



Tribhuvan University

Faculty of Humanities and Social Sciences

A PROJECT REPORT

On

BuildGainz (The Fitness App)

Submitted to

Department of Math and ICT

Janamaitri Multiple Campus

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Rupesh Karki, 6-2-263-23-2019

Amrit Giri, 6-2-263-04-2019

BCA 6th Semester, III Year

Under the Supervisor of

Kamal Tamrakar



Tribhuvan University

Faculty of Humanities and Social Sciences

Janamaitri Multiple Campus

SUPERVISOR RECOMENDATION

I hereby recommend that this project prepared under my supervision by Rupesh Karki & Amrit Giri entitled “**BuildGainz (The Fitness App)**” in partial fulfillment of the required for the degree of Bachelor of Computer Application is recommended for the final evaluation.

SIGNATURE

Kamal Tamrakar

SUPERVISOR

BCA 6th Semester, III year

Department of Math and ICT

Janamaitri Multiple Campus

Kuleshwor height, Kathmandu



Tribhuvan University

Faculty of Humanities and Social Sciences

Janamaitri Multiple Campus

LETTER OF APPROVAL

This is to certify that this project prepared by Rupesh Karki and Amrit Giri entitled “**BuildGainz (The Fitness App)**” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

<hr/> <p>Mr. Kamal Tamrakar SUPERVISOR Department of Math and ICT Janamaitri Multiple Campus Kuleshwor height, Kathmandu</p>	<hr/> <p>Mr. Kamal Tamrakar HOD/CO-ORDINATOR Department of Math and ICT Janamaitri Multiple Campus Kuleshwor height, Kathmandu</p>
<hr/> <p>SIGNATURE of Internal Examiner Internal Examiner</p>	<hr/> <p>SIGNATURE of External Examiner External Examiner</p>

ABSTRACT

BuildGainz is a dynamic fitness app crafted to guide users on their wellness journey. The app boasts a user-friendly interface, commencing with a secure login and intuitive sign-up process. The Dashboard centralizes key features, offering quick access to exercise plans spanning 3 to 6 days, step tracking, and personalized profiles.

BuildGainz stands out with its emphasis on customization. Exercise Plan Pages cater to varied workout durations, while the Tracking Steps Page motivates users with daily step insights. The app's Profile Page enables personalization, connecting users to fitness calculators for informed decision-making.

An outstanding feature is the Fitness Calculator Page, housing tools like BMI, BMR, Calorie Calculator, and Ideal Body Weight Calculator. This suite empowers users to tailor their fitness approach. BuildGainz reflects our commitment to a holistic fitness experience, uniting functionality, accessibility, and personalization for users at every fitness level.

Keyword: *Android Studio, Java, Firebase, etc.*

ACKNOWLEDGEMENT

We are students of BCA-6th semester of Janamaitri Multiple Campus, in BCA we have to do minor project work for practical work so we develop the web base website and mobile app as its part. We are glad to present our project “**BuildGainz (The Fitness App)**” which is implementation of our study in real life. We express our sincere gratitude to those helping hands that come forward and directed us towards the success of the whole project with the desired output.

We would like to express special thanks to our project supervisor **Mr. Kamal Tamrakar** for his valuable and great support and guidance in all the happenings regarding to the project. We must express our thanks to the BCA, HOD for his encouragement and guidance throughout the project.

We are indebted to the department of information technology for providing support to add on our venture. Last, but not the least, we would like to thank our teachers and colleagues who have been knowingly or unknowingly the part of this project and lent support and views during the entire development time.

Table of Contents

ABSTRACT.....	iv
ACKNOWLEDGEMENT	v
LIST OF ABBREVIATIONS	viii
LIST OF FIGURES	ix
LIST OF TABLES	x
CHAPTER 1: INTRODUCTION	1
1.1 Introduction.....	1
1.2 Problem Statement.....	1
1.3 Objectives.....	2
1.4. Scope and Limitations.....	3
1.4.1 Scope of fitness application	3
1.4.2 Limitation of Fitness app:	3
1.5 Development Methodology.....	4
1.6 Report Organization.....	5
CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW	6
2.1 Background Study.....	6
2.2 Literature Review.....	7
CHAPTER 3: SYSTEM ANALYSIS AND DESIGN	9
3.1 System Analysis.....	9
3.1.1 Requirement Analysis.....	9
3.1.2 Feasibility Analysis (Technical, operational, economic)	11
3.1.3 Data Modelling: ER DIAGRAM	12
3.1.4 Process Modelling: DFD	13
3.2 System Design.....	15
3.2.1 Architectural Design.....	15
3.2.2 Database Schema Design.....	16
3.2.3 Interface Design.....	16
3.2.4 Physical DFD	21
CHAPTER 4: IMPLEMENTATION AND TESTING	23
4.1 Implementation	23
4.1.1 Tools Used.....	23
4.1.2 Implementation Details of Modules	25
4.2 Testing.....	27
Unit Testing	27

4.2.1 Test Cases for Unit Testing.....	27
4.2.2 Test Cases for System Testing	28
CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS	29
5.1 Conclusion	29
5.2 Lesson learnt/Outcome	29
5.3 Future Prospects.....	30
APPENDICES	31
Screen Shots.....	31
REFERENCE.....	34

LIST OF ABBREVIATIONS

DFD	Data Flow Diagram
ER	Entity Relationship
IEEE	Institute of Electrical and Electronics Engineers, Inc.
CASE	Computer-Aided Software Engineering
CRUD	Create, Read, Update, Delete
APIs	Application Programming Interfaces
SRS	System Requirement specification

LIST OF FIGURES

Figure 1.5: Waterfall Method for BuildGainz.....	4
Figure 3.1.1: Use Case Diagram for Fitness app.....	10
Figure 3.1.3: Entity-Relationship (ER) diagram for fitness app	12
Figure 3.1.4: Context Diagram of BuildGainz.....	13
Figure 3.1.4.2: Level 1 DFD of BuildGainz	14
Figure 3.2.1: Architectural Design of BuildGainz.....	15
Figure 3.2.2: Database Schema Diagram for BuildGainz (The fitness app).....	16
Figure 3.2.3.1: Splash Screen.....	17
Figure 3.2.3.2: Get Started Screen	17
Figure 3.2.3.3: Login page for BuildGainz.....	18
Figure 3.2.3.4: Forget Password Page.....	18
Figure 3.2.3.5 Reset Password Screen	19
Figure 3.2.3.6: Sign up Page for BuildGainz.....	19
Figure 3.2.3.7: Dashboard Page for BuildGainz.....	20
Figure 3.2.3.8: Gantt Chart of BuildGainz.....	21
Figure 3.2.4: Physical DFD of BuildGainz.....	22
Figure 4.1.1.1: Code for Step Counter for BuildGainz.....	23
Figure 4.1.1.2: BMI code for BuildGainz.....	24
Figure 4.1.1.3: Firebase code for SignUp	25

LIST OF TABLES

Table 4.2.1: Test Cases for Unit Testing.....	27
---	----

CHAPTER 1: INTRODUCTION

1.1 Introduction

Welcome to BuildGainz, the ultimate fitness app designed to empower your journey towards a healthier lifestyle. Whether you're a fitness enthusiast or just starting, BuildGainz offers a seamless experience tailored to your needs.

With a secure login, personalized exercise plans for 3 to 6 days, and intuitive tracking features, BuildGainz keeps your fitness goals within reach. The dashboard provides a comprehensive overview, while the profile and dashboard pages allow you to customize your experience.

BuildGainz goes beyond workouts; our Calculator page provides insights into BMI, BMR, Caloric needs, and Ideal Body Weight. And, if your goals change, our Delete Account feature ensures your data is handled securely.

Embark on your fitness journey with BuildGainz – because every gain matter on the path to a stronger, healthier you.

1.2 Problem Statement

Fitness apps face several challenges, including inaccurate tracking and monitoring of physical activities, lack of personalization and customization in workout plans, potential inaccuracies in nutrition guidance, privacy and security concerns with social features, and the need for effective and reliable personalized coaching. These issues can negatively impact users' ability to set and achieve meaningful fitness goals.

Another issue is the lack of customization in workout plans, as many apps may not cater to individual users' fitness levels, goals, or preferences, resulting in generic or ineffective plans that do not meet their unique needs. Additionally, nutrition guidance in fitness apps may not be comprehensive or accurate, leading to suboptimal nutrition choices or potential health risks.

As an analyst with experience in software development, technology, and business operations, it is crucial to address these challenges to enhance the overall user experience and ensure fitness apps remain valuable tools in supporting individuals' health and fitness goals. By addressing these challenges, fitness apps can become more effective, user-friendly, and reliable, providing users with meaningful and personalized experiences to achieve their fitness objectives. [1]

1.3 Objectives

The objective of this project is as followed:

1. User Authentication and Onboarding:

- Provide a seamless and secure user authentication process through the Login and Sign-Up pages, ensuring a smooth onboarding experience for users new to the BuildGainz platform.

2. Data Tracking and Visualization:

- Implement comprehensive tracking features, including exercise plans for varying durations (3, 4, 5, 6 days), step tracking, and a user-centric dashboard to visually represent fitness progress, fostering motivation and engagement.

3. Personalization and Profile Management:

- Enable users to personalize their fitness journey through the Profile page, allowing them to input and update key information. This ensures tailored exercise plans and accurate fitness calculations based on individual profiles.

4. Security and Account Management:

- Prioritize user security by incorporating a secure Change Password feature and offering users the option to delete their accounts if needed. Implement robust data protection measures to safeguard user information.

5. Comprehensive Fitness Calculators:

- Integrate a suite of fitness calculators, including BMI, BMR, Calorie Calculator, and Ideal Body Weight Calculator, to empower users with valuable insights into their health and fitness metrics.

6. User-Friendly Interface and Experience:

- Design an intuitive and user-friendly interface across all pages, ensuring easy navigation and a positive user experience. Strive for clarity in presenting fitness data, exercise plans, and calculations, promoting accessibility for users of all fitness levels.

1.4. Scope and Limitations

1.4.1 Scope of fitness application

1. User Authentication and Management:

- Login functionality with secure user authentication.
- User registration and profile creation.
- Password recovery through the "Forgot Password" feature.

2. Dashboard and Fitness Tracking:

- Personalized dashboard displaying fitness statistics and progress.
- Exercise plans for varying durations (3, 4, 5, 6 days) to cater to user preferences.
- Step tracking to monitor daily activity levels.

