WELLFIT: A SMART FITNESS COMPANION FOR PERSONALIZED WELLNESS

LEE LOK YEE

UNIVERSITI TUNKU ABDUL RAHMAN

WELLFIT: A SMART FITNESS COMPANION FOR PERSONALIZED WELLNESS

LEE LOK YEE

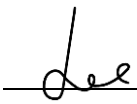
A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

September 2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Lee Lok Yee

ID No. : 2200488

Date : 12/9/2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **" WELLFIT: A SMART FITNESS COMPANION FOR PERSONALIZED WELLNESS"** was prepared by **LEE LOK YEE** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Software Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : Faranak Nejati

Date : 12/09/2024

Signature :

Co-Supervisor :

Date :

ACKNOWLEDGEMENTS

First, I want to express my deepest gratitude to my supervisor, Dr. Faranak Nejati, for her assistance and mentorship, which have been very helpful throughout the journey. Her steadfast assistance, perceptive criticism, and unceasing encouragement have shaped my effort and enhanced my own development. She was sympathetic and patient with me, especially when it came to my last-minute contributions. Her belief in my potential has been a great source of inspiration.

I would also like to extend my heartfelt appreciation to my mother, whose unwavering love with support, and never-ending encouragement have been my pillar of strength through all of life's setbacks and triumphs. Her confidence in me has always given me strength, even in the moments when I doubted myself. Without her, none of this would have been possible.

I am deeply grateful to my friends for their willingness to lend a helping hand that provides me with emotional and moral support.

Finally, I would want to thank all of the amazing people who have impacted my life in any manner. This journey has been richer and more important because of the compassion, support, and presence provided.

ABSTRACT

In the era where fitness is still a popular workout trend, it has become a daily routine for many people. The increasing demand of workout has contributed to the growth of workout gym equipment and an expanding variety of workout solutions. Nevertheless, the current fitness mobile application often lack a truly wide workout variety in the middle of expanding fitness industry and demand as fitness approaches grew. Hence, presenting a mobile fitness application that offer features based on user requirements is very important. This application provides a wide variety of workout solutions that offers clear instructions, provide insightful information on nutritional intakes, progress tracking and workout libraries plan. In addition, the application also incorporates BMI and Calories calculator to assist user in achieving health goals. The application features an AI gym equipment recognition, seeing how crucial user experience is in mitigating user risk and promote awareness of workout varieties by detect gym equipment accurately and provide guidances on how to use the detected gym equipment correctly. In conclusion, this project combines a wide range of features together with AI powered support, thereby increasing user satisfaction and experiences.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
MCO	Movement Control Order
UI	User Interface
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
JSON	Javascript Object Notation
CRUD	Create, Read, Update, Delete
BMI	Body Mass Index
RAD	Rapid Application Development
HICT	High Intensity Circuit Training
WBS	Work Breakdown Structure
SQL	Structured Query Language
UAT	User Acceptance Testing
SUS	System Usability Scale

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Many years ago, fitness wasn't that popular and only enough to maintain the overall health of individual. As the year passes, people become more into the fitness not just for health measures but also wants to get improvement on the physical appearance and body images. Hence, that is how the fitness embrace this demand and trend. As fitness approaches grew, fitness centre and industry emerged, which offering a technological advancement like platform to make users accessible and engageable to the fitness companion features, enhancing a convenient and enjoyable wellness journey. Today, fitness had contributed to the tremendous growth, highlighting the importance of wellness where 1 in 5 adults take part in fitness each day. Studies have also shown that actively engaging in fitness physically can to improve the brain health, help manage weight, lower the risk of disease, strengthen bones and muscles, and enhance the ability to do daily activities (Centers for Disease Control and Prevention, 2023). Thus, the influence of fitness on people's life is apparent with such prevalent engagement and health advantages.

