that what user have perform last time and it will store the all time data what he have done from the first day of installation of the app.

User can also check the BMI by entering his/her details as per asked there are two types of formats In which user an enter data

Metric Units – Here user have to enter weight in Kilograms and height in Centimetre.

Us Units - Here user have to enter weight in Pounds and height in Foot and Inches.

## CHAPTER-4 TECHNOLOGY USED

### 4.1 ANDROID

Android is a linux-based operating system designed primarily for touch screen devices such as smart phone tablets and computers. Released in 2008, is now owned by Google. So android is a operating system like Windows, Ubuntu and Mac OS and a lot number of devices use Android these days like mobile phones, watches, laptop and television. So we also created an android application “Spot Fitness”. Play Store is a market place for all the Android Apps. So we need to know what basically an android app is. An Android app is software running on a Android Platform. So this can be concluded that like all the software it is a combination of Backend and Frontend. Backend to design the logical parts of the app, for the functionality whereas Front End to develop the User Interface. And to implement the various parts of the android app, we require a number of tools and technologies which will come into picture. But first it would be great to see the three different type of Android Apps:-

- **Native Apps:** An executable program coded in the machine language of the hardware platform it is running in. **Native applications** are compiled into the machine language of that CPU. For example, **Windows** and Mac executable **apps** are in x86 machine language, while **mobile apps** are ARM based. Native apps are the most common. They're coded in a specific language like Swift for **iOS** or Java for Android. A popular example is WhatsApp.
- **Web Apps:** are accessed via the internet browser and will adapt to whichever device you're viewing them on. They are not native to a particular system, and don't need to be downloaded or installed. Due to their responsive nature, they do indeed look and function a lot like mobile apps — and this is where the confusion arises.
- **Hybrid Apps:** Hybrid apps are deployed in a native container that uses a mobile Web View object. When the app is used, this object displays web content thanks to the use of web technologies (CSS, JavaScript, HTML, HTML5). It is in fact displaying web pages from a desktop website that are adapted to a Web View display. The web content can either be displayed as soon as the app is opened or for certain parts of the app only i.e. for the purchase funnel. In order to access a device's hardware features (accelerometer, camera, contacts...) for which the native apps are

installed, it is possible to include native elements of each platform's user interfaces (iOS, Android): native code will be used to access the specific features in order to create a seamless user experience. Hybrid apps can also rely on platforms that offer JavaScript APIs if those functionalities are called within a Web View

## 4.2 VERSION OF ANDROID

Each year Android releases a new version with better features, better security and better User Interface experience and a new symbol. Here is the table of list of versions



**Figure-5: Android Kitkat**

Code name	Version number	Initial release date
(No codename)	1.0	September 23, 2008
Petit Four	1.1	February 9, 2009
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15, 2009
Eclair	2.0 - 2.1	October 26, 2009
Froyo	2.2 - 2.2.3	May 20, 2010
Gingerbread	2.3 - 2.3.7	December 6, 2010
Honeycomb	3.0 - 3.2.6	February 22, 2011
Ice Cream Sandwich	4.0 - 4.0.4	October 18, 2011
Jelly Bean	4.1 - 4.3.1	July 9, 2012
KitKat	4.4 - 4.4.4	October 31, 2013
Lollipop	5.0 - 5.1.1	November 12, 2014
Marshmallow	6.0 - 6.0.1	October 5, 2015
Nougat	7.0 - 7.1.2	August 22, 2016
Oreo	8.0 - 8.1	August 21, 2017
Pie	9.0	August 6, 2018

**Table -1: Versions of Android**



## 4.3 TOOLS AND LANGUAGES

Tools used to build the Android App are:-

- **Android Studio:** Android Studio is an environment that help us create and edit Android applications. It is the official IDE for Android App Development. It has IntelliJ's powerful code editor and developer tools and various features that enhance productivity while developing apps.
- **Software Development Kit (SDK):** Android Studio requires a collection of libraries and data therefore SDK is mandatory.

Languages used in building an Android Application are classified as per the Front End and Back End. For designing the Front End of an application we have used XML and for designing the Back End we have used Kotlin.