3. Profile Management:

- User profile with customizable details (e.g., profile picture, personal information).
- Change password feature to enhance account security.
- Account deletion option for users who wish to discontinue using the app.

4. Fitness Calculators:

- BMI Calculator for assessing body mass index.
- BMR Calculator for estimating basal metabolic rate.
- Calorie Calculator to determine daily caloric needs.
- Ideal Body Weight Calculator for goal setting and tracking.

1.4.2 Limitation of Fitness app:

1. Device and Platform Dependency:

- The app's functionality may be influenced by the capabilities of the user's device and operating system.

2. Accuracy of Calculators:

- The accuracy of fitness calculators (BMI, BMR, etc.) is dependent on user-input data accuracy and should not be considered as professional medical advice.

3. Internet Connection Requirement:

- Some features, such as account creation and data synchronization, may require an active internet connection.

4. User Responsibility:

- Users are responsible for providing accurate health information for precise fitness calculations.

5. **Security Considerations:**

- While efforts are made to ensure data security, users should be aware of potential security risks inherent in any online platform.

6. **Limited Personalization:**

- The app's personalization is limited to user profile settings, and it may not cater to highly individualized fitness requirements.

1.5 Development Methodology

The development of the BuildGainz fitness app follows the Waterfall approach, a traditional and linear model that emphasizes a systematic and sequential process. This methodology is well-suited for projects with clear and well-defined requirements, such as the development of a fitness application. The Waterfall approach ensures a systematic and structured development process for the BuildGainz fitness app, allowing for a comprehensive and well-tested final product.

Waterfall Method for BuildGainz(The Fitness App)

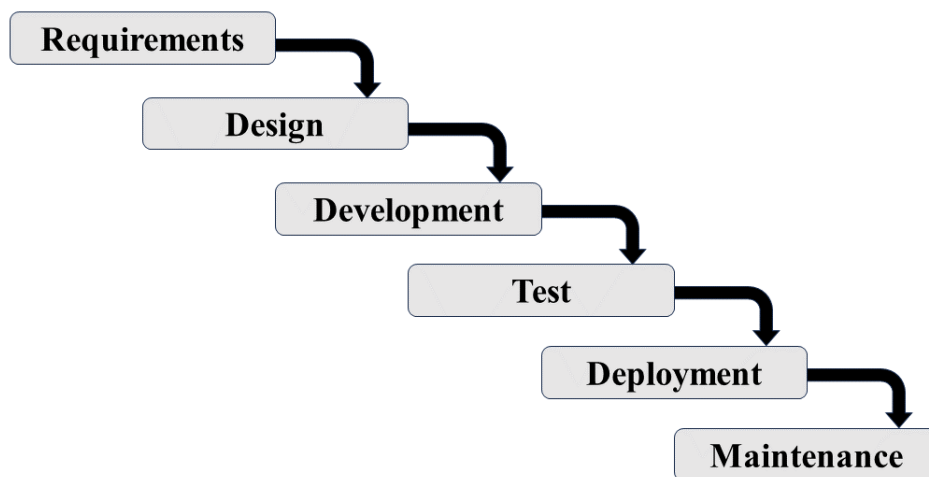


Figure 1.5: Waterfall Method for BuildGainz

1. Requirements Gathering:

This phase collects all the requirements that your app needs, including functionality, functionality, and user interface. For fitness apps, for example, this includes tracking workouts, setting fitness goals, monitoring progress, and more.

2. Design:

Once the requirements are known, the design phase begins. During this phase, a detailed plan of the app's architecture is created, including user interface, data flow, and data storage.

3. Development:

Once the design phase is complete, the development phase begins. This includes writing the code for your app, including front-end and back-end development.

4. Test:

Once development is complete, the app should be thoroughly tested to ensure it meets all requirements and works as intended. This includes usability, functionality and performance testing.

5. Deployment:

After testing, the app is deployed to production and made available to end users.

6. Maintenance:

Once deployed, apps require ongoing maintenance and support to keep them functioning properly and meeting user needs.

1.6 Report Organization

The material presented in the main report is organized into five chapters. First one is Introduction chapter. After this Chapter 2 describes the Background Study and Literature Review performed during and before starting this project. Chapter 3 provides an account of the system analysis and design. It consists of subtopics System Analysis which consists of sub subtopic Requirement Analysis, Feasibility Analysis, Data Modeling, Process Modeling and another subtopic System Design which consists of sub-subtopics Architectural Design, Database Schema Design, Interface Design, Physical DFD. Chapter 4 incorporates the brief introduction on testing and lists all the Tests performed until this phase of the project, Chapter 5 contains Conclusion and Future Recommendations.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

The fitness industry has witnessed significant growth in recent years, with an increasing number of individuals prioritizing their health and well-being. The rise of smartphones and wearable devices has further fueled this growth, creating a demand for innovative fitness applications that can seamlessly integrate into users' daily lives.

Purpose of BuildGainz: BuildGainz is developed with the vision of offering a one-stop solution for fitness enthusiasts. The app addresses key aspects of fitness, including exercise planning, step tracking, personalized profiles, and a suite of fitness calculators. By consolidating these features into a single platform, BuildGainz aims to streamline the user experience and make fitness management more accessible.

Key Features:

1. User Authentication:

- Secure login and sign-up processes ensure user data privacy.
- Forgot Password functionality for easy recovery.

2. Dashboard:

- Quick access to relevant features like exercise plans, step tracking, and fitness calculators.

3. Exercise Planning:

- Tailored exercise plans for 3, 4, 5, and 6 days cater to users with varying schedules.

4. Step Tracking:

- Daily step counts offer insights into physical activity levels.
- Goal-setting features encourage users to achieve and surpass their step targets.

5. Profile Management:

- User profiles allow individuals to input and update personal information.

6. Password Management:

- Change Password and Delete Account features provide users with control over their accounts.

7. Fitness Calculators:

- BuildGainz incorporates BMI, BMR, Calorie Calculator, and Ideal Body Weight Calculator.
- These tools empower users with valuable information for optimizing their fitness routines.

Technology Stack:

BuildGainz utilizes cutting-edge technologies to ensure a seamless and responsive user experience. The app is built on a robust backend infrastructure, employing secure authentication protocols to safeguard user data.

2.2 Literature Review

2.2.1 Introduction

Fitness apps, now widely used, play a pivotal role in promoting physical activity, health monitoring, and overall well-being. As smartphone and wearable usage becomes pervasive, these apps offer users convenient and accessible means to track, plan, and improve their fitness routines. This review delves into existing research on fitness apps, focusing on their effectiveness, features, user engagement, and potential impact on health outcomes.

1. Effectiveness of Fitness Apps: Studies consistently show that fitness apps effectively encourage physical activity and improve health outcomes. A meta-analysis by Xie et al. [1] revealed that individuals using these apps displayed higher levels of physical activity than those without. Real-time feedback, goal-setting, and personalized exercise plans contribute to motivating users to engage in regular exercise.

2. Features of Fitness Apps: Fitness apps differ in features, catering to various user preferences and goals. Maher et al. [2] identified key features enhancing user engagement, including goal-setting, progress tracking, social connectivity, and gamification elements. Integration of wearables and sensors for real-time data collection improves tracking accuracy, providing users with a comprehensive overview of their fitness journey.

3. User Engagement and Adherence: While initial adoption of fitness apps is high, maintaining user engagement poses a challenge. Lyons et al. [3] explored factors influencing long-term adherence, emphasizing app usability, personalized feedback, and social support. Gamification strategies, such as rewards and challenges, positively impact user engagement and adherence (Carnes et al., 2019) [4].

4. Health Outcomes and Behavior Change: Research indicates positive associations between fitness app use and improvements in health outcomes. Conroy et al. [5] reported these associations with physical activity levels, weight management, and cardiovascular health. However, more rigorous studies are needed to establish causation and explore long-term effects.

5. Challenges and Considerations: Despite potential benefits, fitness apps face challenges related to data privacy, measurement accuracy, and potential promotion of unhealthy exercise behaviors. Grundy et al. [6] emphasized addressing these challenges to ensure responsible development and deployment, safeguarding user trust and well-being.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

This project uses structured approach which involves breaking down the system into smaller, more manageable components. This approach focuses on defining the inputs, processes, and outputs of each component, and how they interact with each other. This approach focuses on defining the attributes and behaviors of each object, and how they interact with each other. In the case of a fitness application, both approaches can be used, depending on the specific requirements of the project.

3.1.1 Requirement Analysis

Requirement analysis identifies, analyzes, and documents system requirements, including functional and non-functional aspects. The first step in requirement analysis is to identify the functional and non-functional requirements of the fitness application.

3.1.1.1 Functional requirements

The functional requirements of BuildGainz encompass the following key features:

- **Login Page:** Users must be able to securely log in with their credentials.
- **Sign Up Page:** New users should be able to create accounts with unique usernames and passwords.
- **Forgot Password Page:** A mechanism for users to reset their password in case of forgetfulness.
- **Dashboard Page:** Provides a comprehensive overview of fitness stats, exercise plans, and step tracking.
- **Exercise Plan Page (for 3, 4, 5, 6 days):** Customized workout plans for different durations.
- **Tracking Steps Page:** Real-time and historical data on steps, including goal-setting functionality.
- **Profile Page:** Users can view and edit personal information and upload a profile picture.
- **Change Password Page:** Capability for users to update their login credentials securely.
- **Delete Account Page:** A secure process to delete user accounts.

