GETFIT- A FITNESS APPLICATION

A MINI PROJECT REPORT SUBMITTED TO THE BHARATHIAR UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS

Submitted by

SOLOMON PRABU D

(Reg. no: 21CSEA32)

Under the guidance of

Dr. GAVASKAR S, MCA., Ph.D.,

Assistant Professor

Dr. MOHANA S, MCA., M.Phil., Ph.D.,

Assistant Professor

DEPARTMENT OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS

BHARATHIAR UNIVERSITY

COIMBATORE-641046.

DECEMBER - 2022

CERTIFICATE

CERTIFICATE

This is to certify that, this mini-project work entitled "**GETFIT- A FITNESS APPLICATION**" was submitted to the Department of Computer Applications, Bharathiar University in partial fulfillment of the requirements for the award of the degree of **Master of Computer Applications**, is a record of original work done by **SOLOMON PRABU D (21CSEA32),** during his period of study in the Department of Computer Applications, Bharathiar University, Coimbatore, under my supervision and guidance, and this project work has not formed the basis for the award of any Degree/ Diploma /Associateship/ Fellowship or similar title to any candidate of any University.

Place: Coimbatore

Date:

Project Guide **Head of the Department**

Submitted for the University Viva-Voce Examination held on ___________

Internal Examiner **External Examiner**

DECLARATION

DECLARATION

I hereby declare that this mini project work titled, "**GETFIT- A FITNESS APPLICATION**" submitted to Department of Computer Applications, Bharathiar University, is a record of original work done by **SOLOMON PRABU D (21CSEA32)**, under the supervision and guidance of **Dr. GAVASKAR S, MCA., Ph.D., Assistant Professor** and **Dr. MOHANA K, MCA., M.Phil., Ph.D. ., Assistant Professor** Department of Computer Applications, Bharathiar University, and that this project work has not formed the basis for the award of any Degree/ Diploma/ Associateship/ Fellowship or similar title to any candidate of any University.

Place: Signature of the candidate

Date:

(**SOLOMON PRABU D**)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I express my sincere gratitude to our Professor& Head of the Department **Dr. T. DEVI, M.C.A., M.Phil., Ph.D. (UK),** Department of Computer Application, Bharathiar University, Coimbatore for acknowledging our request and permitting me to undertake this project work.

I am forever indebted to **Dr. S GAVASKAR, MCA., M.Phil., Ph.D., Assistant Professor & Dr. K. MOHANA, Guest Faculty** Department of Computer Applications, Bharathiar University for providing valuable suggestions and always been enthusiastically guiding us throughout the project.

Finally, I also extend my special thanks to my family, friends, who have kindly provided the necessary support for the successful completion of the project and their moral support.

ABSTRACT

ABSTRACT

This project entitles “GETFIT- A FITNESS APPLICATION” is an android application mainly focuses on the physical intake of food, that helps an individual to be aware of what amount of nutrients is present in he/she eats. This app calculates the Body Mass Index (BMI) of an individual and lets him know under what category his physic is present, this helps him/her to achieve fitness (i.e., the normal BMI rate 19-24). The application was inspired by the FIT INDIA movement Launched by the Honourable Prime Minister.

The individual can update all his food intakes and can track them over a period of time, this can improve his/her food habits and help them to live a healthy life. Being fit, lowers obesity and other fats which will help them to stay away from diseases and disorders such as cancer and diabetes. Unhealthy food intake can also cause digestive problems and food poisoning.

INDEX

INTRODUCTION

Chapter 1
INTRODUCTION

This chapter starts with a high-level overview of the project. It then describes the specific aims and objectives of the project. Finally, it analyses the feasibility of the project and provides a feasibility report of the system.

1.1 Background Study

In India, Fitness is one of the main concerns of the Government as they have included many programs for the fitness of the country. FIT INDIA Movement was launched on 29th August, 2019 by Honorable Prime Minister with a view to make fitness an integral part of our daily lives. The mission of the Movement is to bring about behavioral changes and move towards a more physically active lifestyle. Towards achieving this mission, Fit India proposes to undertake various initiatives and conduct events to achieve. This inspired me to make an application that helps in fitness goal not only through physical activity but also through daily diet.

1.1.1 Application Overview

This android application getfit, only focuses on the physical intake of food, that helps an individual to be aware of what amount of nutrients is present in he/she eats. This app calculates the Body Mass Index (BMI) of an individual and lets him know under what category his physic is present, this helps him/her to achieve fitness (i.e the normal BMI rate 19-24).

The individual can update all his food intakes and can track them over a period of time, this can improve his/her food habits and help them to live a healthy life. Being fit, lowers obesity and other fats which will help them to stay away from diseases and disorders such as cancer and diabetes. Unhealthy food intake can also cause digestive problems and food poisoning.

1.2 Problem Statement

In the current model world, there are plenty of foods available in all types and varieties, each food has different composition of nutrients, specifically in India people have a vast amount of food and different food habits. So, it is really a hard to task to maintain fitness and keep track of all the intakes he/she has in a day and to know what type of nutrients those foods lacks. To solve this problem getfit will be a good solution to know what they eat and to be aware of bad food habits;

1.3 Aim and Objectives

In the quest to design a successful system to tackle the issues stated in the problem statement, the aim and objectives of the project are outlined below.

1.3.1 Aim

The aim of this project is to design and implement a user-friendly real-time fitness tracking Android Mobile Application.

1.3.2 Objectives

The objectives include,

1. To know the calculating formula of BMI, and to show the BMI value for the given weight and height.
2. To give a detailed nutrient value for the desired food by the individual.
3. To show the daily intake ratio of carbohydrates, fat and protein.
4. To store, update and view all intakes by the individual in cloud over a period of time.
5. Run simulations and compare the results of all the intakes and other application systems.

1.4 Significance of the Project

In the advanced world of smartphones, and rapid development of innovation that reduces one's physical exercise, it is important for everyone to stay fit at least through diet methods. So, the application helps in exactly what type of food the individual is eating and what amount of nutrients is present in it. Being fit can make one's life stress free and disease free.

1.5 Scope/Limitations of the Work

- The application mainly focuses on the data entered by the individual, so it is hard to have continuous data collection without any breakage, the individual may not remember to update the food he/she ate. So, loss of information occurs which might affect the accuracy of the application.
- In our current application food tracking isn't possible without the individual's support.
- It requires network access: Since the collection and sending of votes to the database requires an internet access which may not be readily available in some urban area would seem a limiting factor, though the local database and the printed vote can be used for counting until network is restored.

SYSTEM SPECIFICATION

Chapter 2

SYSTEM SPECIFICATION

In this, chapter we will see the system specifications that were used to create the android application, such as hardware specifications, software specifications and selected software description.

2.1 Hardware Specifications

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list like (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application.

The following subsections discuss the various aspects of the hardware requirements.

➢ System	: Ryzen 5
➢ Storage Type	: SSD
➢ Storage Capacity	: 512 GB
➢ Mobile	: Android
➢ RAM	: 8 GB

2.2 SOFTWARE SPECIFICATIONS

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the installation of the software.

➢ Operating System	: Windows OS

➢ Front End	: XML
➢ Coding Language	: JAVA
➢ Database	: Firebase
➢ IDE	: Android Studio

2.3 Selected IDE Description:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. The current version that was used to develop this application is, Dolphin 2021.3.1.

Languages supported: Android Studio supports all the same programming languages of IntelliJ and CLion e.g., Java, C++, and more with extensions, such as Go and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version."

2.3.1 Why Android Studio?

- Code and iterate faster: Based on IntelliJ IDEA, Android Studio provides fast turnaround on your coding and running workflow.
- Fast and feature-rich emulator: The Android Emulator installs and starts your apps faster than a real device and lets you prototype and test your app on various Android device configurations
- Code templates: Android Studio includes project and code templates that make it easy to add well-established patterns such as a navigation drawer and view pager.
- Testing tools: Android Studio provides extensive tools to help you test your Android apps with JUnit 4 and functional UI test frameworks.
- Firebase and Cloud integration: The Firebase Assistant helps you connect your app to Firebase and add services such as Analytics, Authentication, Notifications and more with step-by-step procedures right inside Android Studio. Built-in tools for Google Cloud Platform also help you integrate your Android app with services

such as Google Cloud Endpoints and project modules specially-designed for Google App Engine.

2.3.2 What is API Level?

API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.

The Android platform provides a framework API that applications can use to interact with the underlying Android system. The framework API consists of:

- A core set of packages and classes
- A set of XML elements and attributes for declaring a manifest file
- A set of XML elements and attributes for declaring and accessing resources
- A set of Intents
- A set of permissions that applications can request, as well as permission enforcements included in the system

2.3.3 Device compatibility:

The application works on all device with minimum API level of 23 i.e) Android 6.0 – Marshmallow . The current version of android is 13, but most of the devices run lesser versions, so using marshmallow will enable more than 96% of the users to run this application in their devices.

SYSTEM ANALYSIS

Chapter 3

SYSTEM ANALYSIS

System study may be defined as the process of dividing the problems into parts, identifying each part and establishing relationship in the part. System study is a detailed study of the various operations performed by a system and relationship within and outside of the system. System study is a continuing activity at all stages of the project. It is the process of studying problem to find the best solutions to the problem, by which the existing problems are understood. Objectives and requirements are defined and the solution is evaluated. Once system study is complicated, the analyst has a firm understanding of what is to be done.

System study consist of two sub phases planning and requirements definitions. They include understanding the customer's problem, performing a feasibility study, developing a recommended solution strategy, determining the acceptance criteria and planning the development process. The products of the planning are system definitions and project plan.

3.1 EXISTING SYSTEM:

To maintain fitness people often do physical exercise such as walking, swimming, using the gym, etc. But people often lack the knowledge of what they eat or what to eat. They don't have an idea of what type of nutrient is present in their food, people often follow the routines suggested by their trainers, models or the routine suggested by their friends, but this doesn't give them a clear view of what is present in the food and is hard to keep track of the foods we eat in a long term. People cannot know the amount of fat, carbohydrates, Protein they consume over a period of the time or their ratio of intake.

3.2 PROPOSED SYSTEM

The aim of project is to provide the detailed view of what the user eats. Through this project provide the facilities to the customer such as registration, search food nutrients and upload what they eat, any time on online.

3.3 FEASIBILITY STUDY:

Feasibility study is carried out when there is a complex problem or opportunity. It is considered as the primary investigation which emphasizes on “Look before You Loop” approach to any project. A Feasibility study is undertaken to determine the possibility of either improving the existing system or developing a completely new system.

We are going to develop the new system which is feasible as our application is very user friendly and easy to understand. There are aspects in feasibility study portion of the preliminary investigation:

1. Technical Feasibility
2. Economic Feasibility
3. Operation Feasibility

3.3.1 TECHNICAL FEASIBILITY:

Technical feasibility is a type of study the current technology in used in an organization is checked such as the existing software, hardware, and personnel staff to determine whether it will work for the proposed system or completely new ones is to be used. The technology that was important in developing a new system such as Development tools, back-end database system was available from within the organization. The proposed system is capable of adding, changing, enhancing functionality, features etc. The proposed system is capable of handling large storage of data. The back-end and front-end technology has greater importance for providing an accurate, error-free, frequencies of data to be used.