- **XML:** XML is the extensible Markup Language. It is the met language which allows users to define their own customized markup language especially in order to display documents on Internet. It is the language that contains tags that store information. And the tags can be used to present data on the screen.
- **Kotlin:** Kotlin is statically typed programming language based on Java Virtual Machine. Kotlin is the fundamental language of Android since 2017 as declared by Google. Developers of Android also prefer to use Java for the backend but Kotlin has a upper-hand due to many features like Java has a lengthy syntax and hence sometimes the code is also redundant. To remove the boiler Plate code, Kotlin is preferred .Kotlin is cross platform, general purpose programming language with type inference. It can interoperate fully with Java but type inference allow its syntax to be more concise.

## 4.4 BASIC TERMINOLOGY

- **Layout:** Layout is the parent of view. It arranges all the views in a proper manner on the screen.

- **Activity**: An activity can be referred as your device's screen which you see. User can place UI elements in any order in the created window of user's choice.
- **View**: A view is an UI which occupies rectangular area on the screen to draw and handle user events.
- **Emulator**: An emulator is an Android virtual device through which you can select the target Android version or platform to run and test your developed application.
- **Manifest file**: Manifest file acts as a metadata for every application. This file contains all the essential information about the application like app icon, app name, launcher activity, and required permissions etc.
- **API**: Short for Application Programming Interface. APIs are functions that developers can call on to access specific features by calling upon programs, code, and services that others have written. For example, if a developer wants to draw a button on the screen, she can insert a small bit of code that says “draw this kind of button, with this color and size and style, at this location” instead of dozens of lines of code that tells the graphics processor, in detail, exactly how to draw a button. If the application wants your location, it can use the location API to “get the device's location” and let Google's code handle the rest, instead of requiring the developer to build an entire location service from scratch just for her own app. There are thousands of APIs in Android, covering everything from drawing interface elements, to the cameras, to location access, to accessing storage, to 3D graphics (see: OpenGL ES) and much more.
- **Intent**: Intents are an essential part of the Android ecosystem. They are used to express an action to be performed. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. It can be classified into implicit and explicit intents.

- **Implicit intent:** It does not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it.
- **Explicit Intent:** It specifies the component to start by name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start.
- **APK:** Short for "Android application package." The extension used in Android app installation files (e.g., app.apk). Similar in nature to an EXE file on Windows.
- **SDK:** Short for "Software Development Kit." As it pertains to Android, the SDK is a set of tools such as code libraries, a debugger, and a handset emulator that can be run on Windows, Mac, or Linux to facilitate the creation of Android apps by developers. While the SDK is generally intended for use by developers, end users can install the software on their home computer to execute ADB and Fast boot commands.
- **Action Bar:** The action bar is an important design element, usually at the top of each screen in an app that provides a consistent familiar look between Android apps. It is used to provide better user interaction and experience by supporting easy navigation through tabs and drop-down lists.
- **Navigation bar:** Android Navigation Drawer is a sliding left menu that is used to display the important links in the application. Navigation drawer makes it easy to navigate to and fro between those links. It's not visible by default and it needs to be opened either by sliding from left or clicking its icon in the Action Bar.
- **Fragment:** A Fragment represents a behavior or a portion of user interface in a Fragment Activity. You can combine multiple fragments in a single activity to build a multi-pane UI and reuse a fragment in multiple activities.
- **Firebase** is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is

built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents. Firebase has three core services: a real-time database, user authentication and hosting. With the Firebase iOS SDK, you can use these services to create apps without writing any server code.

**JSON** stands for JavaScript Object Notation. It is an independent data exchange format and is the best alternative for XML. JSON is used for data interchange (posting and retrieving) from the server. Hence knowing the syntax and its usability is important.

JSON is the best alternative for XML and its more readable by human

## **CHAPTER -5**

### **IMPLEMENTATION AND USER INTERFACE**

Creating an app concept design with screen sketches and functional flow diagrams is the best way to communicate your vision to the mobile app developer. Making the concept clear to the developer is probably the most important factor in successful mobile app development. Yet it is one of the most common problems or obstacles in a mobile app development outsourcing project.