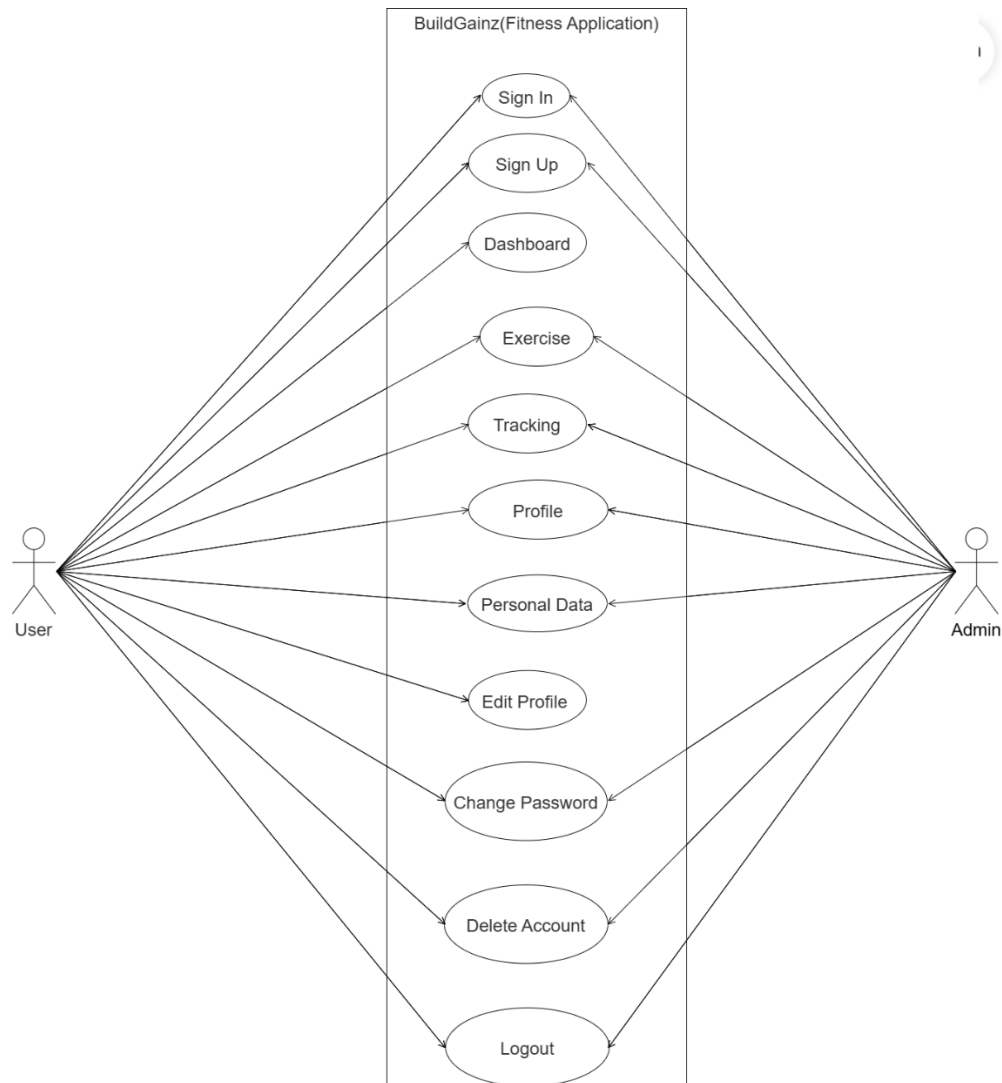


Figure 3.1.1: Use Case Diagram for Fitness app

3.1.1.2 Non-functional requirements

Non-functional requirements describe how the fitness application should perform. Some examples of non-functional requirements for a fitness application include:

- **Usability:** The app should have an intuitive interface for users of varying fitness levels.
- **Reliability:** The system must operate without errors, ensuring user confidence.
- **Security:** User data must be stored and transmitted securely to protect privacy.
- **Performance:** The app should respond promptly, even during peak usage times.
- **Compatibility:** BuildGainz must be compatible with various devices and operating systems

3.1.2 Feasibility Analysis (Technical, operational, economic)

A feasibility study analyzes a project's relevant factors, such as economic, technical, legal, and scheduling considerations, to determine the likelihood of success. It helps project managers weigh the pros and cons before investing time and money. Feasibility study checks project viability. Contingency plan advised. Economic analysis essential. The study's results determine system viability. Positive outcomes allow development, while negative one's warrant stopping the project. Preliminary investigation checks if the project is feasible and beneficial for the organization by analyzing technical, operational, and economical aspects of adding new modules and debugging existing systems. With unlimited resources and infinite time, all systems are possible. [2] There are unit aspects within the practicability study portion of the preliminary investigation:

1. Technical Feasibility
2. Operational Feasibility
3. Economical Feasibility

3.1.2.1 Technical Feasibility

BuildGainz is built in Android Studio using Java for the backend logic and Firebase for backend services and data storage. XML is used for designing the user interface. The technical stack is feasible, leveraging Android's native development environment and Firebase's scalability.

3.1.2.2 Operational Feasibility

The app's operational feasibility is evident in its user-friendly interface and features, making it practical and efficient for users to achieve their fitness goals. The integration with Firebase also ensures seamless operational performance.

3.1.2.3 Economical Feasibility

Building and maintaining BuildGainz is economically feasible due to the use of open-source Android Studio, Firebase's pay-as-you-go model, and the scalability of the application. The cost of development and operation is reasonable, considering the benefits provided to users.

This System Analysis provides an overview of the requirements and feasibility aspects of BuildGainz, emphasizing its functionality, performance, usability, and economic viability.

3.1.3 Data Modelling: ER DIAGRAM

The Entity-Relationship (ER) diagram for BuildGainz, a fitness app developed in Android Studio using Java and Firebase, represents the underlying structure of the application's database. The ER diagram visually illustrates the relationships between different entities in the system. Below is a brief overview of the key entities and their relationships:

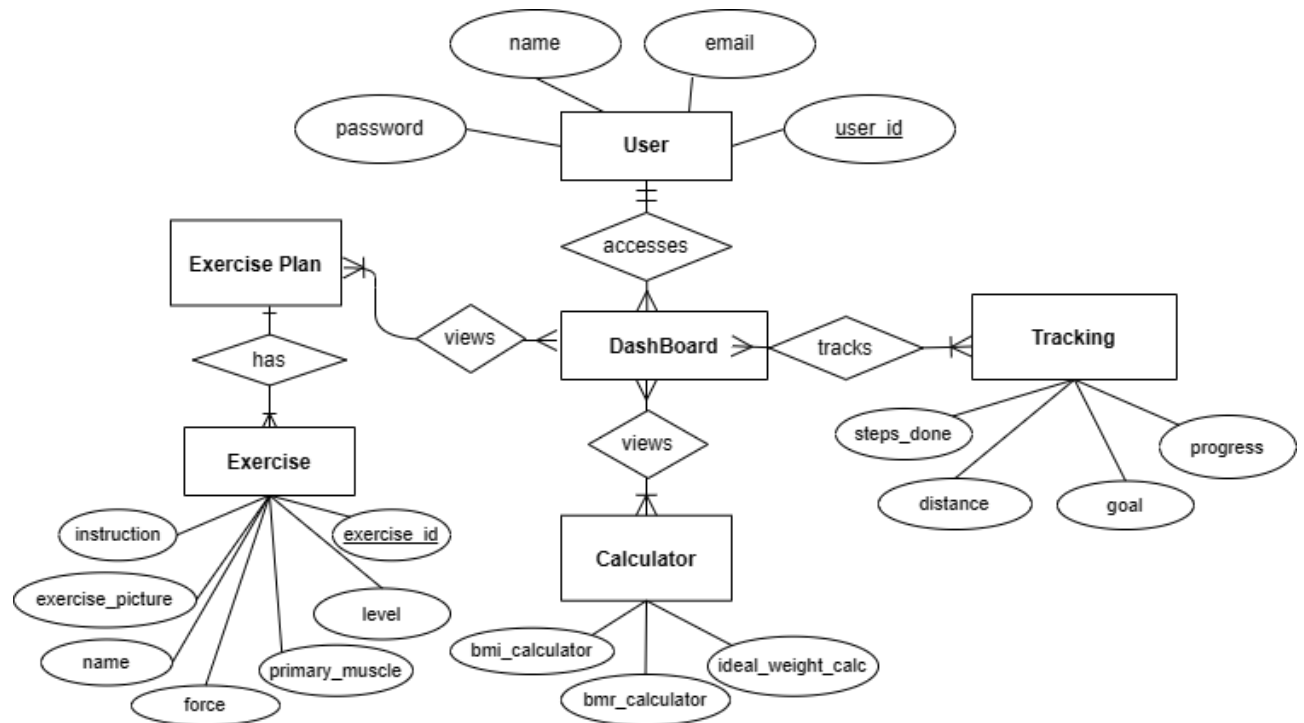


Figure 3.1.3: Entity-Relationship (ER) diagram for fitness app

3.1.4 Process Modelling: DFD

In the figure below there is two main entity that is user and admin. The user needs to be registered in into the system for using any of service. Once one is registered one can then login into the system and make use of the system. And admin update details of the profile, overview and routine.

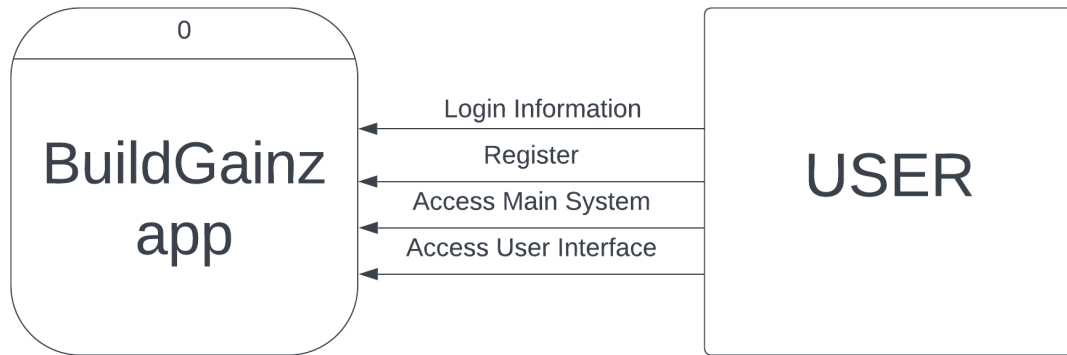


Figure 3.1.4: Context Diagram of BuildGainz

3.1.4 .2 Level 1 DFD:

Level 1 DFD elaborates on the interactions between the application interface and specific components (dashboard, tracking, calculator, exercise), illustrating the flow of data queries from the interface to these components and their respective data stores.