Besides that, the fitness trend was accelerated after Malaysia's Movement Control Order (MCO) was put into place during the pandemic outbreak. This is due to the strict guidelines for social distancing and physical stores as well as business deemed unnecessary were told to cease operations. Hence, online fitness platform start to emerge, harnessing the trend to provide a tailored solutions for personal wellness that leads to the efficiency and effectiveness of the paradigm shift as well as users experiences. As a result, it had affected the lifestyle as well as the acceptance level towards the paradigm shift of digital transformation.

In response to the trend, the fitness application have been made available in the App Store or Google Play Store in which user may download these apps from their smartphones and utilize the features and tools as an online platform for their fitness companion to customize their wellness. However, problem occur when user do not have knowledge on the gym equipment type

where their fitness journey do not meet up to their satisfaction and expectation if wrong equipment tools were used. Additionally, users may find it difficult to use the gym equipment correctly in accordance with their own wellness solutions, which increases the risk of injury.

Consequently, a fitness application featuring an entertaining and intuitive UI has been designed incorporating all of these capabilities and functionalities. This application includes a broad selection of numerous fitness workout routines, nutritional intakes, BMI calculator and calories tracking. In addition to response to the previously mentioned problem as above, the application now features an AI gym equipment recognition in order to detect gym equipment accurately and provide guidances on how to use the detected gym equipment correctly.

1.2 Importance of Study

Fitness become a necessity in the modern period, not just for health reasons but also to improve their physical appearance and reduce stress (Jain, 2017). Thus, the fitness industry has responded to this trend and needs by providing a platform that leverages technological advancements to make fitness companion features more accessible and engaging for users, ultimately resulting in a more comfortable and pleasurable journey toward well-being. Hence, the development of WellFit mobile application highlights how important the study in meeting changing trends and needs. WellFit mobile application offers rich features that have greatly aided in technologies, and its impact on people's lives is evident from its widespread use and positive health effects.

WellFit undergo a thoroughly grasp of data analytics to capitalize and analyse on the trend and aims to offer customized wellness solutions by fusing workout recommendations, personalized workout routines, nutritional intake guidance, and dietary calculators that seamlessly improve user experiences. Additionally, the project also demonstrates how the importance of study enhance the application with cutting-edge technologies by featuring an AI gym equipment recognition which designed to achieve the goal of accurately detecting gym equipment and provide guidance on how to use the detected gym equipment correctly.

Last but not least, the project had advantages society including individuals. WellFit offers customizable workout plans, nutritional guidance, and progress tracking that improve users' fitness. Through this app, individuals can minimize the frequency of visiting to fitness centre and needs of seeking a trainer. By integrating the AI gym equipment recognition features, individuals can identify the unfamiliar equipment and ways to use them.

All in all, this project serves as an excellent example in meeting and bringing change in society and industry needs, enhance users experience in the journey of fitness and encourage healthier lifestyles through all the features implemented.

1.3 Problem Statement

This section define problem statements based on an analysis of the existing mobile application.

i. Absence of AI gym equipment recognition

According to Statista Market Insights(2023), the market for image recognition technology is expected to increase by around 10% anually and reach a value of nearly $21 billion by 2030. Hence, AI recognition has grown in significance and widely accessible by solidified its position across various industries. According to the literature review, the top ten highly rated fitness mobile applications from the Play Store were chosen for further analysis, review and discussion on the features and areas of improvement for the development of the project. Nevertheless, despite the technological advancement of AI recognition, all of the mobile fitness applications features that are discussed in Chapter 2 do not implement such AI recognition features. Hence, it is seen that the absence of the AI recognition features is a shortfall among the comparison of existing mobile application and had bring further issue in the mobile fitness applications.