- Scalability and Extensibility
- Flexibility
- Robustness
- Easy to debug and maintain
- Must provide the excellent reporting features with good printing support

- Event driven programming facility

3.3.2 ECONOMIC FEASIBILITY:

For proving that system developed is economical, the economic feasibility study takes place to check the cost of developing a system against the benefits that it provides. If the cost is less and benefits are more than we can define our system to be economically developed. User saves time in searching for a particular product to be purchased by simply few clicks. The registration process is speedier than the registered manually. The saving of papers as all data are stored computerized.

The record is of free of human errors as there is less chance of mistakes. The above benefits are in terms of saving time, minimize errors and provide efficiency in work done. In terms of economic feasibility our application is very reasonable in cost. So, application is economically feasible.

This is a personal application which does not require any economical investments.

3.3.3 OPERATION FEASIBILITY

The operational feasibility is concerned with the operability of the system after it has been installed. That is, some programmer may not like changes in their routine method of work or has fear that they will lose their peer group. The following areas will have the operational feasibility in the proposed project

- The organization has approved this system as their working system.
- The User of the system has accepted the proposed system as their new working system and realized the benefits of it.
- The system will work in a proper way after it has been installed and the installation process is easy to use.

SYSTEM DESIGN AND DEVELOPMENT

Chapter 4
SYSTEM DESIGN AND DEVELOPMENT

4.1 ELEMENT OF DESIGN:

System development can be generally thought of as having two major components: System analysis and system design is the process of planning a new business system or one to replace or complement an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used to make its operation more effective.

➢ UML Design:

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing and documenting the software system and its components. It is graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain and control information about the systems.

The UML language is for;

- ➢ Visualizing
- ➢ Specifying
- ➢ Constructing
- ➢ Documenting

➢ Visualizing:

Through UML we see or visualize an existing system and ultimately, we visualize how the system is going to be after implementation. Unless we think, we cannot implement.

UML helps to visualize how the components of the system communicate and interact with each other.

- Specifying:

Specifying means building models that are precise, unambiguous and complete UML address the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

- Constructing:

UML models can be directly connected to a variety of programming languages through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward engineering and reverse engineering is possible through UML.

- Documenting:

The deliverables of a project apart from coding are some artifacts, which are critical in controlling measuring and communicating about during its developing requirements, architecture, desire, source code, project plans, tests, prototypes, releases, etc.

4.2 UML APPROACH

- UML Diagram

A diagram is the graphical representation of a set of elements, most often rendered as a connected graph of vertices and arcs. You draw a diagram to visualize a system from a different perspective, so diagram is a projection into systems. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams, or in no diagrams at all. In theory, a diagram may contain any combination arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

- Use Case Diagram
- Data Flow Diagram

4.2.1 USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), any dependencies between those use cases.

- Use case diagram for Registering and Login

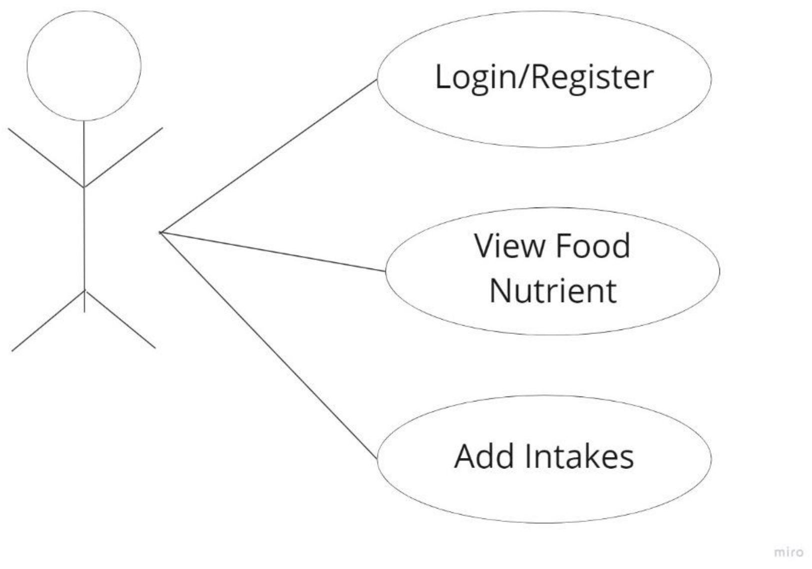


Fig .10 Use Case diagram

4.2.2 DATA FLOW DIAGRAM

The DFD was first developed by Larry Constantine as a way of explaining system requirements in graphical form. A DFD also known as a bubble chart has the purpose of clarifying system requirements and identifying major transformation that will become programs in system design. This is a starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail.

4.2.2.1 USER SIGN IN ACTIVITY

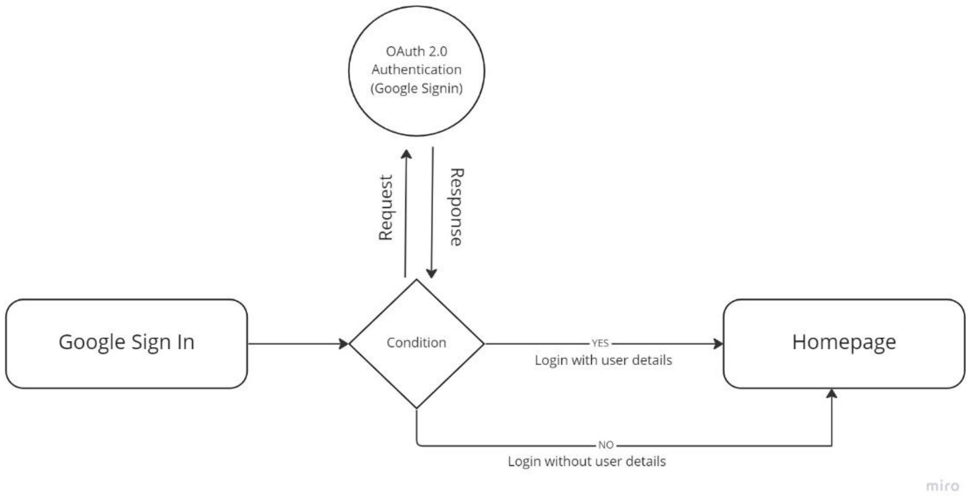


Fig .9 User Sign in Activity

4.2.2.2 FIRESTORE DATABASE ACTIVITY

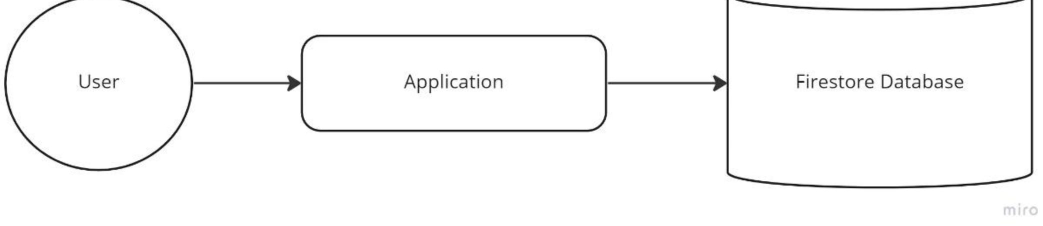


Fig .11 Firestore Database Activity

APPLICATION DEVELOPMENT ENVIRONMENT

Chapter 5
APPLICATION DEVELOPMENT ENVIRONMENT

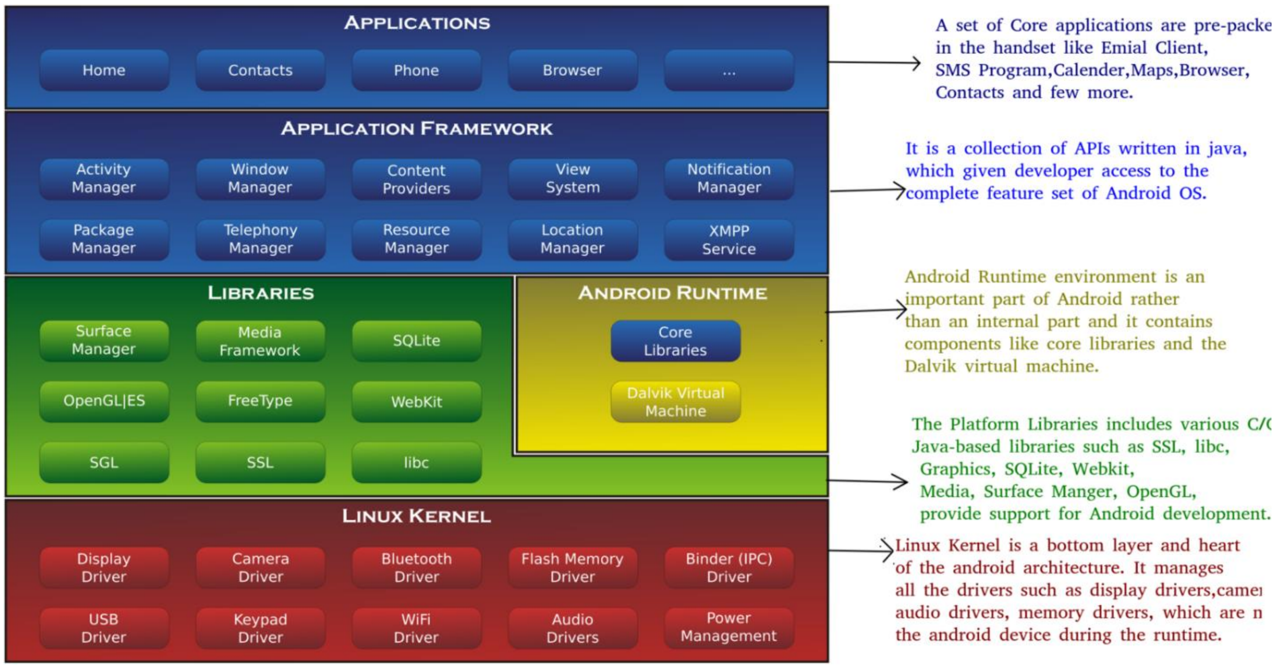
This chapter gives a detailed view of what the development environment is composed of and how it is used. We can also know the installation of IDE, life cycle of an Android, and Android architecture.

5.1 Overview

Android applications, like most mobile phone applications, are developed in a host target development environment. In other words, one can develop an application on a host computer (where resources are abundant) and download it to a target mobile phone for testing and ultimate use. Applications can be tested and debugged either on a real Android device or on an emulator. For most developers, using the emulator is easier for initial development and debugging, followed by final testing on real devices. Testing an application on the emulator is almost the same as testing it on a real device, except for some certain features that are limited on the emulator, such as video, which cannot be viewed properly like the real device, location-based application, which will require a GPX or KML file to simulate the movement of the device, and some few others. In developing an Android application, the required tools will be used to set up an appropriate development environment on a computer. Linux, Windows and OS X support the development environment. This project used Windows 11 as the development platform. The Android SDK supports several integrated development environments (IDEs), but for the purpose of this project, Android Studio was considered, due to the fact that it is best integrated with the Android SDK and it is free. Android Studio, Sun's Java Development Kit (JDK), Android Software Developer's Kit (SDK) and Android Development Tool (ADT) are what is needed to set up the development environment.

5.2 Android Architecture

Android is ship with a set of core applications including an email client, SMS program, calendar, maps, browser, and contacts. The applications are written using the Java programming language. By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, among others. Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user. Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.



Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
XMPP Service
Libraries
Surface Manager
Media Framework
SQLite
OpenGL|ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Bluetooth Driver
Flash Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management
A set of Core applications are pre-packe in the handset like Emial Client, SMS Program,Calender,Maps,Browser, Contacts and few more.
It is a collection of APIs written in java, which given developer access to the complete feature set of Android OS.
Android Runtime environment is an important part of Android rather than an internal part and it contains components like core libraries and the Dalvik virtual machine.
The Platform Libraries includes various C/(Java-based libraries such as SSL, libc, Graphics, SQLite, Webkit, Media, Surface Manger, OpenGL, provide support for Android development.
Linux Kernel is a bottom layer and heart of the android architecture. It manages all the drivers such as display drivers,camei audio drivers, memory drivers, which are n the android device during the runtime.

Fig 12 Android Architecture

Figure 1, shows Android architecture from the Kernel level to the application level. This also includes a set of core libraries that provides most of the functionalities available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been designed so that multiple instances can efficiently run on a single device. It executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

Dalvik Virtual Machine

Dalvik is the name of Android’s virtual machine. It was named by Bornstein after the fishing village of Dalvik in Eyjafjörður (Iceland), where some of his ancestors lived. As explained in section 2.3, it is an interpreter-only virtual machine that executes files in the Dalvik Executable (.dex) format, a format that is optimized for efficient storage and memory-map able execution. The virtual machine is register-based, and it can run classes compiled by a Java language compiler that have been transformed into its native format using the included "dx" tool. The VM runs on top of a Linux 2.6 kernel, which it relies on for underlying functionality (such as threading and low-level memory management). The DalvikVM was also optimized to be running in multiple instances with a very low memory-footprint. Several VMs protect one’s application from being dragged down by another crashed Application. Unlike Java virtual machine (Stack-based Virtual Machine) found in other desktop computers, DalvikVM is register-based virtual machine which suits well for mobile processors. Besides, registered-based VM provides faster execution time

5.3 Android Application Life cycle

An application life cycle consists of the steps that the processes of the application must follow from execution to termination. Every application, regardless of the language it was written in, has a specific life cycle, and Android applications are no exception. All of the running processes are watched by Android and, depending on how the activity is running (that is, a foreground activity, background activity, and so forth), Android may choose to end the activity to reclaim needed resources. The cycle starts when an Android application is created, the processes are started, events are fired, processes are stopped, and the application is destroyed.

Unlike most traditional environments, Android applications have no control over their own life cycles. Instead, application components must listen to changes in the application state and react accordingly, taking particular care of being prepared for untimely termination. The Android application runs in its own process, this implies that it runs a separate instance of Dalvik. Memory and process management of each application is handled exclusively by the runtime. Android aggressively manages its resources, ensuring that the device remains responsive. This means that processes (and their host applications) will be killed, without warning, if necessary, to free resources for higher-priority applications, generally those that are interacting directly with the user at the time.

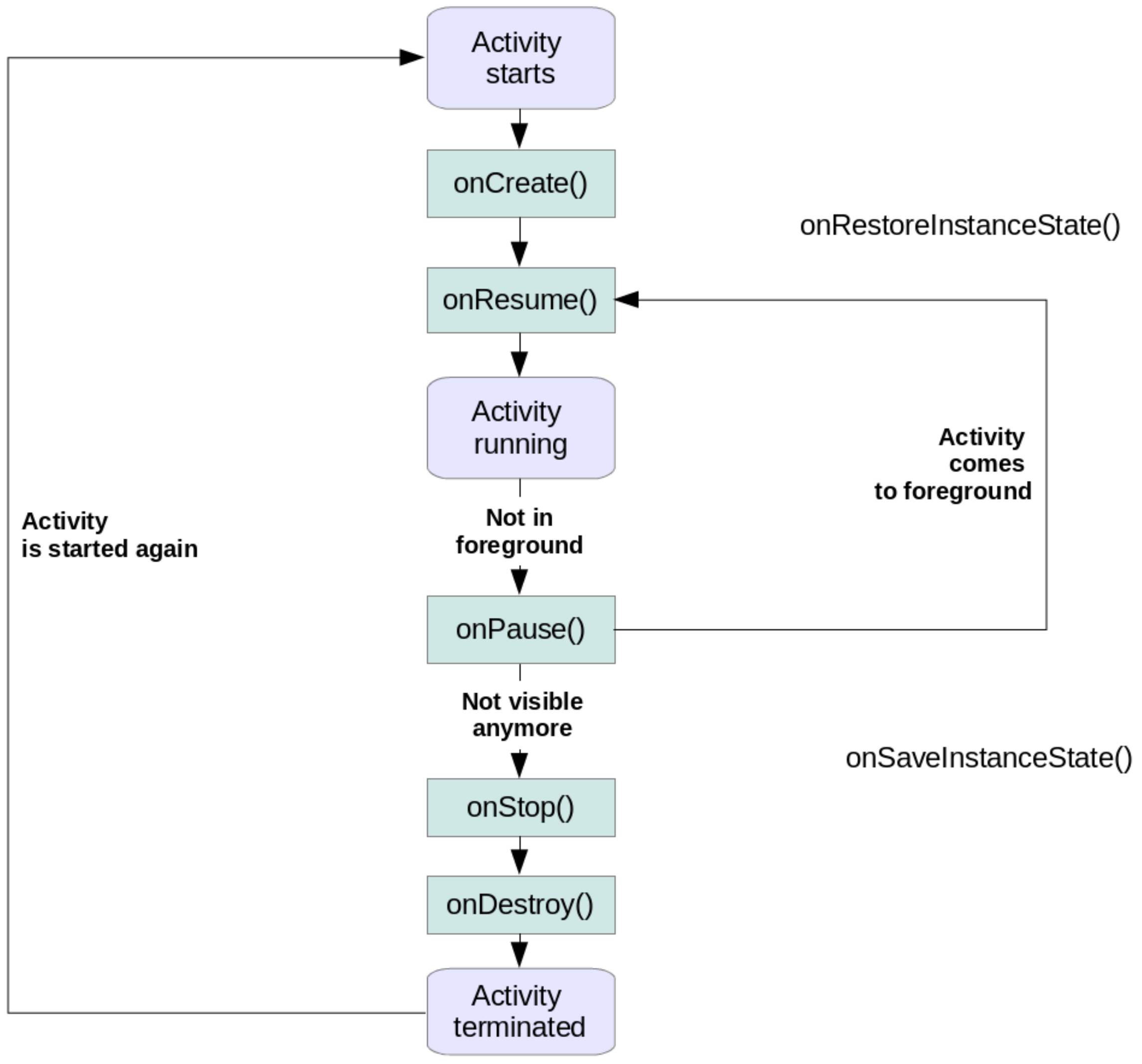


Fig.2 ***Android Application Life cycle***

- **onCreate()**

It is called when the activity is first created. This is where all the static work is done like creating views, binding data to lists, etc. This method also provides a Bundle containing its previous frozen state, if there was one.

- **onStart()**

It is invoked when the activity is visible to the user. It is followed by onResume() if the activity is invoked from the background. It is also invoked after onCreate() when the activity is first started.

- **onRestart()**

It is invoked after the activity has been stopped and prior to its starting stage and thus is always followed by onStart() when any activity is revived from background to on-screen.

- **onResume()**

It is invoked when the activity starts interacting with the user. At this point, the activity is at the top of the activity stack, with a user interacting with it. Always followed by onPause() when the activity goes into the background or is closed by the user.

- **onPause()**

It is invoked when an activity is going into the background but has not yet been killed. It is a counterpart to onResume(). When an activity is launched in front of another activity, this call back will be invoked on the top activity (currently on screen). The activity, under the active activity, will not be created until the active activity's onPause() returns, so it is recommended that heavy processing should not be done in this part.

- **onStop()**

It is invoked when the activity is not visible to the user. It is followed by **onRestart()** when the activity is revoked from the background, followed by onDestroy() when the activity is closed or finished, and nothing when the activity remains on the background only. Note that this method may never be called, in low m55emory situations where the system does not have enough memory to keep the activity's process running after its onPause() method is called.

- **onDestroy()**

The final call received before the activity is destroyed. This can happen either because the activity is finishing (when finish() is invoked) or because the system is temporarily destroying this instance of the activity to save space. To distinguish between these scenarios, check it with is **Finishing()** method.

5.4 Android Application Framework

Android provides an open development platform that gives developers the opportunity to take advantage of for example the device hardware, access location information, run background services, add notification to the status bar, among others. Since full access to the framework APIs is given to developers and the application architecture is designed to simplify the reusability of the components, any application can publish its capabilities. Any other application may then make use of those capabilities (subject to security constraints enforced by the framework).

5.4.1 Activity Manager

Activity Manager is a project management tool that is simple to use, lightweight, very efficient and customizable. It features collaborating repository administration, tasks repository administration, contributions management (activity management), and an extensible report facility (with built-in templates). It allows one to build and maintain a hierarchical task tree. In Android the activity manager manages the life cycle of the application. This is managed as an activity stack when a new activity is started, it is kept on top of the stack and it becomes the running activity. The essential states are specified as follows:

- If an activity is on the foreground of the screen (at the top of the stack), it is active or running.
- If an activity has lost focus but is still visible (that is, a new non-full-sized or transparent activity has focus on top of your activity), it is paused. A paused activity is completely alive (it maintains all state and member information and remains

attached to the window manager), but can be killed by the system if the system is in extreme low memory situations.

- If an activity is completely obscured by another activity, it is stopped. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere.
- If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish, or simply killing its process. When it is displayed again to the user, it must be completely restarted and restored to its previous state.

5.4.2 Content Provider

In the Android application, data can be stored in files, in the SQLite database or in another mechanism. Classes that implement a standard set of methods letting other applications store and retrieve the type of data they handle are known as content providers. This is useful when there is need for an application to share data with other applications. In view of providing an avenue for data to be shared across an application, Android ship a number of contents provides for common data types such as audio, video, images, and personal contact information. They can be queried for the content they hold or they can be modified, and it is also possible to create once own content provider. In any case some content providers demand that proper permission be taken before they can be accessed.

5.4.3 Resource Manager

The resources required in an application development are handled by the Resource Manager. The Application resources are stored under the “res” folder of the project hierarchy. In this folder, each of the available resource types can have a subfolder containing its resources. There are seven primary resource types that have different folders:

- Simple values
- Drawable
- Layouts
- Animations

- XML
- Styles
- Raw resources

When the application is built, these resources will be compiled as efficiently as possible and included in the application package. This process also creates an R-class file that contains references to each of the resources that are included in the project. This helps for an easy reference to the resources in the code, with the advantage of design time syntax checking.

NETWORK AND NUTRITIONAL API

Chapter 6

Network and Nutritional API

In this Chapter we will see in detail about how an web API works and how it is used in our Android application. We will also know about the network enabling services in android and the required libraries to implement them in our application.

6.1 Network

In Android network must be enabled by the developer at the time of app development, Network is important to our application to use API services. Most network-connected apps use HTTP to send and receive data. The Android platform includes the HttpsURLConnection client, which supports TLS, streaming uploads and downloads, configurable timeouts, IPv6, and connection pooling. In this topic, we use the Volley HTTP client library, which lets you create an HTTP client declaratively.