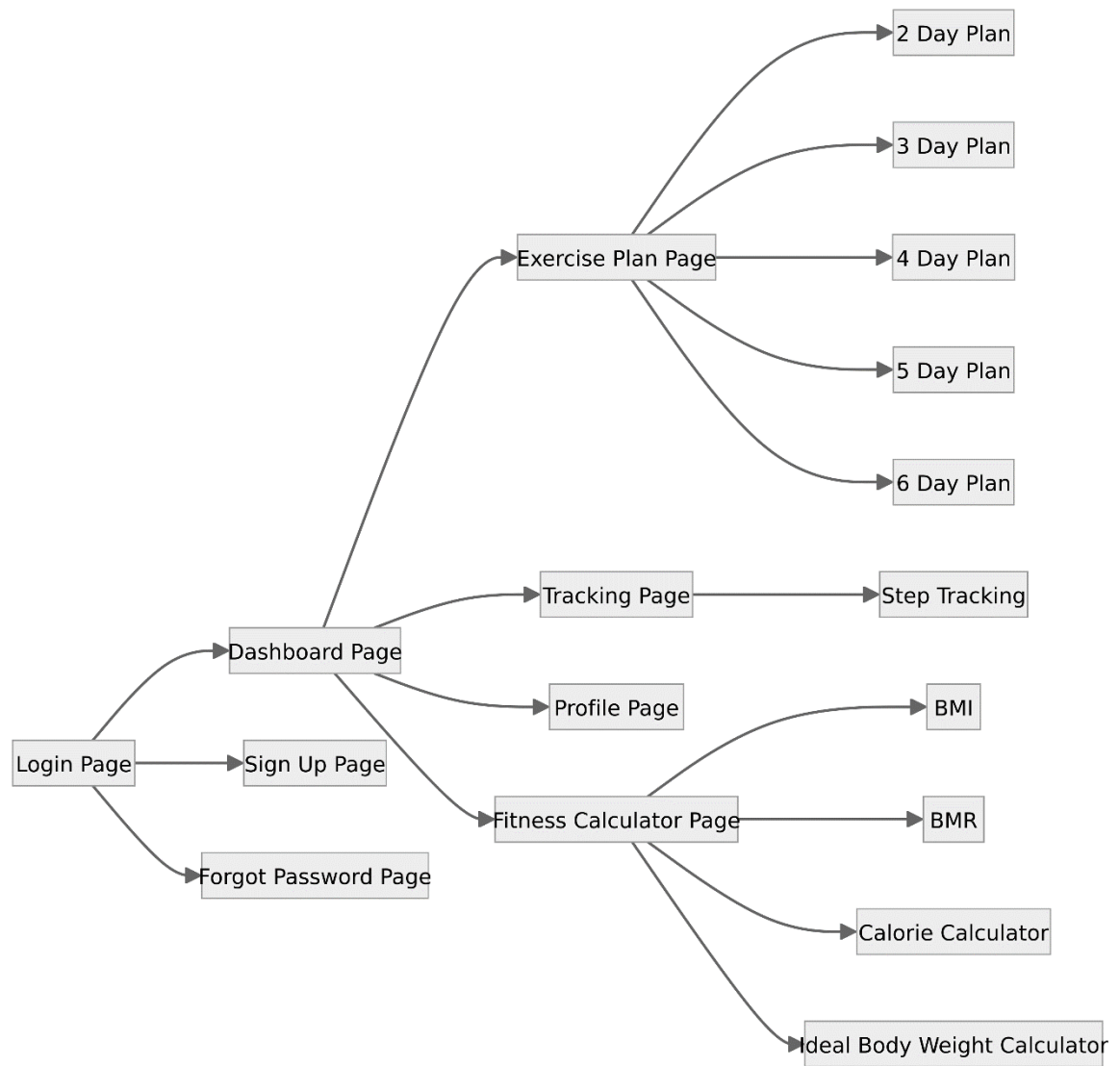


Figure 3.1.4.2: Level 1 DFD of BuildGainz

3.2 System Design

Designing a fitness app requires a structured and systematic approach to ensure a scalable, reliable, and efficient system. This brief will outline the key components and considerations involved in the system design of a fitness app, covering the architecture, scalability, performance, security, and data management aspects.

3.2.1 Architectural Design

The architectural design diagram presented depicts a high-level view of a software application. The application is organized into distinct components within a package named "Application." Here's a brief summary of the design:

1. **Users:** Represents the user interface for managing user-related functionalities.
2. **Dashboard:** Serves as a central hub or interface where users interact with various components.
3. **Tracking:** A component responsible for tracking and monitoring activities or data relevant to the application.
4. **Calculator:** A component likely focused on calculations or related functionalities within the application.
5. **Exercise:** A component likely involved in managing exercise-related functionalities within the application.

The arrows indicate the flow of data or interaction between the components. Users interact with the dashboard, which acts as a gateway to access functionalities related to tracking, calculation, and exercise. This design provides a clear overview of how the application is structured and how users can interact with its key components through the dashboard.

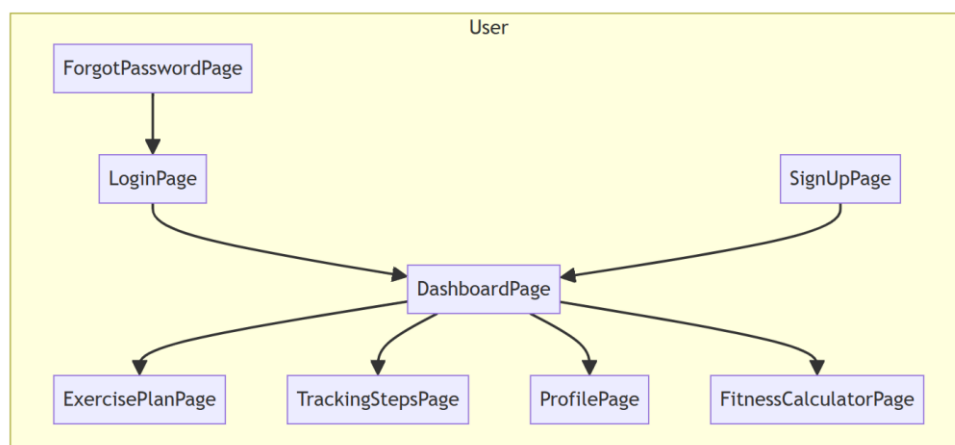


Figure 3.2.1: Architectural Design of BuildGainz

3.2.2 Database Schema Design

The database schema design for a fitness app plays a crucial role in efficiently managing and organizing the app's data. It enables seamless tracking of user activities, personalized workout plans, and progress monitoring. This brief will outline a high-level overview of the essential components and relationships within a fitness app database schema.

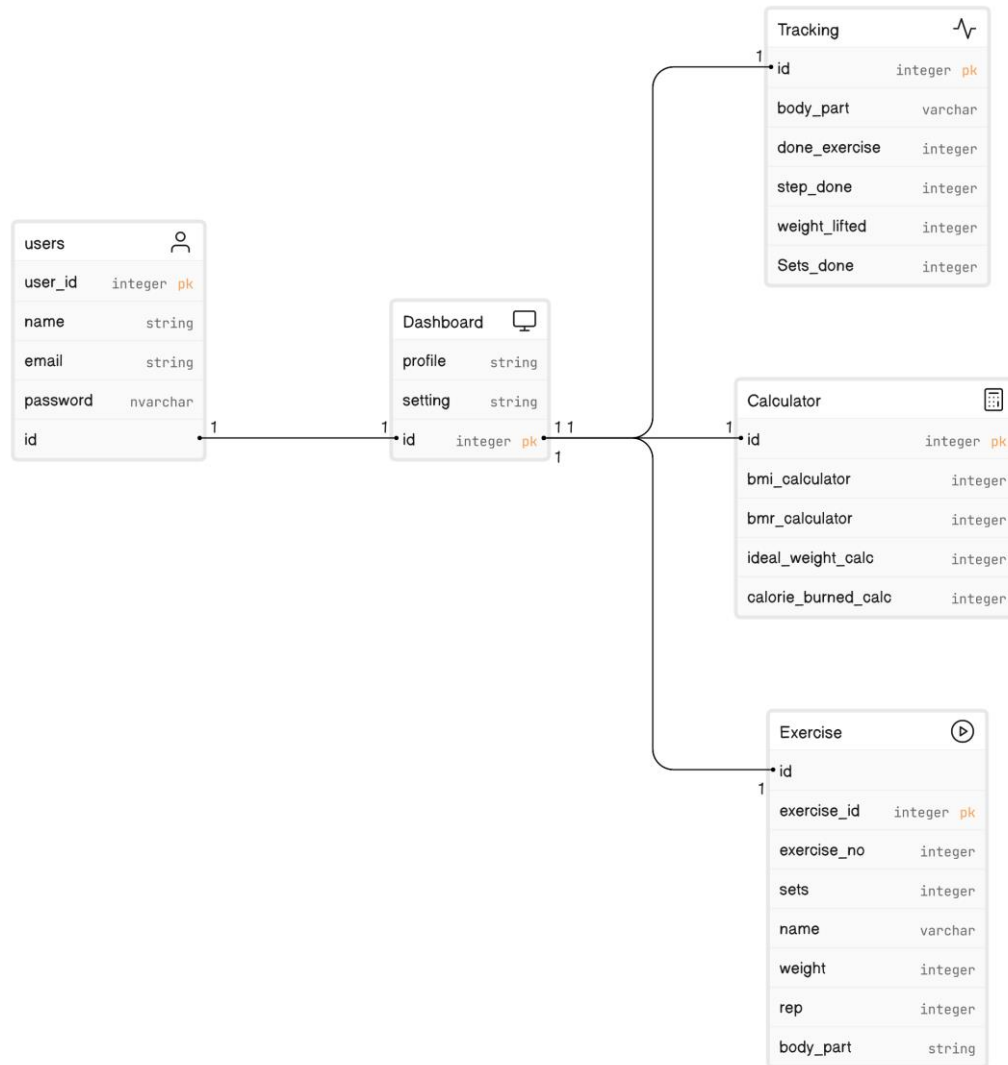


Figure 3.2.2: Database Schema Diagram for BuildGainz (The fitness app)

3.2.3 Interface Design

The user interface (UI) and user experience (UX) design of a fitness app are crucial in providing a seamless and engaging experience to users. It involves creating intuitive, visually appealing, and functional interfaces that facilitate easy navigation, effective tracking of fitness activities, and motivation for users to achieve their fitness goals.

This brief will outline key considerations and elements involved in designing the interface of a fitness app.

Here is the following design of BuildGainz (The fitness app)

3.2.3.1 Splash Screen

A splash screen is a graphical element that is displayed when a software application or video game is launched. It typically appears for a few seconds and serves as an introduction or branding element for the application.



Figure 3.2.3.1: Splash Screen

3.2.3.2 Get Started Screen

It is the type of splash screen which comes after the splash screen which is just for motivation before taking the challenge.