ii. Safety Issue

Without AI gym equipment recognition, it limits user's ability to recognize various gym equipments and provide proper instructions on the gym equipment, especially for beginner users who may not have extensive knowledge of gym equipment. Consequently, this noteworthy absence poses a significant shortfall

on inappropriate and acquainted with the utilization of gym equipment. Users will results in an increase of accidents and injuries risk due to the utilization of wrong equipment with unskilled adjustment to fitness level, incorrect posture and instructions when doing exercise routines. Users may not be aware of their own limitations and may overweight themselves during workouts. This safety issue not only threaten users safety, discourage users from working out regularly, it also resulted user in negative user experience towards utilizing the fitness application. Moreover, it is challenging for user to remember various type of gym equipment together with the instructions. Instead of having to memorize every piece of equipment in the gym, users may use AI gym equipment recognition to recognize the name of the equipment and get the necessary instructions to reduce the risk of injury while exercising. As a result, this can improve user experiences while reducing safety risk.

iii. Limited Workout Variety

There are various types of gym equipment in the fitness industry where some gym equipment name and appearance might be unfamiliar and unknown to user. Without AI gym equipment recognition, users may be limited in their ability to explore different forms of gym equipment as each type of equipment will produce a different workout and exercises. Hence, users will just stick to familiar and the same equipments and exercises over and over. Moreover, user will not explore and get introduced to different workouts that comes with different target body areas and instructions. This will resulted in lack of variety in user's fitness regimens that cause boredom and make them less enthusiastic in the fitness journey. In addition, user will have stagnant growth in their workout skills and levels due to limited workout variety caused by the absence of AI gym equipment recognition to guide user in new workouts and equipments.

Hence, incorporate AI gym equipment features is an intelligent strategy that genuinely improves users experience and engagement within the mobile application.

1.4 Project Objectives

1. To identify the features and functionalities to be incorporated into the mobile fitness application.
2. To integrate AI gym equipment recognition into the mobile application to enhance user safety by providing information regarding each equipment, thereby increasing user's awareness and understanding of their usage.
3. To enhance user workout skills by incorporating features into the mobile application that promote awareness of workout varieties.
4. To conduct thorough testing and evaluation of the integrated AI gym equipment recognition system within the mobile application to assess its user satisfaction.

1.5 Proposed Solution

A mobile fitness application is constructed by referring to the problem statement as a guide to provide users with an extensive range of exercises and workouts for fitness purposes, either at gym or home. The application will also include workout library, information, calculators and reports on related fitness topics. Moreover, an AI gym equipment recognition feature is incorporated to offer user the ability to recognize gym equipment and make use of it. Hence, it has expands the knowledge of user towards gym quipment and also enhance user's engagement within the application.

The system architecture of the mobile application is displayed as below. The user interface is created using React Native which can be used in the mobile application's screen, navigation element, forms and user input. React Native components will use HTTP request to communicate with the backend server. Upon executing the task, such as user sign in or fetching workout data, the React Native frontend initiates an HTTP request to the backend server. Once received, it will process and reply the response in JSON format. After handling the response, the React Native components will update the user interface accordingly.

This project employs two backend systems. Firstly, it uses Node.js as its primary backend for the standard features and functionalities. It makes the RESTful Api endpoint available to interact with the frontend. When the request

is forwarded to Node.js, it executes business logic such as Create, Read, Update, Delete (CRUD) operations, retrieves or updates data from the database and replies the response back to the client.

The second backend system named Flask is utilized for AI-related tasks in the project. When an AI task is required, Flask receives HTTP requests from the Node.js to process the AI-related business logic with Python and a pre-trained machine-learning model to generate the outcome of the AI recognition. In return, the model will provide the recognition result as a response back to the React Native frontend with the help of Node.js.

In this project, the choice for selecting database for the mobile application will be MySQL. To run the database, PhPMyAdmin will be used together with WampServer to operate the database. The database will interact with backend for the fetching or modifying data. Once done, the database will return the response back to backend, indicating whether the action was successful. The message will then be sent back to the React Native front end to display information to the user of the success or failure operation.