To enable Networking add,

In **AndroidManifest.xml** add the internet permission:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

6.2 Web API

A web API is an application programming interface for either a web server or a web browser. It is a web development concept, usually limited to a web application's client-side (including any web frameworks being used), and thus usually does not include web server or browser implementation details such as SAPIs or APIs unless publicly accessible by a remote web application.

6.3 Nutritional API

Edamam Recipe Search API (B2B API), is accessed by sending HTTPS requests on specific URLs as described below. The base URL is https://api.edamam.com, and you obtain the full URL by appending request's path to the base URL for example, https://api.edamam.com/api/recipes/V2. This API provides vast information's about various food and their nutritional values, and also has various tags and labels such as Nutrient tags, health labels, diet labels, meal types etc. The edamam server's support standard HTTP compression using gzip. Using compression can reduce the size of the response and thus, increase the transfer speed. It is an open API.

6.4 Dependencies to use API

6.4.1 Using VOLLEY Library:

Volley is an **HTTP library** that makes networking very easy and fast, for Android apps. It was developed by Google and introduced during Google I/O 2013. It was developed because there is an absence in Android SDK, of a networking class capable of working without interfering with the user experience. Although Volley is a part of the Android Open-Source Project (AOSP), Google announced in January 2017 that Volley will move to a standalone library. It manages the processing and caching of network requests and it saves developers valuable time from writing the same network call/cache code again and again. In our application volley is used to communicate with the web API, to get the results through a secured connection from the internet. Results are obtained in the form of JSON file format.

6.4.1.1 Implementing VOLLEY:

Before getting started with Volley, we need to import Volley and add permissions in the Android Project.

Open build.gradle(Module: app) **and add the following dependency:**

```
dependencies {

//...

implementation 'com.android.volley:volley:1.0.0'

}
```

6.4.1.2 Advantages of Using Volley:

- All the tasks that need to be done with Networking in Android, can be done with the help of Volley.
- Automatic scheduling of network requests.
- Catching
- Multiple concurrent network connections.
- Cancelling request API.
- Request prioritization.
- Volley provides debugging and tracing tools.

6.4.2 Using GLIDE library:

Glide is a fast and efficient image loading library for Android focused on smooth scrolling. Glide offers an easy-to-use API, a performant and extensible resource decoding pipeline and automatic resource pooling. Glide supports fetching, decoding, and displaying video stills, images, and animated GIFs. Glide includes a flexible API that allows developers to plug in to almost any network stack. By default, Glide uses a custom HttpUrlConnection based stack, but also includes utility libraries plug in to Google's Volley project.

Glide's primary focus is on making scrolling any kind of a list of images as smooth and fast as possible, but Glide is also effective for almost any case where you need to fetch, resize, and display a remote image.

We use Glide in our application to mainly display the images obtained JSON objects from the API to the UI for users to see. The informations are collected to Array list and are set to recyclerview according to the length of the array.

6.4.2.1 Implementing GLIDE:

Before getting started with Glide, we need to import Glide and add permissions in the Android Project.

Open **build.gradle(Module: app)** and add the following dependency:

```
dependencies {

//…

implementation ‘com.bumptech.glide: glide:4.14.2’

}
```

SIGN IN AUTHENTICATION AND STORAGE

Chapter 7

SIGIN AUTHENTICATION AND STORAGE

In this chapter, we will see in detail about implementing and connecting firebase in our android application and accessing firestore for storing user data's required for the application. Also, to use google API for registering new users and signing in authentications.

7.1 Firebase

Firebase is a set of hosting services for any type of application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). It offers NoSQL and real-time hosting of databases, content, social authentication (Google, Facebook, Twitter and Github), and notifications, or services, such as a real-time communication server. Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

In 2014, Firebase launched two products: Firebase Hosting and Firebase Authentication. In October 2014, Firebase was acquired by Google. A year later, in October 2015, Google acquired Divshot, an HTML5 web-hosting platform, to merge it with the Firebase team. In October 2017, Firebase launched Cloud Firestore, a real-time document database as the successor product to the original Firebase Realtime Database. Firebase is considered as web application platform. It stores the data in JavaScript Object Notation (JSON) format which doesn't use query for inserting, updating, deleting or adding data to it. It is the backend of a system that is used as a database for storing data.

In our application we use fireStore as our primary database to store all data collected from the user, we follow the basic read and write functions in our project.

- **Firebase analytics**

It provides insight into app usage. It is a tool that helps you to understand how Android and iOS users are engaging with your application. Firebase is Google’s mobile based application development platform that helps you develop, improve and grow your app.

➢ Firebase Realtime Database

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in Realtime. Realtime syncing makes it easy for your users to access their data from any device: web or mobile, and it helps your users collaborate with one another. When your users go offline, the Realtime database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronised.

➢ Firebase Authentication

Firebase Auth supports social login provider like Google, Facebook, GitHub and Twitter. It is a service that can authenticate users using only client-side code. It also includes a user management system where by developers can enable user authentication with email and password login stored with firebase. Firebase Authentication also handles password reset emails.

➢ Firebase Storage

It facilitates easy and secure file transfer regardless of network quality for the Firebase apps. It is backed by google cloud storage which is cost-effective object storage service. The developer can use it to store images, audio, video, or other user generated content.

➢ Firebase Crash Reporting

The detailed reports of the errors are created in the app. The errors are grouped into clusters of similar stack traces and triaged by the severity. The other features are:

The developer can log custom events to help capture the steps leading up to a crash.

7.2 Implementing Firebase

Android studio IDE supports Firebase implementation as an inbuilt Firebase assistant. Google developed a platform called Firebase that provide all these online services. It also gives a daily analysis of usage of these services along with the details of user using it.

To simplify, it can be said that Firebase is a mobile and web application development platform. It provides services that a web application or mobile application might require. Anyone can easily include firebase to there application and it will make their online work way easier than it was used to be.

7.2.1 Manually adding firebase

Implementing firebase with firebase assistant involves a sequel of steps to be followed, they are

i. Open your Browser and open Firebase Console.

ii. Create a **new project** in the firebase by clicking **create project** in the firebase console.

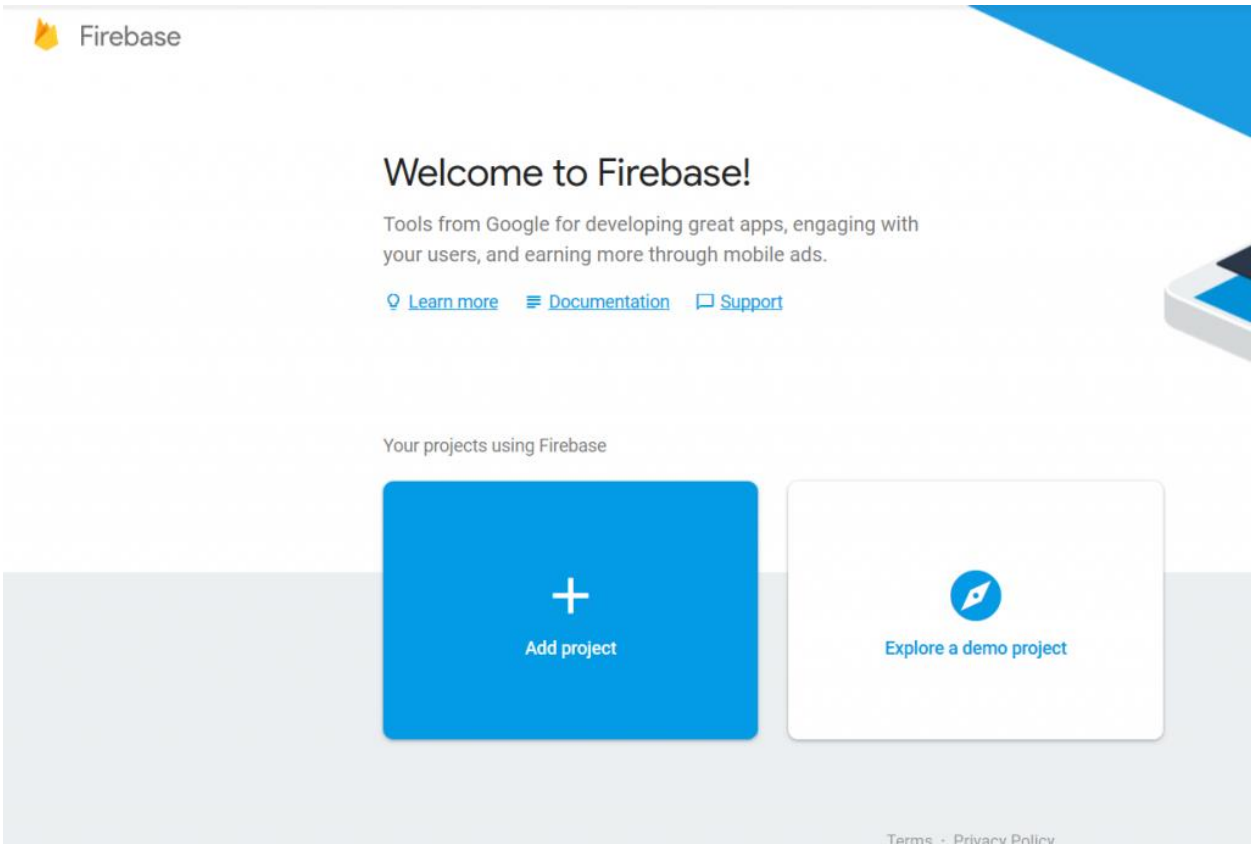
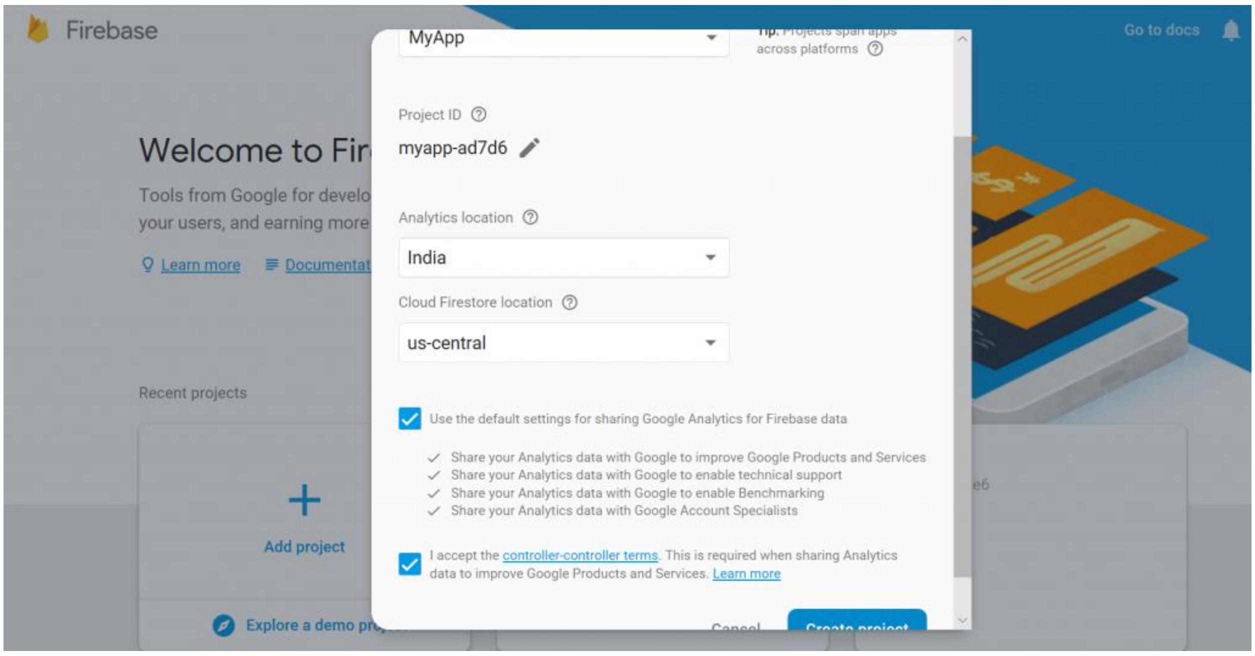

Firebase
Welcome to Firebase!
Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads.
Learn more
Documentation
Support
Your projects using Firebase
Add project
Explore a demo project
Terms · Privacy Policy

Fig .13 Firebase welcome screen

- Fill the necessary details in the pop-up window about the project. Edit the project ID if required.