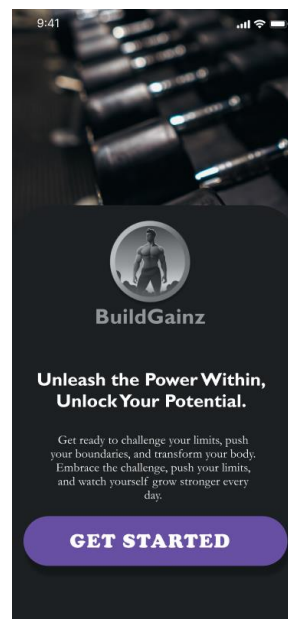


Figure 3.2.3.2: Get Started Screen

3.2.3.3 Login Page

A login page, also known as a login screen or sign-in page, is a user interface element that allows users to access a system, application, or website by entering their credentials, such as a username and password.

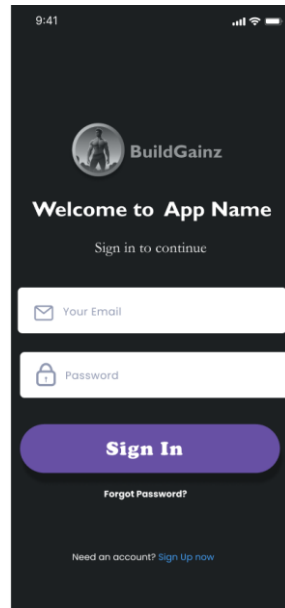


Figure 3.2.3.3: Login page for BuildGainz

3.2.3.4 Forget Password Page

A "Forgot Password" page, also known as a password recovery or password reset page, is a user interface element that allows users to regain access to their accounts when they have forgotten their password.

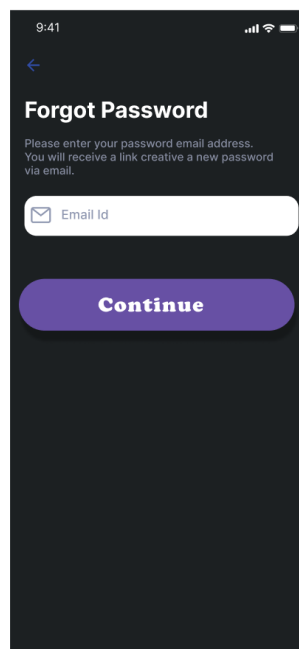


Figure 3.2.3.4: Forget Password Page

3.2.3.5 Reset Password Screen

A "Reset Password" screen or page is a user interface element that allows users to change or reset their current password to regain access to their account.

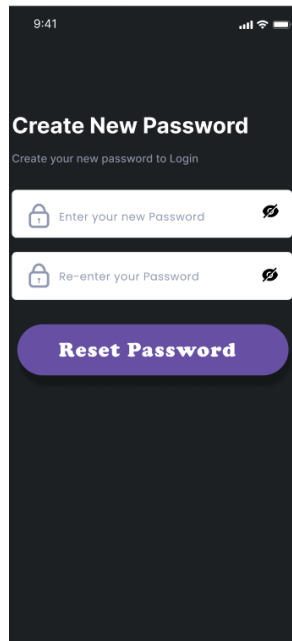


Figure 3.2.3.5 Reset Password Screen

3.2.3.6 Sign up Page

A "Sign-Up" page, also known as a registration page or create account page, is a user interface element that allows individuals to create a new account or register for a service, platform, or application.

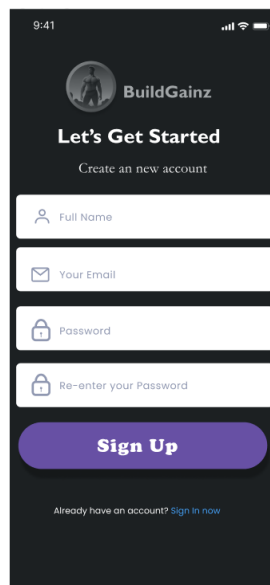


Figure 3.2.3.6: Sign up Page for BuildGainz

3.2.3.7 Dashboard

The "Dashboard" component in the fitness app acts as a central interface, offering users an overview of their fitness journey. It provides key progress, achievements, and personalized settings information. Users can access workout data, health metrics, and goals, aiding efficient fitness tracking.

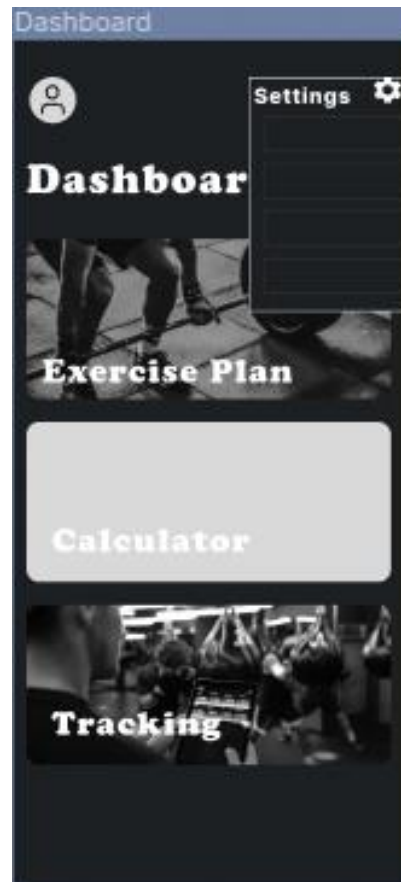


Figure 3.2.3.7: Dashboard Page for BuildGainz

Gantt chart

Our fitness project is a comprehensive endeavor that involves planning, designing, and developing, implementing and testing a fitness application over a 5-month period. The project begins with two weeks of planning, during which the project scope, objectives, and milestones are defined, and the project plan is developed. This is followed by three weeks of design and wireframing, four weeks of front-end development, five weeks of back-end development, three weeks of implementation. Finally, the project concludes with two weeks of testing and quality assurance to ensure that the app is functional and user-friendly and rest on documentation.

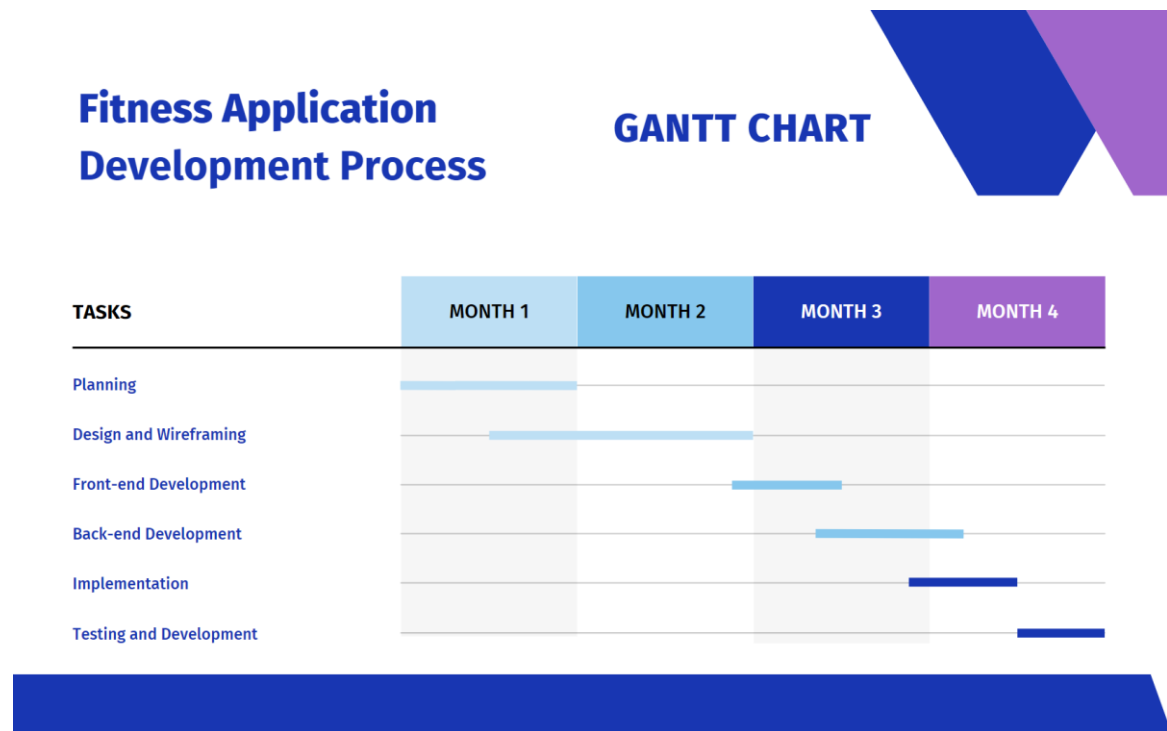


Figure 3.2.3.8: Gantt Chart of BuildGainz

Note: The above Gantt chart is just an example and the timeline and duration of each task may vary depending on the project's scope, team, and other factors.

3.2.4 Physical DFD

The Physical DFD details the system's implementation, illustrating how the application interface communicates with different components (dashboard, tracking, calculator, exercise) to interact with their respective data stores by sending and receiving requests and queries.