No matter what the marketing and profit goals are or if you are outsourcing an app for your personal use, you need to fully design and document the app concept if you expect a programmer to make your vision a reality. Developers are not mind readers and even descriptions given during conversations can be very fleeting or interpreted differently. Fully documenting your concept, therefore, leaves little to chance. The two most important things to do are: A) make a comprehensive description of how the app works and what it does (functionality) and B) create a comprehensive description of what the user sees and does (look and feel).

#### **5.1 Implementation of the Spot Fitness:**

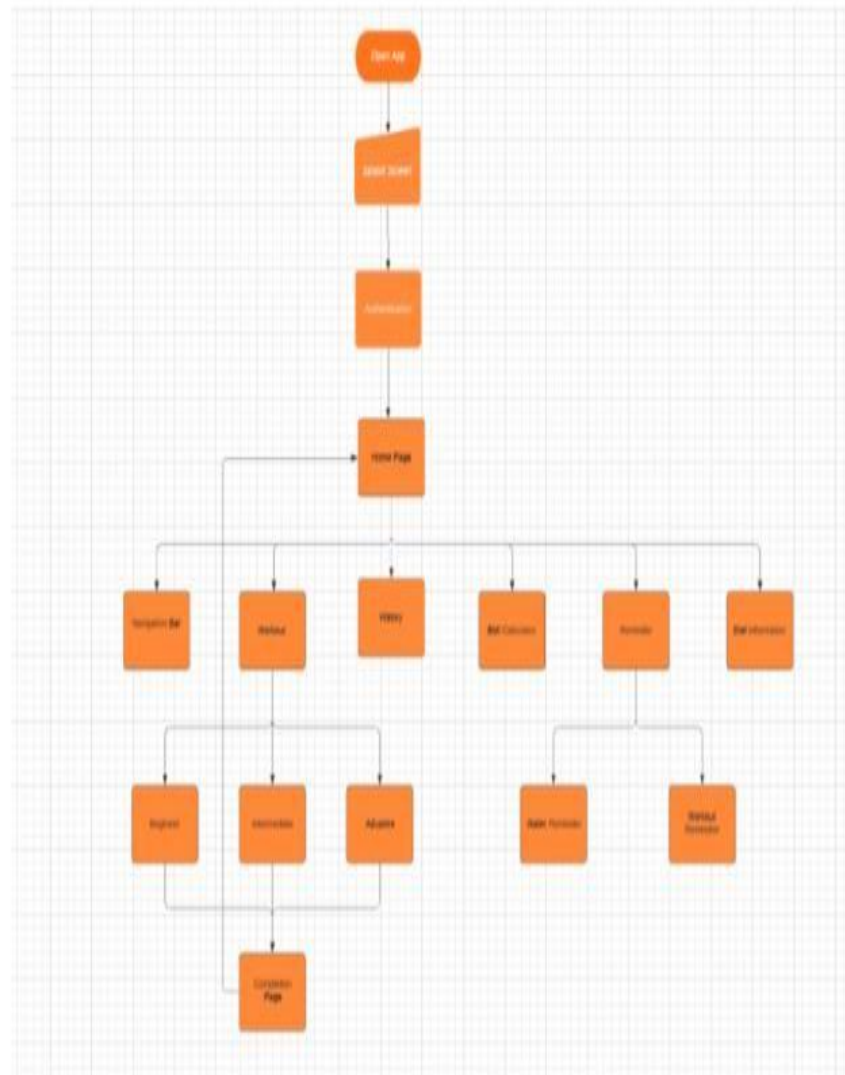
Implementation of this application is taken place in various phases. Firstly we build the login interface then Navigation drawer i.e. make fragment for each of the list item using the Navigation view and the make various layout for the supporting features and connect the app with the Google API for fetch the required nutrition value. And finally we parse the Jason object to get the data in the required format and then display the result.

##### **5.1.1 Step to be followed to develop the app:**

1. Firstly we create the splash screen with animated text using XML and linked it with the main Activity through Kotlin.
2. After that we create login phase which comprises of various phases that are mentioned below:
  - Login Page: allows user to login into the app if the user is existing one
  - Register Page: If the user is new to our app then firstly he/she have to register themselves on the app.
  - Forgot Password: allows user to reset the password if it forget the previous password.