MyApp

Project ID

myapp-ad7d6

Analytics location

India

Cloud Firestore location

us-central

☑ Use the default settings for sharing Google Analytics for Firebase data

- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable technical support
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google Account Specialists

☑ I accept the controller-controller terms. This is required when sharing Analytics data to improve Google Products and Services. Learn more

Fig .14 Firebase project creation

- Click on create project to finally create it.

iii. Now add this project to the android app

- Click on the **Add firebase to your android app** option on the starting window.

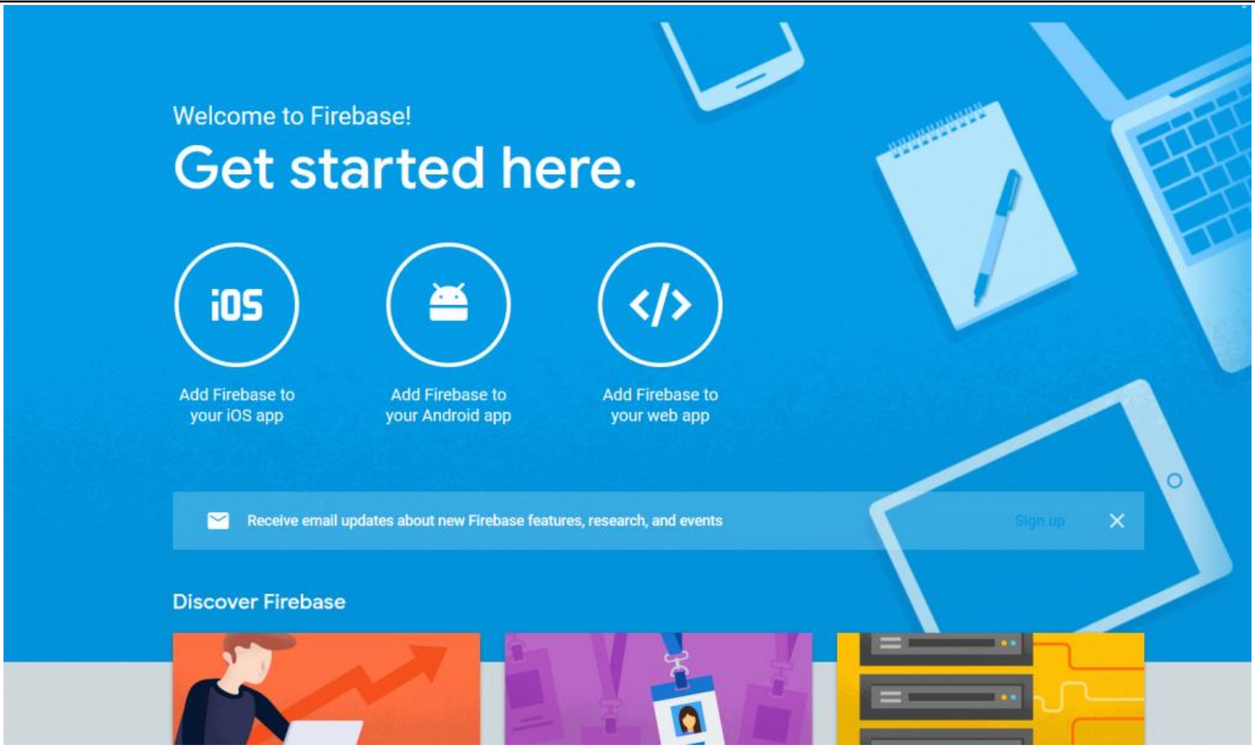
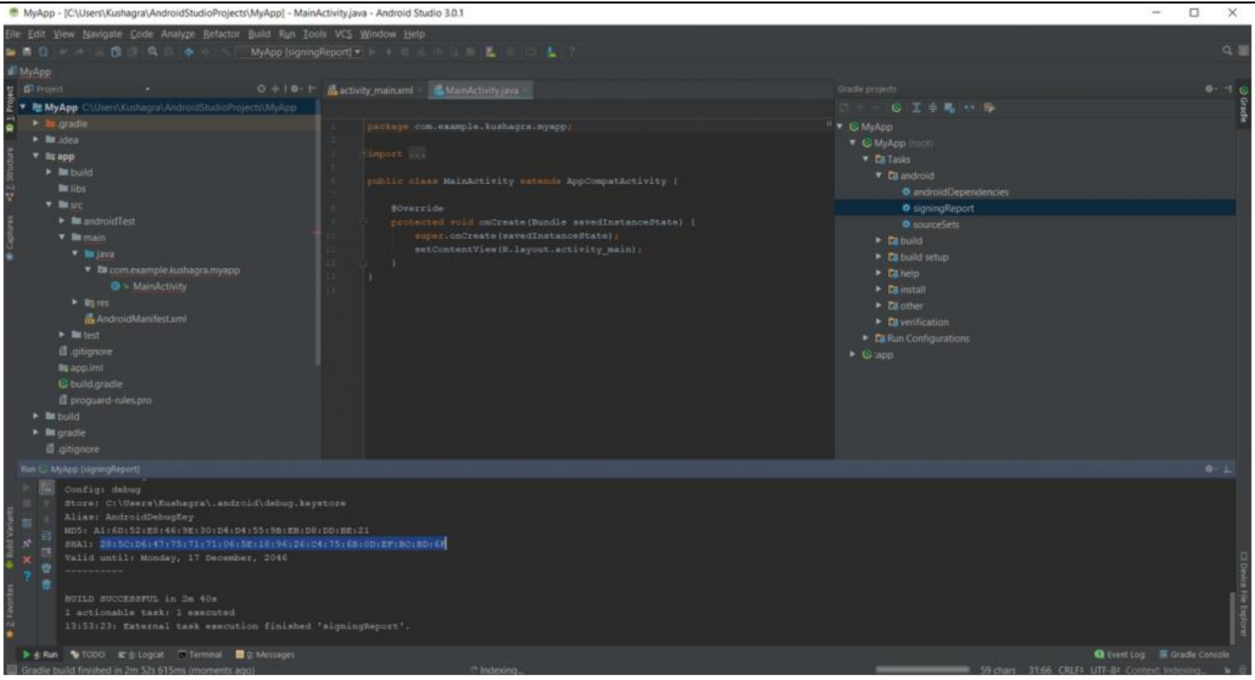

Welcome to Firebase!
Get started here.
iOS
Add Firebase to your iOS app
Add Firebase to your Android app
</>
Add Firebase to your web app
Receive email updates about new Firebase features, research, and events
Sign up
Discover Firebase

Fig .15 Choosing preferred application

- A prompt will open where to enter the package name of the app.
- Now the app is connected to the Firebase. Now all the cloud based as well server based services can be easily used in the app.
- Now the app will be registered with firebase.

iv. Also, the SHA1 certificate, can be given, of the app by following steps:

```
Go to android studio project
 ↳ gradle
   ↳ root folder
     ↳ Tasks
       ↳ Android
         ↳ signingReport
           ↳ copy paste SHA1 from console
```



```
package com.example.kushagra.myapp;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

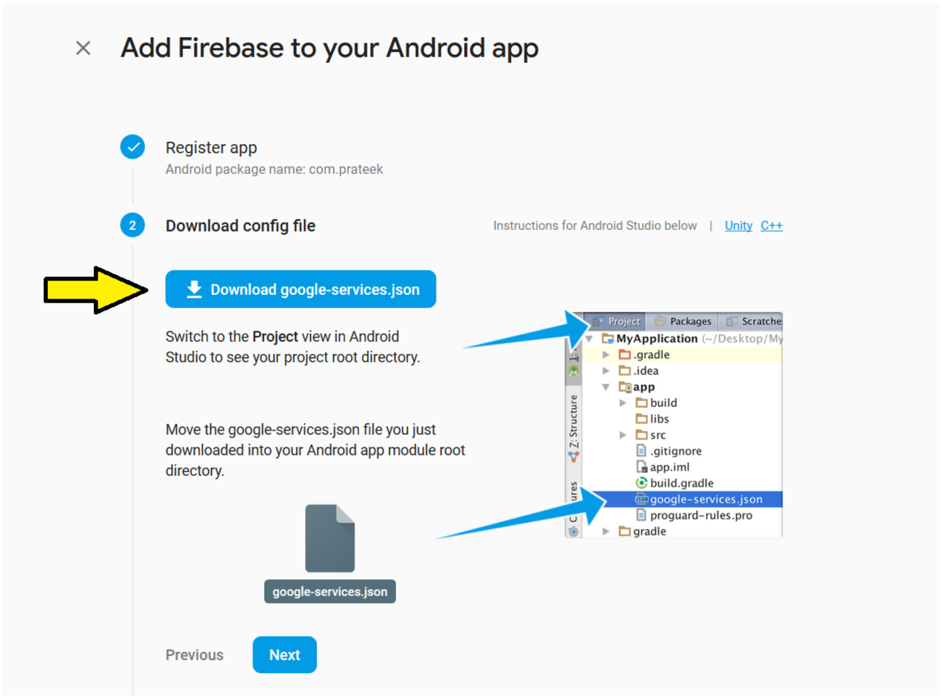
```
Config: debug
Store: C:\Users\Kushagra\.android\debug.keystore
Alias: AndroidDebugKey
MD5: A1:6D:52:E8:46:9E:30:D4:D4:55:9B:EB:D8:DD:BE:21
SHA1: 28:5C:D6:47:75:71:71:06:5E:18:96:26:C4:75:6B:0D:EF:BC:BD:6E
Valid until: Monday, 17 December, 2046
----------