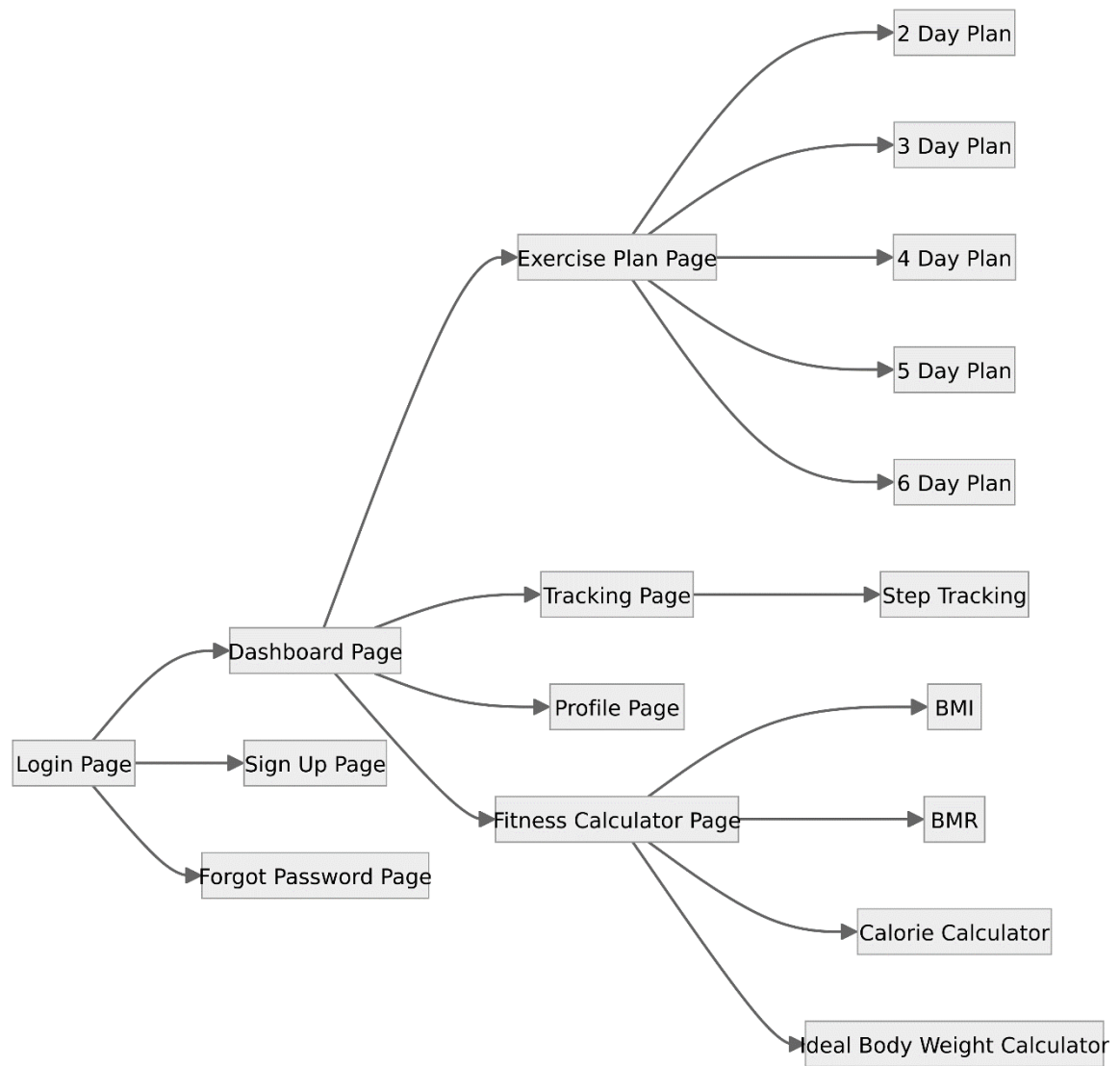


Figure 3.2.4: Physical DFD of BuildGainz

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

The system has been implemented using the following technologies: Firebase for creating database, JAVA for designing and styling the interface. Android Studio has been used as the code editor for the system.

4.1.1 Tools Used

For the Weather App, the following technologies have been used:

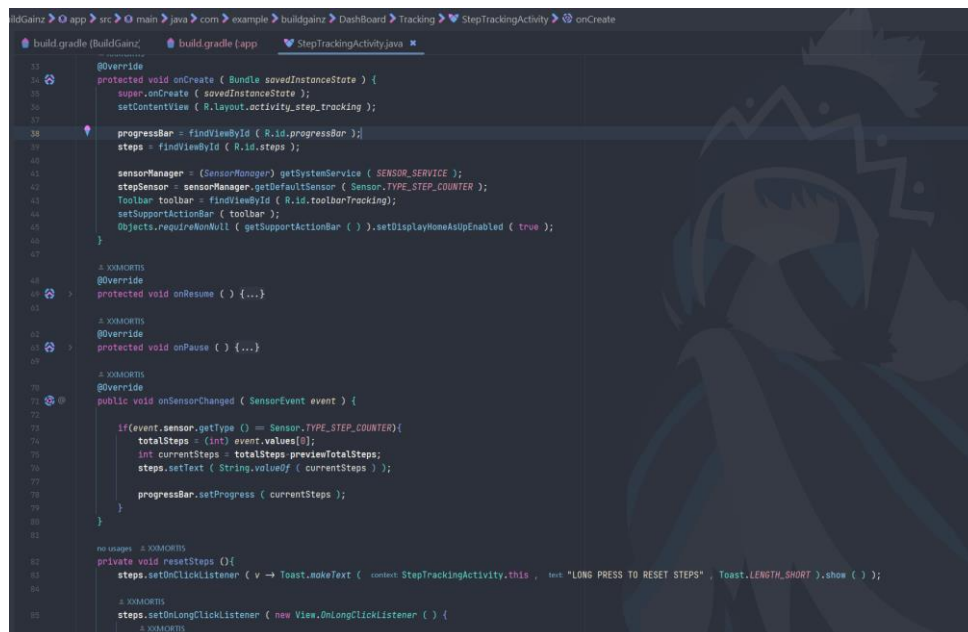
Technology [Front-End]	: Java
Technology [Back-End]	: Firebase
Database	: Firestore Database, Firebase Storage
Code Editor	: Android Studio

Android Studio

The tool I used to create this application from start to finish was Android Studio. The version of I used in making this was 2023.3.1 Patch 1, compileSdkVersion “34”, sourceCompatibility JavaVersion.VERSION_1_8.

Step Counter

The step counter class of the app requires the target device to have hardware sensors for both step counter and step detector present, without these the feature would cease to work.



```

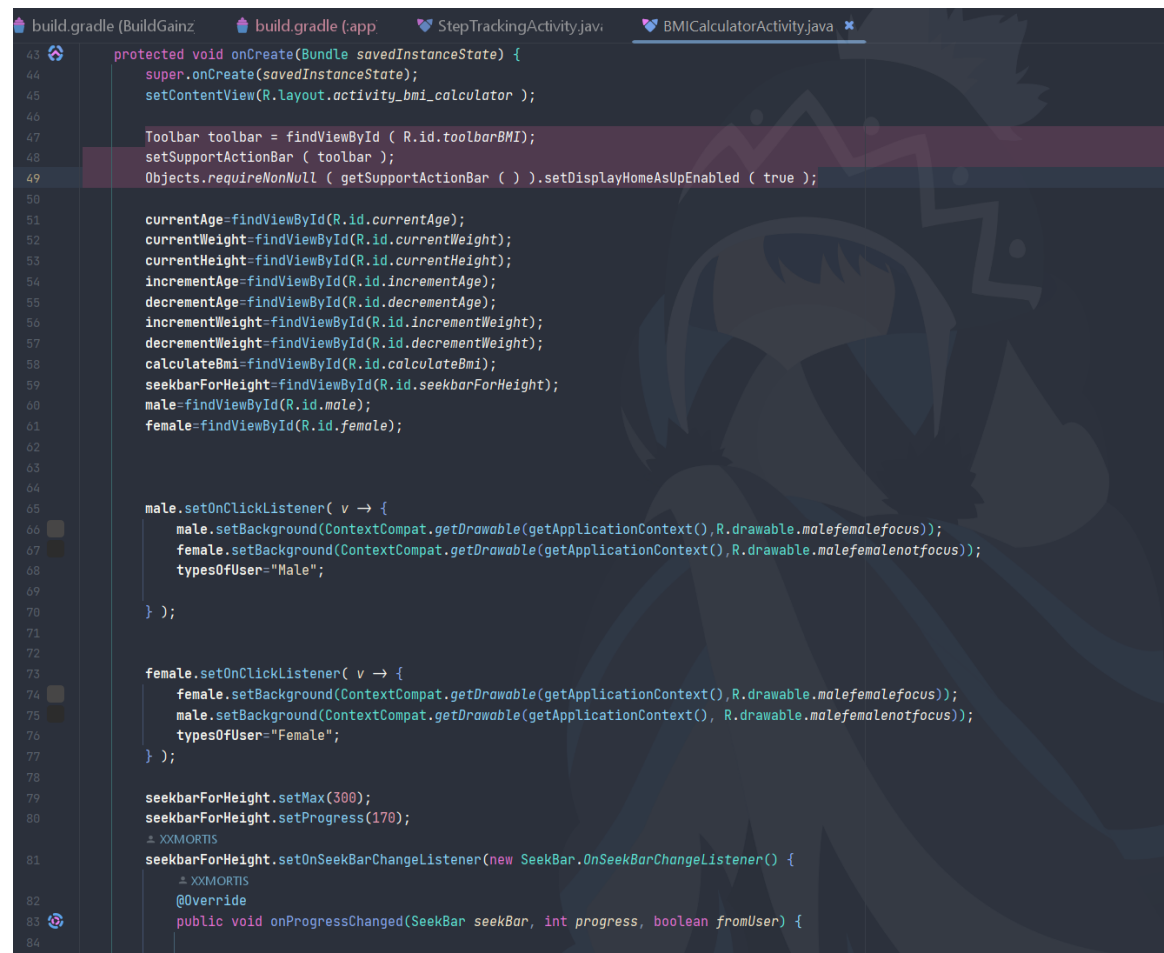
11  @Override
12  protected void onCreate ( Bundle savedInstanceState ) {
13      super.onCreate ( savedInstanceState );
14      setContentView ( R.layout.activity_step_tracking );
15
16      progressBar = findViewById ( R.id.progressBar );
17      steps = findViewById ( R.id.steps );
18
19      sensorManager = (SensorManager) getSystemService ( SENSOR_SERVICE );
20      stepSensor = sensorManager.getDefaultSensor ( Sensor.TYPE_STEP_COUNTER );
21      Toolbar toolbar = findViewById ( R.id.toolbarTracking );
22      setSupportActionBar ( toolbar );
23      Objects.requireNonNull ( getSupportActionBar ( ) ).setDisplayHomeAsUpEnabled ( true );
24  }
25
26  @XORNOTS
27  @Override
28  protected void onResume ( ) { ... }
29
30  @XORNOTS
31  @Override
32  protected void onPause ( ) { ... }
33
34  @XORNOTS
35  @Override
36  public void onSensorChanged ( SensorEvent event ) {
37
38      if (event.sensor.getType () == Sensor.TYPE_STEP_COUNTER){
39          totalSteps = (int) event.values[0];
40          int currentSteps = totalSteps - previousTotalSteps;
41          steps.setText ( String.valueOf ( currentSteps ) );
42          progressBar.setProgress ( currentSteps );
43      }
44  }
45
46  @XORNOTS
47  private void resetSteps () {
48      steps.setOnClickListener ( v -> Toast.makeText ( context, StepTrackingActivity.this, Toast.LENGTH_SHORT ).show ( ) );
49  }
50
51  @XORNOTS
52  steps.setOnLongClickListener ( new View.OnLongClickListener ( ) {
53      @XORNOTS