- **For authenticating the user we have used firebase authentication.**
3. Now, we are going to create Navigation drawer for that purpose we have used following functionality of android:
    - Fragments(SupportFramentManager)
    - Menu – items
    - Drawer header
    - Hamburger icon
    - ActionBarDrawableToggle (help to create navigation Bar)
  4. .Creating fragment for each of the menu item. Our Menu items are:
    - Dashboard
    - Profile
    - Sign-Out
  5. Now we have created various activities like Home , Nutrition , History , Bmi , Workout Description and many more.
  6. In this step we connect our app with the API .
  7. After that we parse the JASON object that we have received as a response for our query to get the data in the standard form.
  8. Now we add data (that we have received from API) to the Nutrition activity .

Flow Chart for the User is given below:



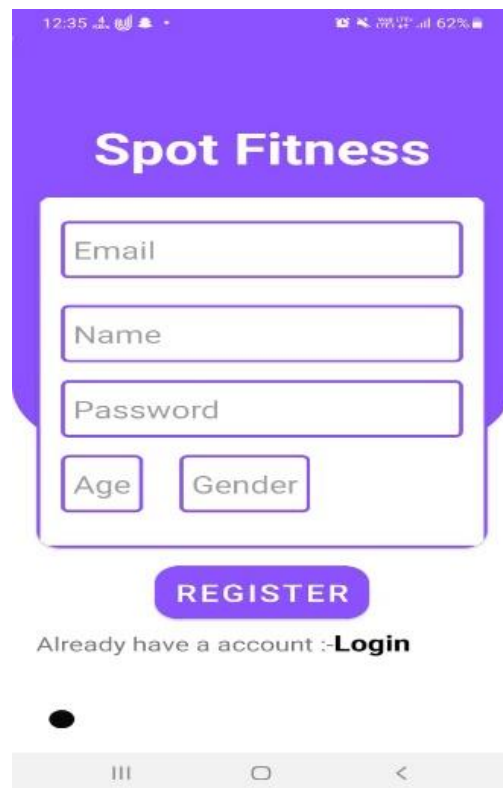
**Figure-6: Flow Chart for User**

### 5.1.2 Step to be followed by the user

1. Firstly, we have build splash activity to start the application.
2. Then, we have the Login activity which consists of following steps
  - Register : for new User
  - Login: For existing as well as new user

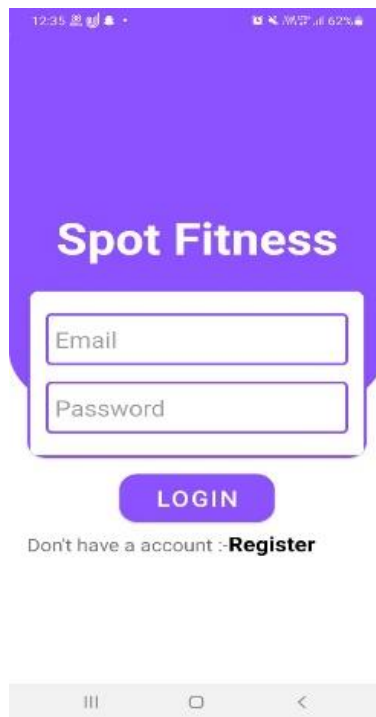
- Forgot Password: To reset your password
- 3. We authenticate and store the user information from the Firebase authentication.
- 4. After that, we made a Drawer layout of our app which includes various functionality
  - Home Fragment: Here various exercises according to the age.
  - Nutrition Fragment: To find the nutrition value of any item.
  - Workout : Various types of excursive available here
  - History Fragment: It contains history of workouts has been done.
  - Profile Fragment: Remove/logout you from the app.