BUILD SUCCESSFUL in 2m 40s
1 actionable task: 1 executed
13:53:23: External task execution finished 'signingReport'.
```

Gradle build finished in 2m 52s 615ms (moments ago)

Fig .16 SHA1 certificate screen

v. Now download the **google-services.json** file and place it in the root directory of the android app.



Add Firebase to your Android app

Register app
Android package name: com.prateek

2 Download config file — Instructions for Android Studio below | Unity C++

Download google-services.json

Switch to the **Project** view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.

Fig .17 Adding google-services.json file

vi. Now add the following in the project.

- Adding the SDK in the project.

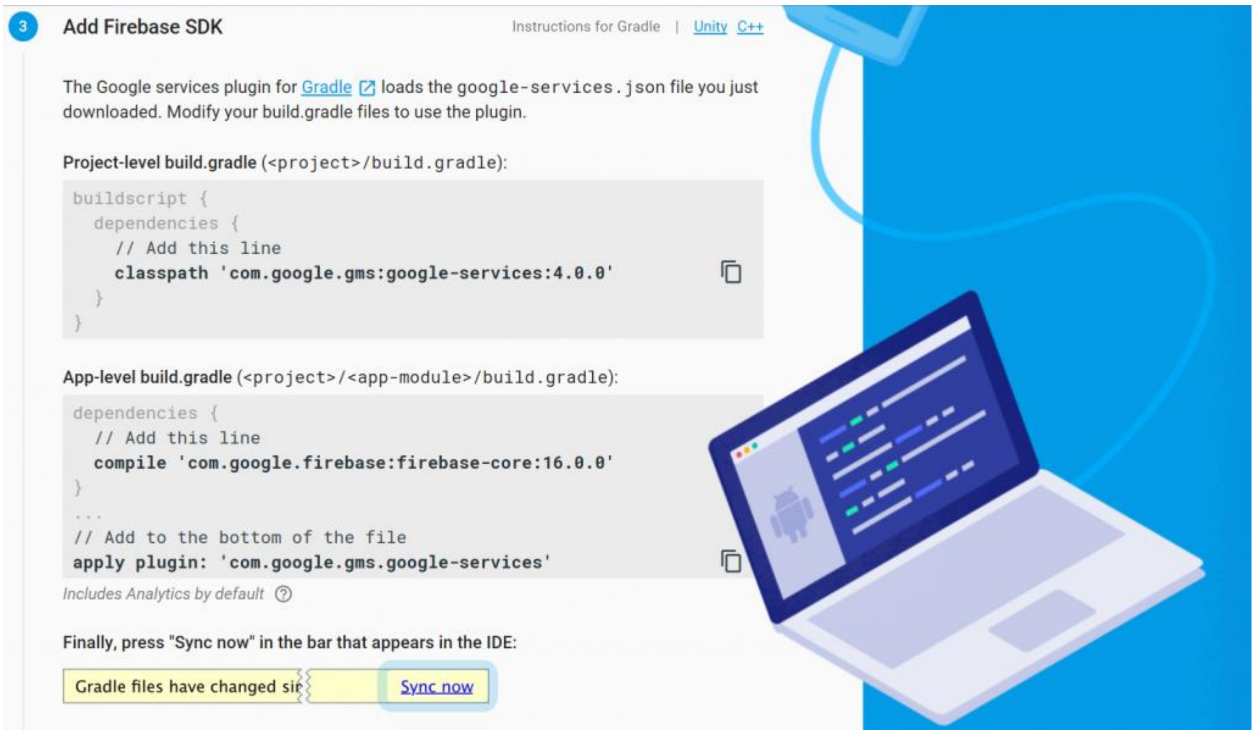
Add the following code to the PROJECT-LEVEL*build.gradle* of the app.

```
buildscript {
```

```
dependencies {
    classpath 'com.google.gms:google-services:4.0.0'
    }
    }
```

- Add the following code to APP-LEVEL *build.gradle* of the app.

```
dependencies {
compile 'com.google.firebase:firebase-core:16.0.0'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```



3 Add Firebase SDK

Instructions for Gradle | Unity C++

The Google services plugin for Gradle loads the google-services.json file you just downloaded. Modify your build.gradle files to use the plugin.

Project-level build.gradle (<project>/build.gradle):

```
buildscript {
  dependencies {
    // Add this line
    classpath 'com.google.gms:google-services:4.0.0'
  }
}
```