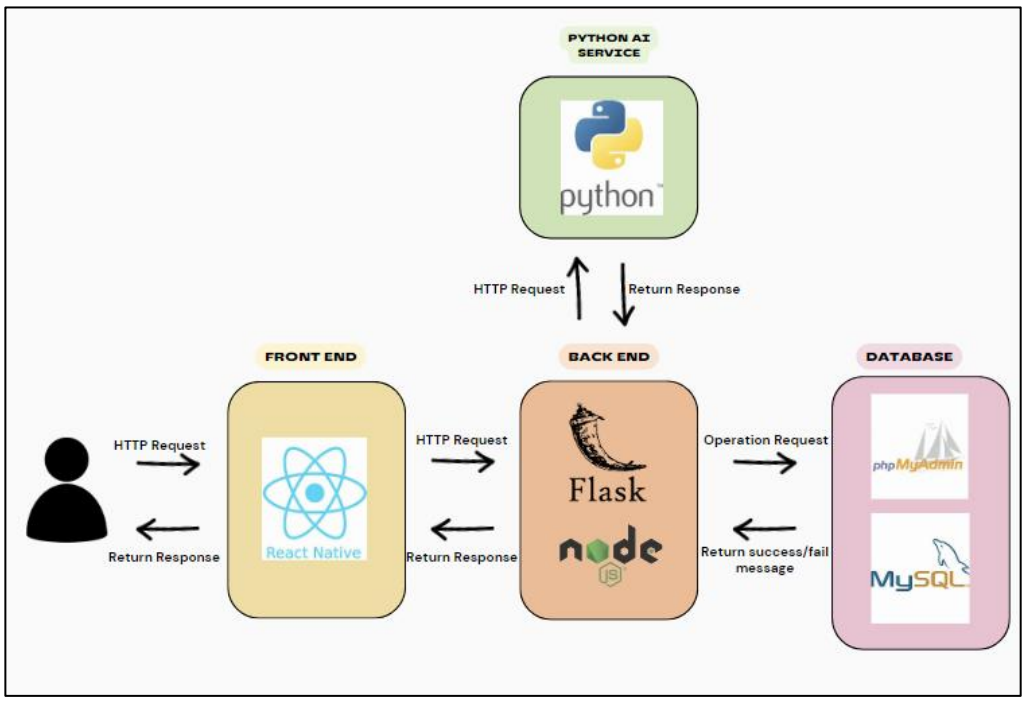


Figure 1.1: System Architecture

1.6 Proposed Approach

In this project, the prototype methodology will be used to develop the mobile fitness application with the integration of AI powered gym recognition

equipments. The figure below depicts the prototype methodology with each of its phase.

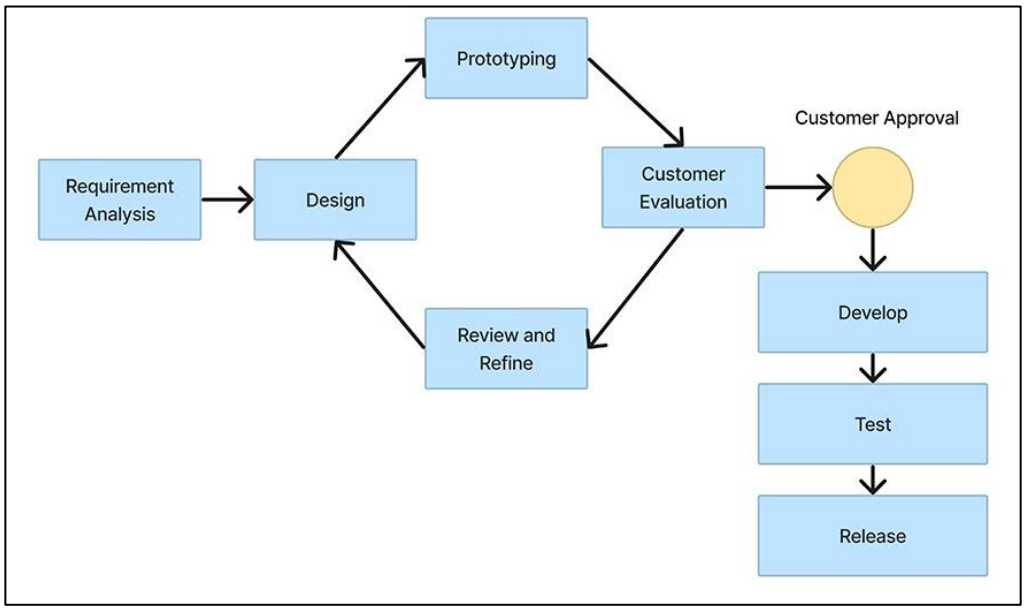


Figure 1.2: Prototyping Methodology (Anon,2022)