## 5.2User Interface

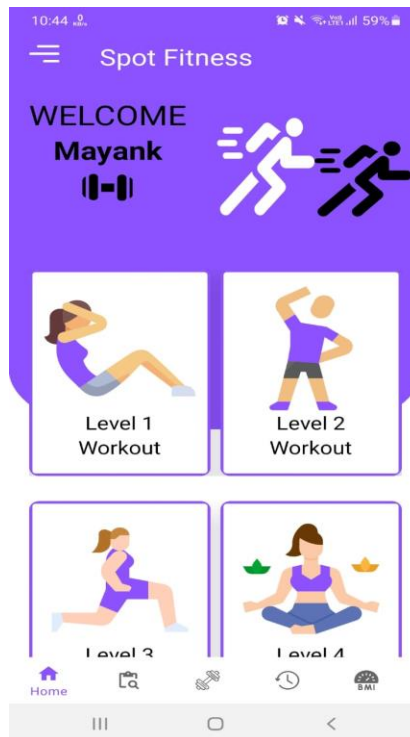


- **Login Page**

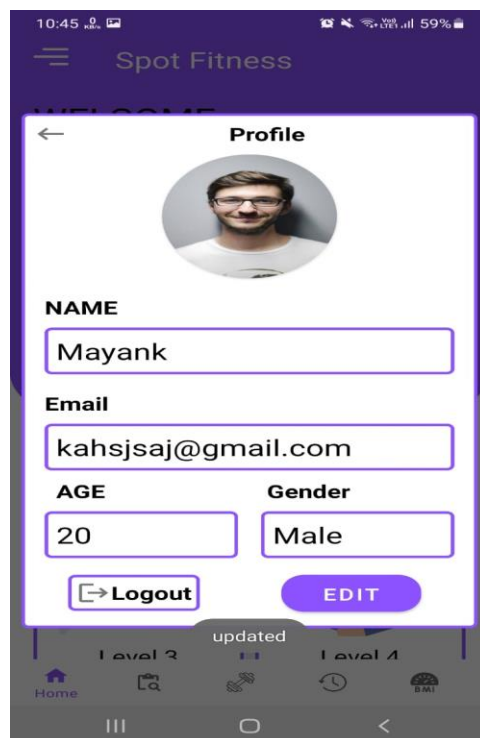




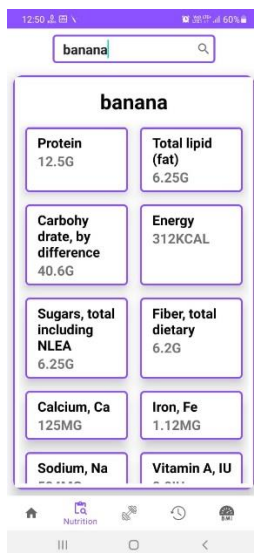
- **Dashboard Fragment**



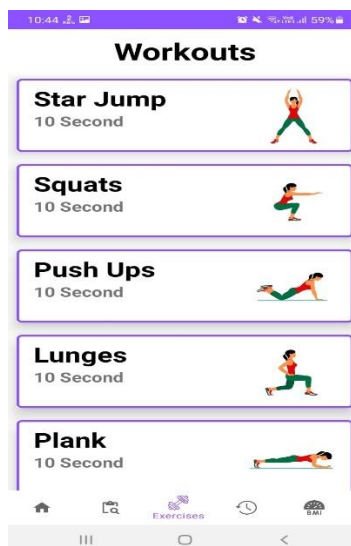
- **Profile**



## • Nutrition Page



## Workout Page



## Bmi Page:

10:44

METRIC UNITS US UNITS

WEIGHT (in kg)

HEIGHT (in cm)

CALCULATE

Home Search Settings History BMI

Navigation icons: Home, Search, Settings, History, BMI

10:45

METRIC UNITS US UNITS

WEIGHT (in lbs)

Feet Inch

CALCULATE

Home Search Settings History BMI

Navigation icons: Home, Search, Settings, History, BMI

- History Page

10:42

### History

Mon, 29 Nov 2021, 00:23

Mon, 29 Nov 2021, 22:42

Home Search Settings History BMI

Navigation icons: Home, Search, Settings, History, BMI

# CHAPTER – 6

## TESTING

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing include designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques.

System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data.

In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The Android framework includes an integrated testing framework that helps you test all aspects of your application and the SDK tools include tools for setting up and running test applications. Whether you are working in Eclipse with ADT or working from the command line, the SDK tools help you set up and run your tests within an emulator or the device you are targeting.