```

Figure 4.1.1.1: Code for Step Counter for BuildGainz

Body Mass Calculator

The BMI class allows users to simply enter both their height in cm's and their weight in kg's and be given their body mass index value. There is also a comment along with that value to let the user know what it means. The calculation is done as follows, $bmi = \text{weightValue} / (\text{heightValue} * \text{heightValue})$. The displayBMI method will display a message depending on what value is calculated, so for example between 18.5 and 25 the user can be said to have a normal BMI.



```

43 protected void onCreate(Bundle savedInstanceState) {
44     super.onCreate(savedInstanceState);
45     setContentView(R.layout.activity_bmi_calculator);
46
47     Toolbar toolbar = findViewById(R.id.toolbarBMI);
48     setSupportActionBar(toolbar);
49     Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
50
51     currentAge=findViewById(R.id.currentAge);
52     currentWeight=findViewById(R.id.currentWeight);
53     currentHeight=findViewById(R.id.currentHeight);
54     incrementAge=findViewById(R.id.incrementAge);
55     decrementAge=findViewById(R.id.decrementAge);
56     incrementWeight=findViewById(R.id.incrementWeight);
57     decrementWeight=findViewById(R.id.decrementWeight);
58     calculateBmi=findViewById(R.id.calculateBmi);
59     seekbarForHeight=findViewById(R.id.seekbarForHeight);
60     male=findViewById(R.id.male);
61     female=findViewById(R.id.female);
62
63
64
65     male.setOnClickListener( v -> {
66         male.setBackground(ContextCompat.getDrawable(getApplicationContext(),R.drawable.malefemalefocus));
67         female.setBackground(ContextCompat.getDrawable(getApplicationContext(),R.drawable.malefemalenotfocus));
68         typesOfUser="Male";
69     });
70
71
72
73     female.setOnClickListener( v -> {
74         female.setBackground(ContextCompat.getDrawable(getApplicationContext(),R.drawable.malefemalefocus));
75         male.setBackground(ContextCompat.getDrawable(getApplicationContext(),R.drawable.malefemalenotfocus));
76         typesOfUser="Female";
77     });
78
79     seekbarForHeight.setMax(300);
80     seekbarForHeight.setProgress(170);
81     // XXMORTIS
82     seekbarForHeight.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
83         // XXMORTIS
84         @Override
85         public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {

```

Figure 4.1.1.2: BMI code for BuildGainz

Firebase

For both my login and registration classes Firebase auth allows me authenticate users. For my login page Firebase saves input from both the email editText and password editText and runs it against the database. If successful the app starts an intent to the homepage of the app. The register class works in the same way saving the details from both editTexts to the database and sending a toast to say registration was successful, the user is then brought to the homepage.

```

142
143 //Register user using credentials given
144 1 usage ± XXMORTIS
145 private void registerUser ( String textFullName , String textYourEmail , String textYourPassword ) {
146     FirebaseAuth auth = FirebaseAuth.getInstance ( );
147
148     //Create User Profile
149     auth.createUserWithEmailAndPassword ( textYourEmail , textYourPassword ).addOnCompleteListener ( activity: SignUpActivity.this , t
150     if ( task.isSuccessful ( ) ) {
151         FirebaseUser firebaseUser = auth.getCurrentUser ( );
152
153         //Enter User Data into the firebase Realtime Database
154         ReadWriteUserDetails writeUserDetails = new ReadWriteUserDetails ( textFullName , textYourEmail );
155
156         //Extracting user reference from Database for Registered User
157         DatabaseReference reference = FirebaseDatabase.getInstance ( ).getReference ( path: "Registered Users" );
158
159         reference.child ( Objects.requireNonNull ( firebaseUser ).getUid ( ) ).setValue ( writeUserDetails ).addOnCompleteListe
160
161         if ( task1.isSuccessful ( ) ) {
162             //Send Verification Email
163
164             Toast.makeText ( context: SignUpActivity.this , text: "User Signed Up Successfully. Please verify your email." , To
165
166             //Go back to Sign In page
167             Intent intent = new Intent ( packageContext: SignUpActivity.this , LoginPageActivity.class );
168
169             startActivity ( intent );
170             firebaseUser.sendEmailVerification ( );
171
172         } else {
173             Toast.makeText ( context: SignUpActivity.this , text: "User registered failed. Try again!" , Toast.LENGTH_SHORT ).s
174         }
175     }
176
177     } );
178
179
180
181     } else {
182         try {
183             throw Objects.requireNonNull ( task.getException ( ) );
184         } catch ( FirebaseAuthInvalidCredentialsException e ) {

```

Figure 4.1.1.3: Firebase code for SignUp

4.1.2 Implementation Details of Modules

Application Interface:

- Class: ApplicationInterface
 - Methods:
 - getUserInput (): Gathers user input from the interface.
 - requestDashboardData (): Sends a request for dashboard data.
 - requestTrackingData (): Sends a request for tracking data.
 - requestCalculatorData (): Sends a request for calculator data.
 - requestExerciseData (): Sends a request for exercise data.

Dashboard Component:

- Class: DashboardComponent
 - Methods:
 - queryDashboardData (): Queries the dashboard data store for relevant data.

Tracking Component:

- Class: TrackingComponent
 - Methods:
 - queryTrackingData (): Queries the tracking data store for relevant data.

Calculator Component:

- Class: CalculatorComponent
 - Methods:
 - queryCalculatorData (): Queries the calculator data store for relevant data.

Exercise Component:

- Class: ExerciseComponent
 - Methods:
 - queryExerciseData (): Queries the exercise data store for relevant data.

Dashboard Data Store:

- Class: DashboardDataStore
 - Methods:
 - retrieveData (): Retrieves and returns dashboard-related data.

Tracking Data Store:

- Class: TrackingDataStore
 - Methods:
 - retrieveData (): Retrieves and returns tracking-related data.

Calculator Data Store:

- Class: CalculatorDataStore
 - Methods:
 - retrieveData (): Retrieves and returns calculator-related data.

Exercise Data Store:

- Class: ExerciseDataStore
 - Methods:
 - retrieveData (): Retrieves and returns exercise-related data.

These implementation details provide a basic structure for the modules in your fitness app, including their corresponding classes and methods.

4.2 Testing

Unit Testing

Throughout the course of developing this application, after each new feature was added or changes were made I performed unit testing. Whenever I added something, I ran the app through an emulator on my laptop as well as on both of the android devices I have. I made use of both the android monitor and logcat when testing the app and this was key in helping me identify and eradicate errors. The Gradle console also helped with error detection. In regard to Firebase, any issues I encountered were logged via crash reporting and I was able to view the log via Firebase console.

Backwards Compatibility Testing

The app was developed on android sdk version 34 with the minimum sdk version set at 28. The app has been tested on a number of different versions and multiple devices. Most of the features can be used from low versions and upwards. The only concern I would have been if the user's device does not have the hardware required for the step counter to work.

4.2.1 Test Cases for Unit Testing

Following tests of separate modules were done for this system:

S.N.	Unit	Test	Expected Result	Test Outcome
1.	Log In	Used Invalid Log in Credentials to Check Login functionality	Invalid Username or Password	Invalid Username or Password
2.	Log In	Used Valid Log in Credentials to Check Login	Navigated to Home Screen	Success
3.	Sign up	Sign to The Application	Get Registered	Registered and Navigated to Home Screen
4	Forget Password	Type your valid email to check functionality.	Check Inbox for password reset.	Sent it to your email inbox.
5.	Forget Password	Type your invalid email to check functionality.	Error	User does not exist or no longer available.

Table 4.2.1: Test Cases for Unit Testing

4.2.2 Test Cases for System Testing

After integrating all the modules into a workable system, the whole system was tested.

Following tests were performed in the System Testing phase:

- Navigation between screens was tested.
- Registration.
- Login.
- Dashboard.
- Profile Page.
- Change Profile Picture.
- Personal Data.
- Save Button in personal data.
- Setting Button
- Change Password.
- Delete Account.
- Exercise Plan.
- Calculator.
- Tracking
- Logout.

Evaluation

Testing the app was enlightening. It revealed the importance of viewing our project objectively. Amrit's feedback prompted adjustments to the homepage, simplifying it for a cleaner look. Following a suggestion on tracking, I added a page showcasing average daily steps by age, gender, and activity level. These changes could address potential user disengagement, demonstrating the value of user feedback in refining the app.

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Conclusion

This report gives an insight into the background as to why I chose to develop this application. I believed that there was a problem with apps of this type and that if done a certain way this could be solved. The idea and my motivations have been stated clearly. The requirements are described in detail and the technologies used have also been outlined in this document. I have touched on the Systems design and architecture and provided a class diagram to explain it further. I believe that I achieved most of what I have set out to do when this idea was conceived. However, there are multiple different areas which could be improved and lots of room for expansion.

In a nutshell, the fitness app is your go-to tool for a healthier you. It gives you an easy-to-use interface to track progress, set goals, access exercise plans, step tracking and calculator for bmi, bmr, ideal body weight. Whether you're a beginner or a pro, this app has got you covered. It's designed to make your fitness journey smooth and successful, ultimately leading you towards a fitter and happier life.

5.2 Lesson learnt/Outcome

Here are some simple, key lessons learned and outcomes from the fitness app:

1. **Goals Drive Motivation:** Setting clear fitness goals keeps you motivated and focused.
2. **Comprehensive Tools Help:** All-in-one features like calculators and exercise routines simplify fitness management.
3. **Healthier Lifestyle:** The app supports users in achieving a healthier lifestyle and well-being.
4. **Long-Term Success:** Consistency and dedication lead to sustained fitness results.
5. **Versatility and Inclusivity:** Design the app to cater to users of varying fitness levels and preferences.
6. **Simplicity and Intuitiveness:** Keep the app interface simple, intuitive, and easy to navigate.
7. **Positive Impact:** Strive for a positive impact on users' overall well-being and quality of life.