App-level build.gradle (<project>/<app-module>/build.gradle):

```
dependencies {
  // Add this line
  compile 'com.google.firebase:firebase-core:16.0.0'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

Includes Analytics by default

Finally, press "Sync now" in the bar that appears in the IDE:

Fig .18 Adding firebase SDK to gradle

vii. Now **Sync** the gradle by clicking on sync now.

viii. After adding the above code (SDK), run the app to send the verification to the Firebase console.

Firebase is now successfully installed.

7.3 Signing In using OAuth 2.0 Authentication (Google Sign)

The Google authentication provider allows users to log in with their existing Google account through Google Sign-In. When a user logs in, Google provides Atlas App Services with an OAuth 2.0 access token for the user. App Services uses the token to identify the user and access approved data from Google APIs on their behalf.

In our application we use google sign in method for users to have tension free signing option, so that they could sign in with a single tap. This eliminates user to keep in mind of their user ID or password. The sign in is authenticated by firebase authentication.

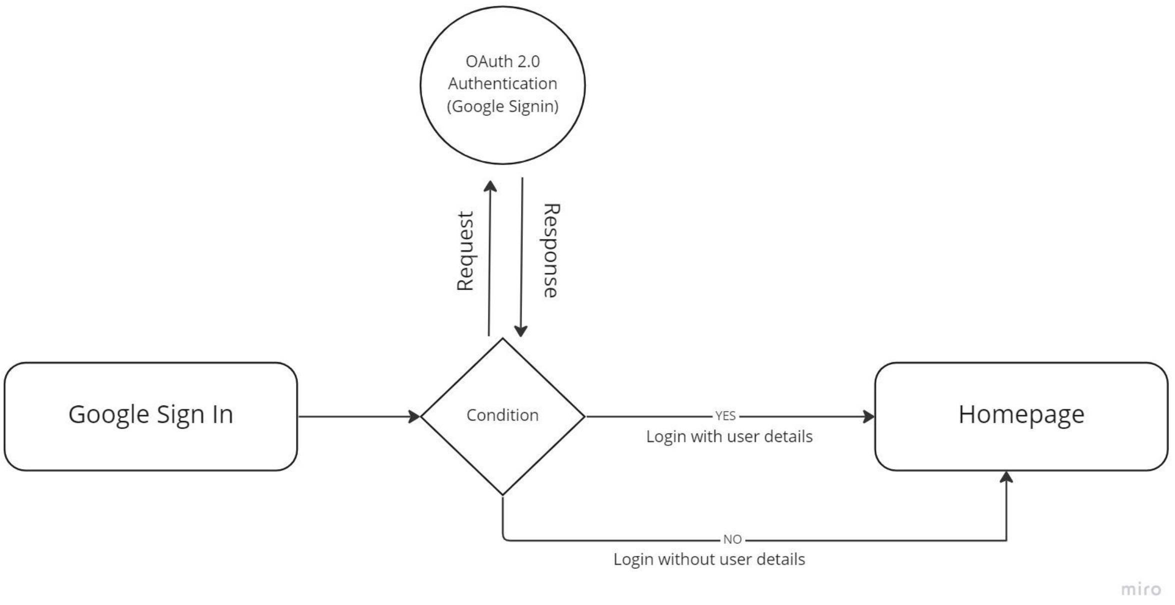


Fig .19 User Sign in Activity

7.4 Firestore database

We have two options with Firebase, i.e., Firebase Real-time Database, which we learned in our previous section and Cloud Firestore. Cloud Firestore is newer, but it is not replacing the Firebase Real-time Database. Cloud Firestore is a flexible as well as scalable NoSQL cloud database. It is used to store and sync data for client and server-side development. It is used for mobile, web, and server development from Google Cloud

Platform and Firebase. Like the Firebase Real-time Database, it keeps syncing our data via real-time listeners to the client app. It provides offline support for mobile and web so we can create responsive apps that work regardless of network latency or Internet connectivity.

Cloud Firestore also provides seamless integration with Google Cloud Platform products and other Firebase, including cloud functions.

7.4.1 Firestore Database Activity

Firestore is quite similar to Firebase Realtime Database. Firestore also uses data synchronization for updating data on any connected device. It is designed for making simple one-time fetch queries efficiently.

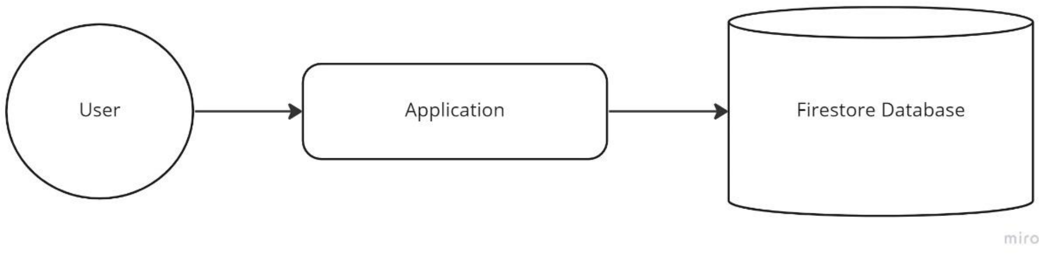


Fig .20 Firestore Database Activity

SYSTEM TESTING

Chapter 8
SYSTEM TESTING

INTRODUCTION:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is a process of exercising the software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each addresses a specific testing requirement.

1. The system should be tested to ensure that it meets the requirements for design and development.

2. The system will be tested to ensure that it responds correctly to all types of inputs.

3. Perform its functions accurately and within a reasonable time frame.

4. Installed and used in the intended environments.

5. Obtain the overall desired and pre-defined results**.**

8.1 SYSTEM FUNCTIONALITY TESTING

System functionality testing entails testing the system's primary function. Testing is normally achieved by user interface-initiated test flows. Not just the flow of a use case is tested, but the various business rules are also tested. Testing is done by certifying the requirements. i.e., whether the application is working based on the requirements.

8.2 UI TESTING