There are different types of testing some of them are listed below:

### **6.1 Installation Testing:**

There are two types of apps on an Android device i.e., Pre-installed applications and the applications which are installed later by the user.

For both of the above, installation testing is carried out by our teammates. It is ensuring smooth installation of the application without ending up in errors, partial installation etc.

### **6.2 Unit Testing**

It focuses on smallest unit of software design. In this we test an individual unit or groups of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs. In this testing technique we are primarily focuses on

- Loop methods and function is working fine or not. □ Misunderstood or incorrect Arithmetic precedence
- Incorrect Initialization

**Unit Testing of the app:**

Test cases	Description	Expected Outcome	Result
1	Start Page – Launch Screen	Should display splash screen with animated text	Pass
2	Register Screen	Should display register activity where you need to fill the required details	Pass
3	Login Screen	Should display login screen And ask for your credentials.	Pa4ss
4	Home Page	Should display all the fragment option	Pass
5	Nutrition Page	Should get the data from the API	Pass
7	Workout	Should display the exercises	Pass

8	History	Should Display the workout history	Pass
9	Bmi Fragment	Should calculate the bmi by entered values	Pass

**Table 1: Unit Testing of Spot Fitness**

### 6.3 User Testing

User testing is the process through which the interface and functions of a website, app, product, or service are tested by real users who perform specific tasks in realistic conditions. The purpose of this process is to evaluate the usability of that website or app and to decide whether the product is ready to be launched for real users.

This app was tested by our team mates and friends who are using different mobile phones (and having different android version) also tested on different emulator to check its performance and it seems to be working fine and users of this app are satisfied with the facilities and performance of the app and like the way how the app is worked.

### 6.4 Performance Testing

In this type of testing we have checked the performances of our application under some peculiar conditions are checked. Those conditions include:

- Low memory in the device.
- The battery in extremely at a low level.
- Poor/Bad network reception.

Performance is basically tested from 2 ends, application end, and the application server end. Our app is also performing well in this phase of testing as well. And we are getting positive feedback from user of our app.

### **6.5 Combability Testing**

This application was tested and used on different devices like LG G3, Google Nexus 4. The application worked fine and is stable. The application worked fine in portrait mode and there isn't any problem with compatibility.

On all types of testing (that we have performed above) our performing well on our app i.e.

## **CHAPTER -7**

### **CONCLUSION**

The Spot Fitness app application meets with the enterprise class application principles. It is designed to be performing, scalable, extensible, and highly available. It also ensures the privacy of the users' data and secures its access. Given that it may be improved in many ways, the application is also easily maintainable. This document summarizes the work that has been done since the beginning of this semester. Indeed, it starts by giving an overview about the project specification and requirements. The document also states the methodology followed and which consists of 5 main parts: The first part will be devoted to data gathering and software requirements specification. Consequently, I will have a look at different mobile apps which target the same goal. They are plenty of Fitness apps. Each one has some various features. The second part will be dedicated to the design phase, including the app and the database. Also, in this phase, the software tools to be used will be specified. For example, the IDE, the database Server, the modelling language for the design, and finally the software testing tools. The third part will be the implementation phase, here, the design will be converted to code in order to develop the targeted app. The fourth step will be devoted to testing the app. In this phase, two testing methods will be used, namely: Black Box testing and White box testing. The last phase will be the deployment phase.



## REFERENCES

**1. Introduction to Android:**

<http://developer.android.com/guide/index.html>.

**2. *Android API:***

<http://developer.android.com/reference/packages.html>

**3. *Android User Interfaces:***

<http://developer.android.com/guide/topics/ui/index.html>

**4. *Layout:***

[http://developer.android.com/guide/topics/ui/declaring-  
layout.html](http://developer.android.com/guide/topics/ui/declaring-layout.html)

**5. *Android Training:***

<http://developer.android.com/training/index.html>.

**6. Internshala: <https://trainings.internshala.com/>**

**7. Android developer Guide: <https://developer.android.com/>**

**8. For rectifying the error :**

<https://stackoverflow.com/>