The prototyping methodology allow developers to build, test and rework on the prototype iteratively instead of creating a fully functional system. In the initial stage, a requirement analysis will be conducted to gather preliminary project specifications. The advantage of prototype methodology is its comprehensive analysis which can find possible issues before real development begins. Next, only essential features and functionalities will be incorporated into the project to create the initial prototype. Subsequently, users provide evaluation, feedback and suggestions to the initial prototype that has been built for greater enhancement and refinement during the second iteration. The iteration of adjusting the prototype are carried out repeatedly until users are satisfied with the prototype that fulfilled their requirements. The project then enters the development phase with the final prototype and undergo testing phase before being implemented as a final product. This prototype methodology not only outlines the software solutions dynamically, it also drives the project towards high efficiency and better caters to user's needs.

1.7 Project Scope

The project scope defines target user modules covered that are divided into main modules and basic modules. The main modules includes data input, workout filtering, workout recommendation, workout information, workout searching, workout customization, workout library, progresss tracking report, ai gym equipment recognition, calories calculator, BMI calculator and nutritional intake information while basic modules include login module and register module. This mobile application can access using mobile devices and the project output will be an customer-side interface.

1.7.1 Targeted Users

The target user of this mobile application will cater to a wide spectrum of fitness users, such as experts who aims to use application for advanced training and workout regimen. Also, beginner who try to start their fitness journey by utilising the application to learn basic workout and fitness lovers who are passionate and actively engaged in various types of workouts.

1.7.2 Modules Covered

1.7.2.1 Main Modules

i. Data Input

The data input module has incorporated a user's data input features, where it require users to input their details related to personal data information, experiences, goals, equipment used, body focus area, duration, frequency and so on. This module allows users to generate personalized workout solutions according to the input.

ii. Workout Filtering

The workout filtering module optimizes the best selection for users on their personalized workout solutions and enhances user experiences through filtering and sorting on the mobile application. Users can select the filtering features and categories to enable users to view their own filtered results. In addition, users can also quickly remove the applied filters with a single click of reset button.

iii. Workout Recommendation

The workout recommendation module will suggest related workouts to users after users provide data input. Users may easily accomplish their goals by clicking on the workout recommended by the module as the workout is customized to meet the user's specific needs.

iv. Workout Information

The workout information module has included a workout information section where users can find out all the detailed information and descriptions of the workout, including written workout instructions, motion guidance, sets, duration, equipment, intensity level and target areas. The comprehensive workout information allows users to stay focused and track their progress along the fitness journey.

v. Workout Searching

The workout searching module enables users to search what they are looking for by entering relevant terms. Next, the app produces a list of workouts with what has been typed in, facilitating user's search for what they need.

vi. Workout Customization

The workout customization module allows users to create workouts from stratch according to their personal taste by adding exercise name, description and selection on parameters. Upon adding workout inputs, the application will generate and display a new workout to users according to their customization. Moreover, users can edit and delete their customized workouts if they wish to keep their fitness progress organized and focused on current goals.

vii. Workout Library

The workout library module incorporates a wide range of workouts such as individual workouts with a wide variety for every fitness level. This allows users to approach with many kinds of workouts and hence guarantees a well-initiated fitness journey.

viii. Progress Tracking Report

The progress tracking report module keeps track of the user's workout progress over time and provides report information. This allows users to monitor the development and results so they can stay motivated along their workout journey.