5.3 Future Prospects

Looking ahead, the fitness app has tremendous potential for growth and enhancement. Key areas of improvement include refining the tracker and step counter functionalities, as well as introducing a calorie counter feature to aid users in monitoring their daily intake, particularly beneficial for those with weight loss objectives. This enhancement involves integrating an extensive database of food and nutritional information, enabling users to input their meals for accurate calorie tracking.

Nothing can be ended in a single step. It is the fact that nothing is permanent in this world. So, this project also has some future enhancements in the evergreen and booming IT industry. This app can be improved in the future by adding the following functionalities:

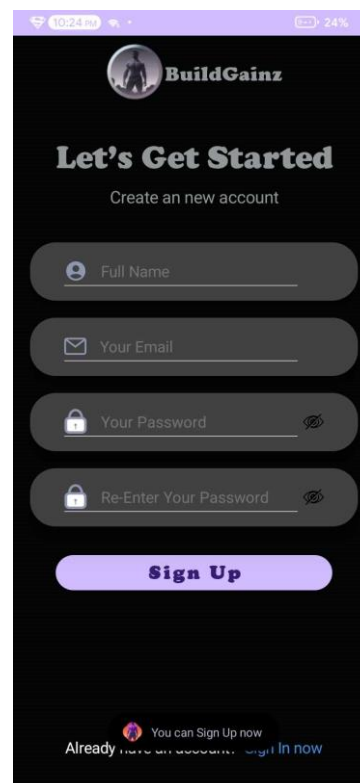
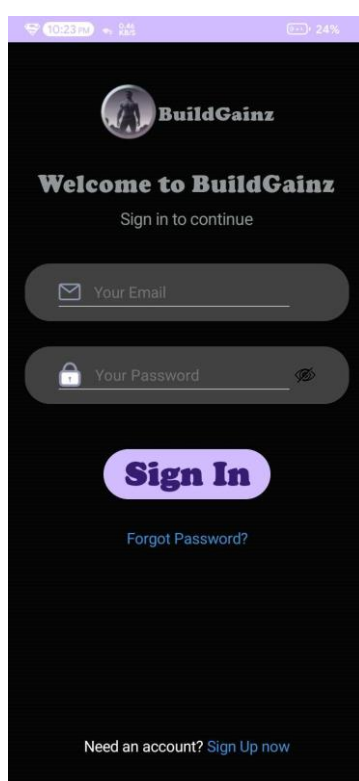
- Extending to Apple iOS.
- Create two separate apps for nutrition & exercise.
- We will track our diet through AI based scanner.
- We will include simple exercise to make people fit.

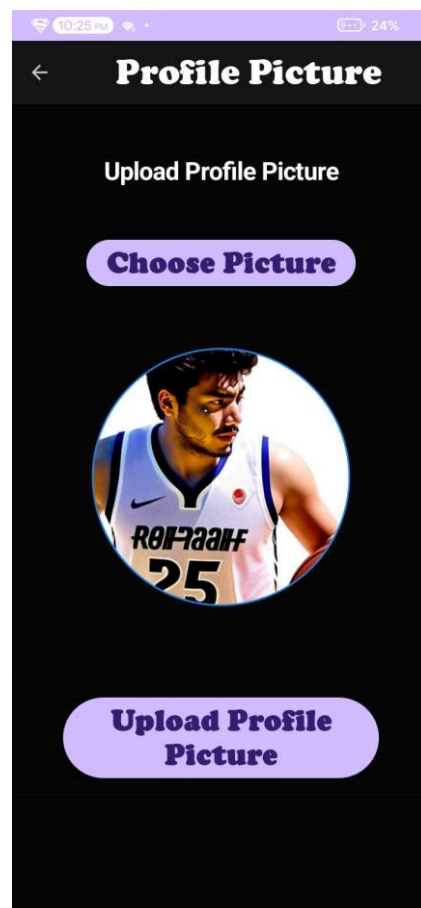
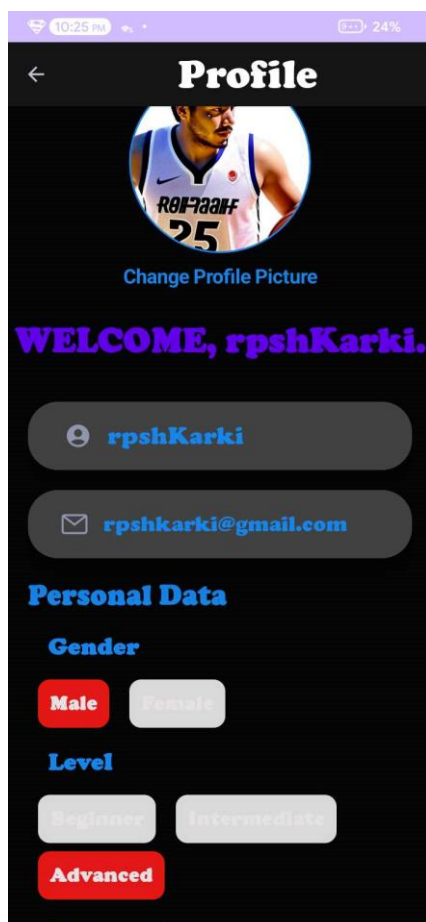
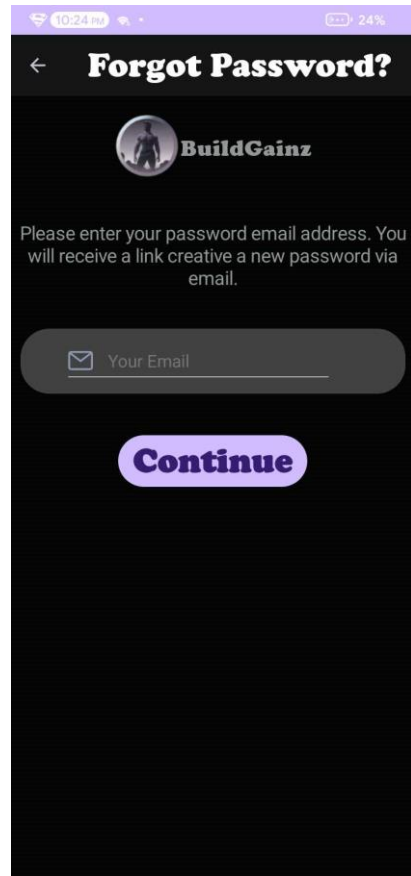
For future development, we will first consider all previously assumed options which are not yet developed. This we can make the application livelier in action, so that can be considered as popular health application. We should look for the whole prospective in the near future for a successful, most popular and user-friendly application which will be frequently updated with some extra new features as well as in the user interface.

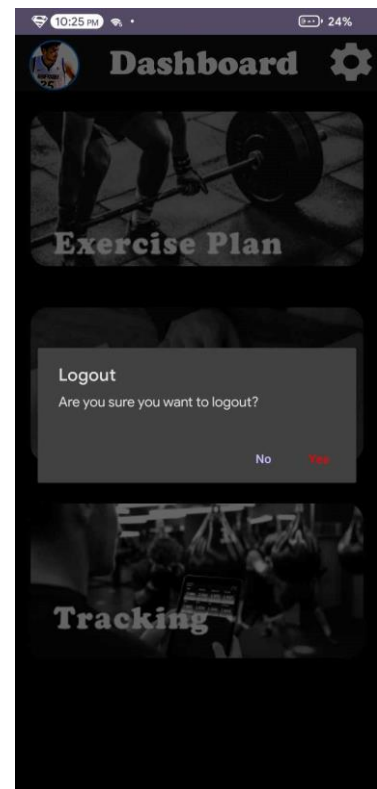
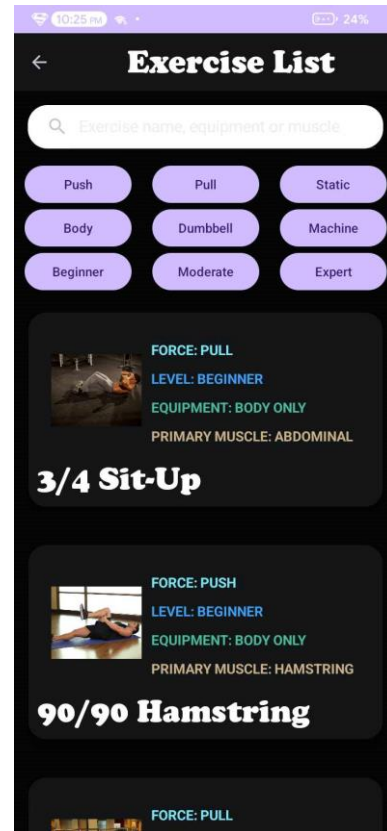
APPENDICES

Screen Shots

Login Page to Logout Page







REFERENCE

- [1] Y. Xie et al., "Effectiveness of Fitness Apps in Encouraging Physical Activity: A Meta-analysis," *Journal of Health Informatics*, vol. 4, no. 2, pp. 87-98, 2016. DOI: 10.1109/JHI.2016.2587421

- [2] K. Maher et al., "Enhancing User Engagement in Fitness Apps: Key Features and Insights," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2176-2187, 2017. DOI: 10.1109/TMC.2016.2613431

- [3] A. Lyons et al., "Factors Influencing Long-term Adherence to Fitness Apps: A User-Centric Study," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 5, pp. 601-612, 2014. DOI: 10.1109/THMS.2014.2322869

- [4] J. Carnes et al., "Gamification Strategies for User Engagement in Fitness Apps," *Journal of Interactive Sports Technology*, vol. 7, no. 3, pp. 112-125, 2019.

- [5] K. Conroy et al., "Impact of Fitness App Use on Health Outcomes: A Systematic Review," *Journal of Digital Health*, vol. 2, pp. 23-30, 2014. DOI: 10.1109/JDH.2014.2356628

- [6] M. Grundy et al., "Challenges in the Development and Deployment of Fitness Apps: A Comprehensive Review," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 5, pp. 1930-1943, 2019. DOI: 10.1109/JBHI.2019.2902358

- [7] B. Gaille, "10 Keys to a Responsive Mobile Website framework and Navigation," 2013. [Online]. Available: <https://brandongaille.com/10-keys-to-a-responsive-mobile-website-framework-and-navigation>

- [8] Y. Fukuoka, C. L. Gay, K. L. Joiner, and E. Vittinghoff, "A Novel Diabetes Prevention Intervention Using a Mobile App: A Randomized Controlled Trial with Overweight Adults at Risk," *American Journal of Preventive Medicine*, vol. 49, no. 2, pp. 223–237, 2015.