This is a user-centric testing of the application. In this test phase, items such as visibility of text in various screens of the app, interactive messages, alignment of data, the look and feel of the app for different screens, size of fields etc are tested under this.

• Verify whether the scrolling of the list is glitch free and the cards are not shown misaligned when a long list of cards is scrolled etc.

• Verify whether the cards are shown properly aligned with the mobile screen size.

SCREEN LAYOUTS

Chapter 9

SCREEN LAYOUTS

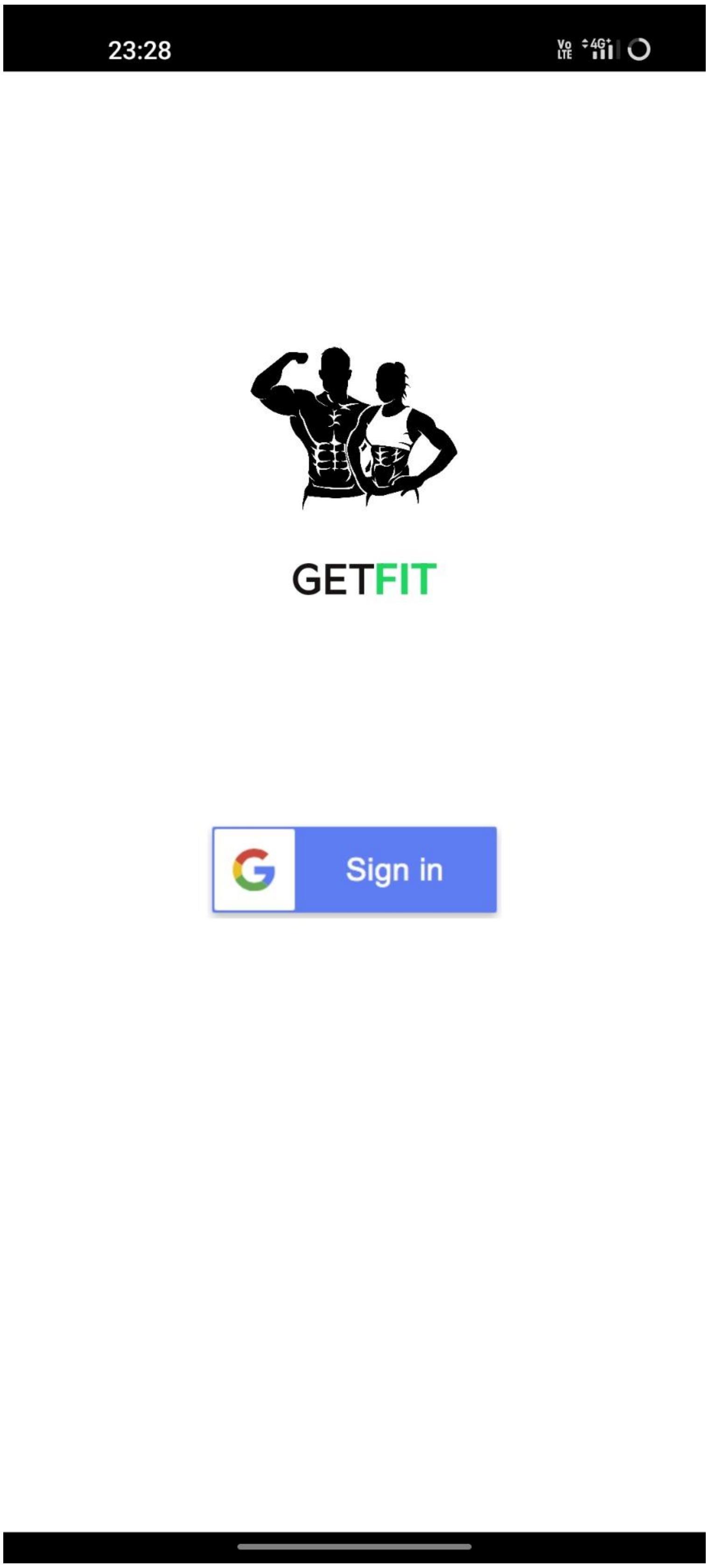


Fig .21 Register Activity

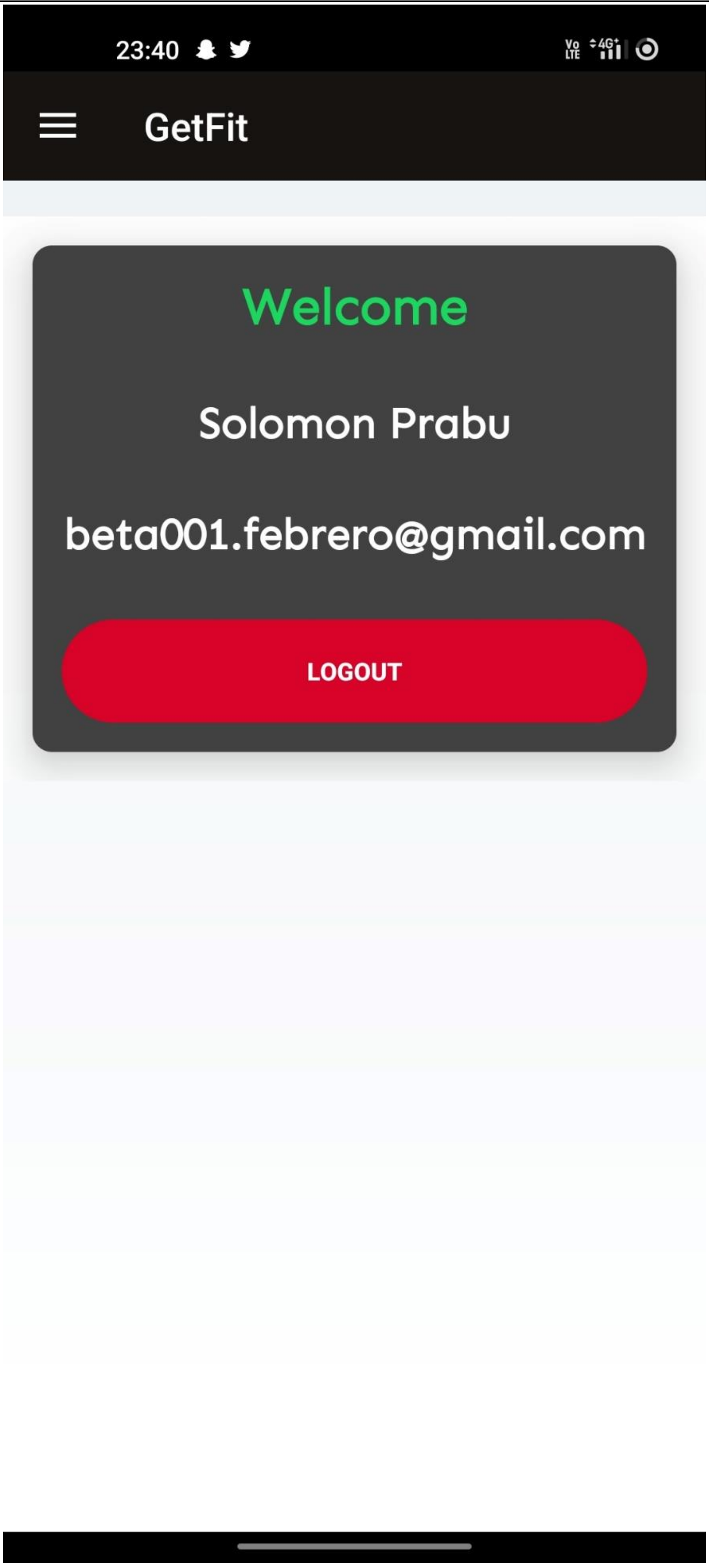
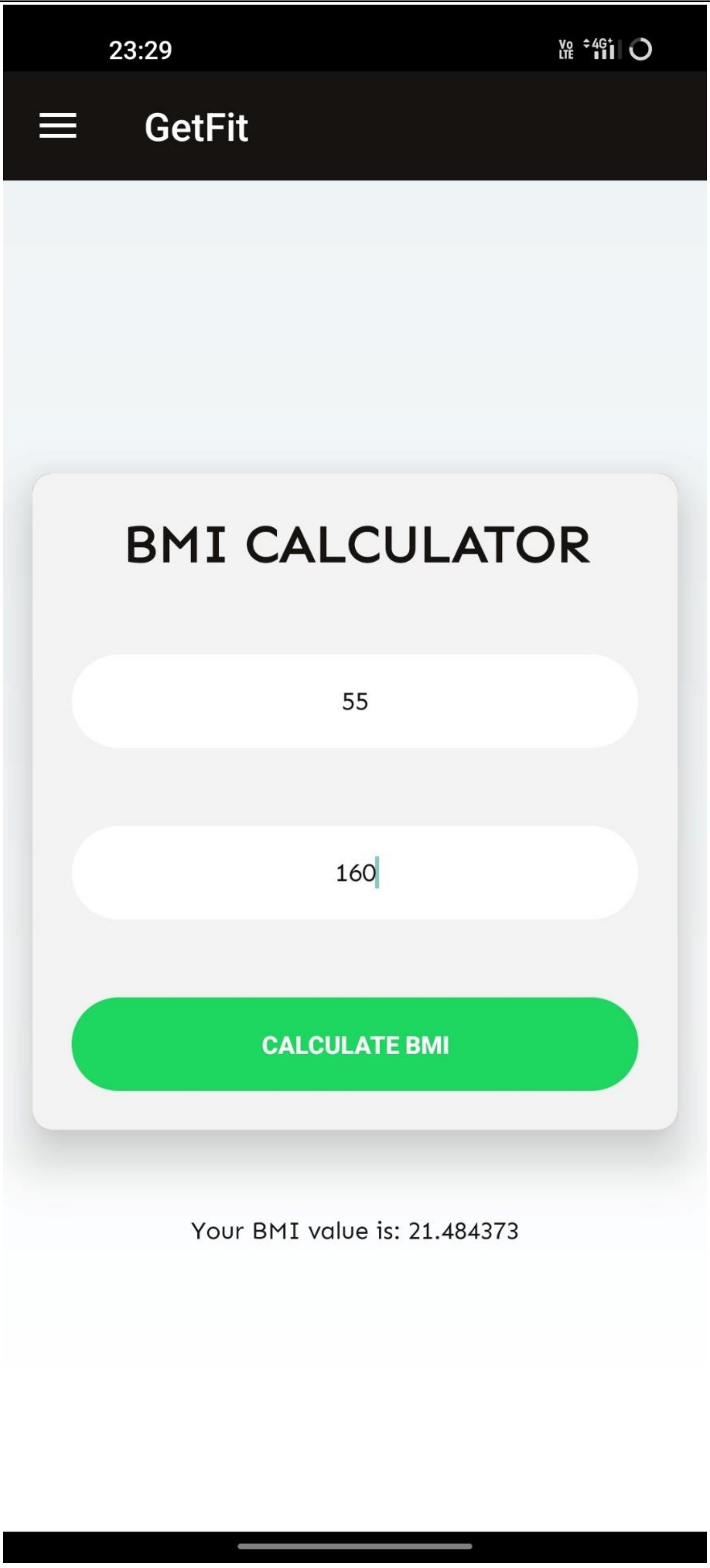
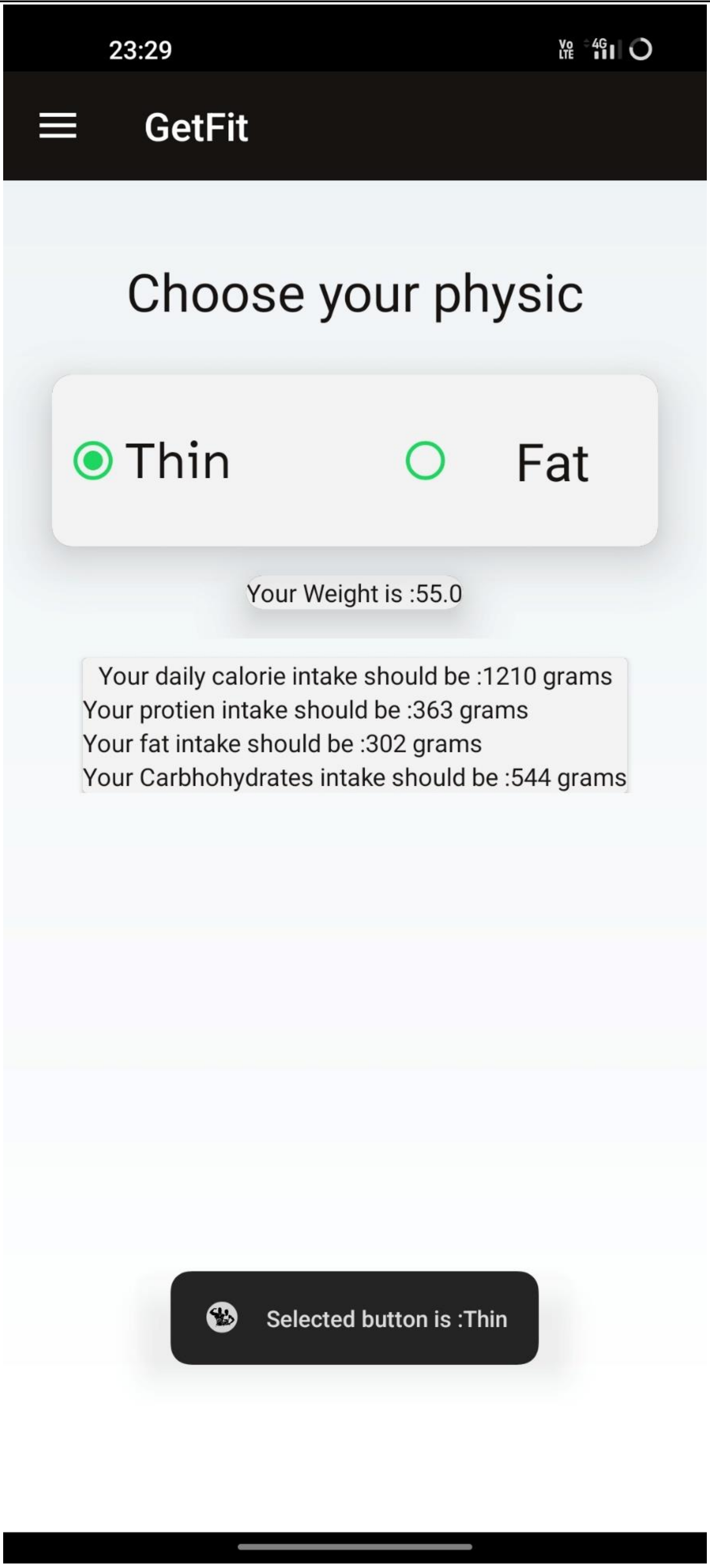


Fig .22 Home Activity



Your BMI value is: 21.484373

Fig .21 BMI Activity



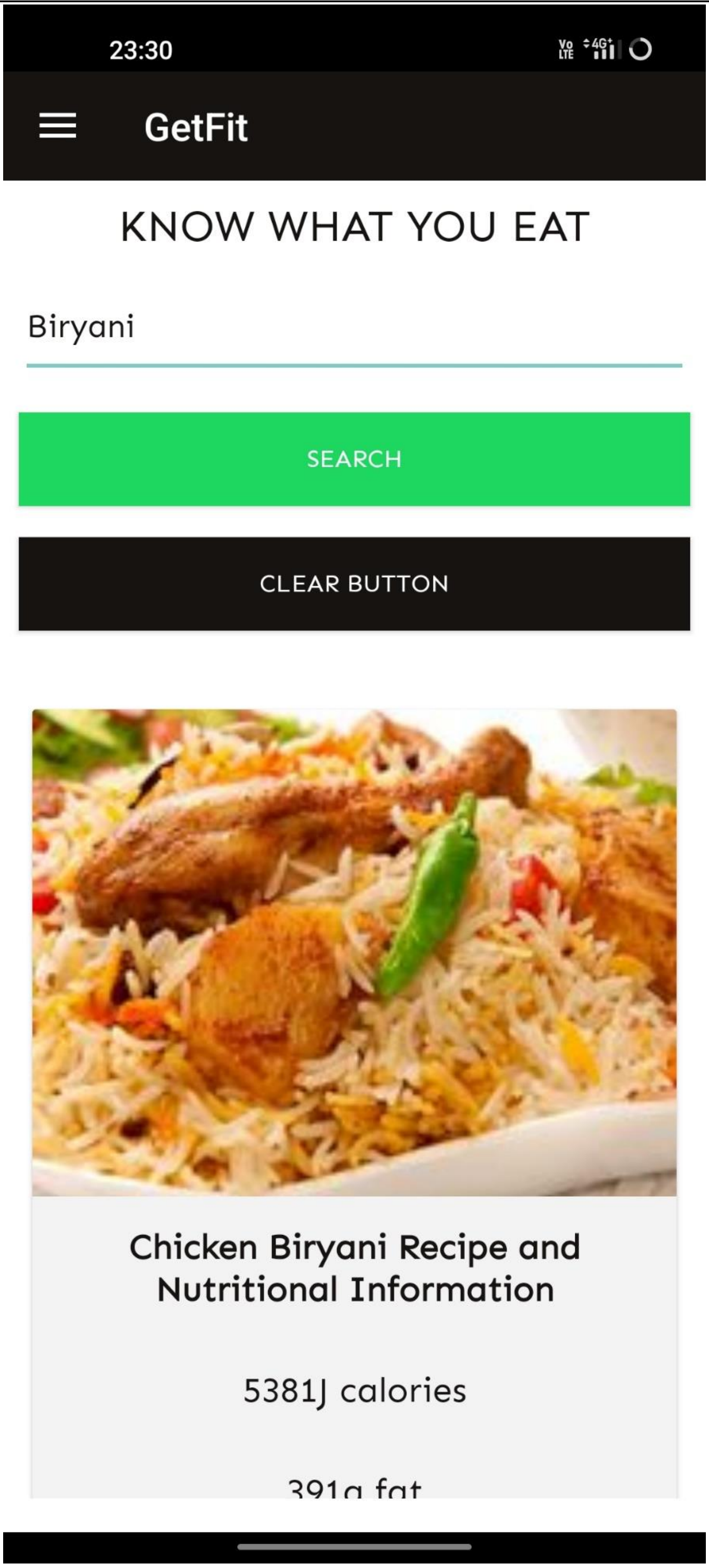
Thin [✓] Fat

Your Weight is :55.0

Your daily calorie intake should be :1210 grams
Your protien intake should be :363 grams
Your fat intake should be :302 grams
Your Carbhohydrates intake should be :544 grams

Selected button is :Thin

Fig .22 Fitness Activity

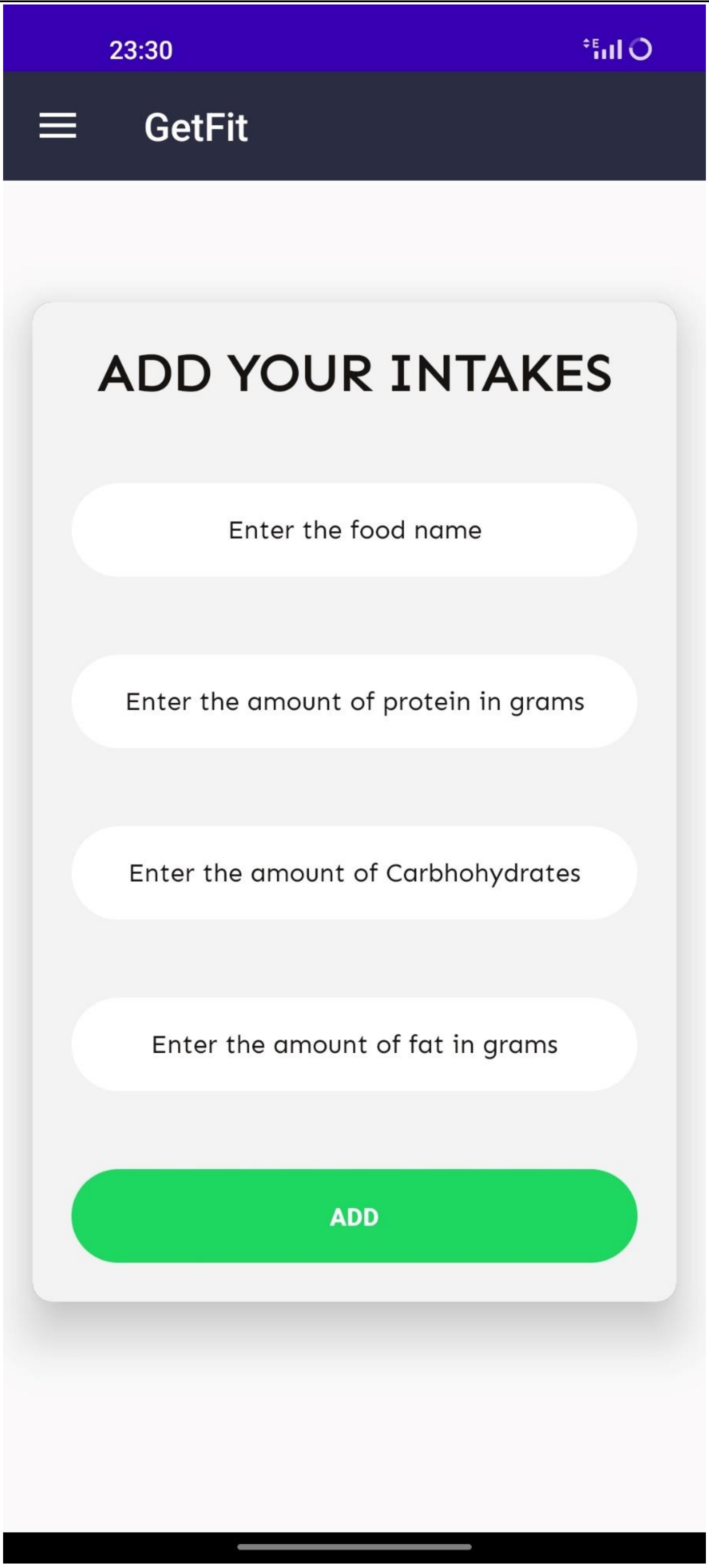


Chicken Biryani Recipe and Nutritional Information

5381J calories

391g fat

Fig .23 Food API Activity



ADD YOUR INTAKES

Enter the food name

Enter the amount of protein in grams

Enter the amount of Carbhohydrates

Enter the amount of fat in grams

Fig .24 Food intake cloud upload Activity

CONCLUSION AND FUTURE ENHANCEMENT

Chapter 10

CONCLUSION AND FUTURE ENHANCEMENT

10.1 CONCLUSION

Fitness is the most lacking aspect of today's individuals; people don't consider fitness as an important part of their routine and try to simply things by reduce physical works. Lacking in fitness might lead to many disorders and diseases, people may become too lazy and prone to easy fractures. This also affects immunity and creates obesity. The human body needs all the vital nutrients not only fats and carbohydrates, the body needs more protein to build up muscle rather than fat. Consuming more fat contents and junk foods might lead to cardio muscular diseases such as heart attacks, vein blockages and cancers. This application eliminates the need for a personal trainer or to eat costly or rare foods to maintain fitness, by helping us to eat our daily foods in respective quantities.

10.2 FUTURE ENHANCEMENTS

There are some enhancements which will be implemented in future with more user-friendly application. Some of the enhancements are given below.

- Guiding physical workouts.
- Finding gyms in new places.
- Booking gyms for workouts.
- Consulting a trainer for extra advice.
- Specific training simulations for their desired body.

BIBLIOGRAPHY AND REFRENCES

Chapter 11
BIBLIOGRAPHY AND REFRENCES

11.1 REFRENCES

1. M Kranz, A Möller, N Hammerla, S Diewald : The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices 2013
2. N Gupta, S Jilla : Digital fitness connector: smart wearable system 2014
3. V Gay, P Leijdekkers : Bringing health and fitness data together for connected health care: mobile apps as enablers of interoperability 2015
4. Y Setiakarnawijaya, E Safadilla : Android-based physical fitness software guidance 2021
5. A Hannan, MZ Shafiq, F Hussain, IM Pires : A portable smart fitness suite for real-time exercise monitoring and posture correction 2021

8.2 REFERENCE WEBSITES

1.http://www.webopedia.com/TERM/A/Android_SDK.html
2.https://developer.android.com/codelabs/build-your-first-android-app#0
3.https://developer.android.com/training/basics/firstapp