ix. AI Gym Equipment Recognition

The AI gym equipment recognition module allows users to identify and accurately detect the gym equipment and provide guidance on how to use the detected gym equipment correctly. This module eases the workout journey for users, especially beginners who do not have much knowledge on the fitness area, which leads to improved user experiences.

x. Calories Calculator

The calories calculator module allows users to provide data input like age, gender, weight and height to calculate their calories need per day. Hence, this module allows users to balance their diet that align with health goals and lifestyle.

xi. BMI Calculator

The BMI calculator module allows users to determine their body composition and be categorized based on metrics by entering data such as age, gender, weight, and height. Hence, the module helps users reduce health risks and appropriately control their weight based on their BMI index.

xii. Nutritional Intake Information

The nutritional intake information module provides users a valuable information of the daily recommended intake for nutrients. This module can also act as nutritional awareness for users that have specific conditions such as diabetes on the types and quantities of nutrients they are consuming.

xiii. User Profile

The user profile module provides users with their personal information like name, email address, age and gender. The module also shows the user's data

inputs for creating a personalized workout recommendation such as goal, level, frequency, duration, focus area and tools. In addition, users are allowed to edit the data inputs to change their fitness objectives and routines.

1.7.2.2 Basic Modules

i. Login Module

The login module enables users to login to the mobile application after registration with email address and password.

ii. Register Module

The register module enables first time users to register their account by providing credentials like name, email address, password, gender and age.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In order to acquire data and information on comparable present-day applications, software system development, methodologies and framework, a literature review is done in this chapter. The focus of this study is to pinpoint what is the approaches to develop a fitness app that meets the needs of users in the end.

2.2 Similar Existing Mobile Applications

In this capter section, there are 4 similar mobile applications that are examined, analyzed and studied based on their features, functionality and services. The 4 comparable fitness applications are Fitbod Workout and Gym Planner, 30 Day Fitness -Home Workout, Fiit and Fitness Buddy: Gym Workout Log were chosen due to each of their advantages and drawbacks which could be analysed further in order to provide a more visible and clearer image.

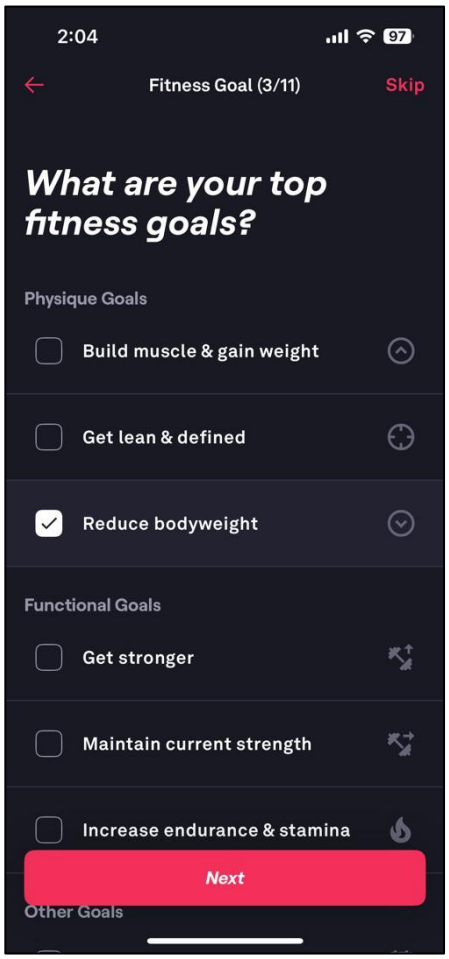
2.2.1 Fitbod Workout and Gym Planner

Fitbod is a fitness app with vast library of workouts designed to work with various kinds of equipments. Fitbod is being recognised with making consistent workout by create and refine highly customized exercises to help you improve with each training session. In an effort to broaden the app's utility, Fitbod has created a well designed application where each training session is custom-built for the fitness level, goals and equipments. Below is the outline of the key features of the Fitbod fitness app along with an explanation of each. These features include:

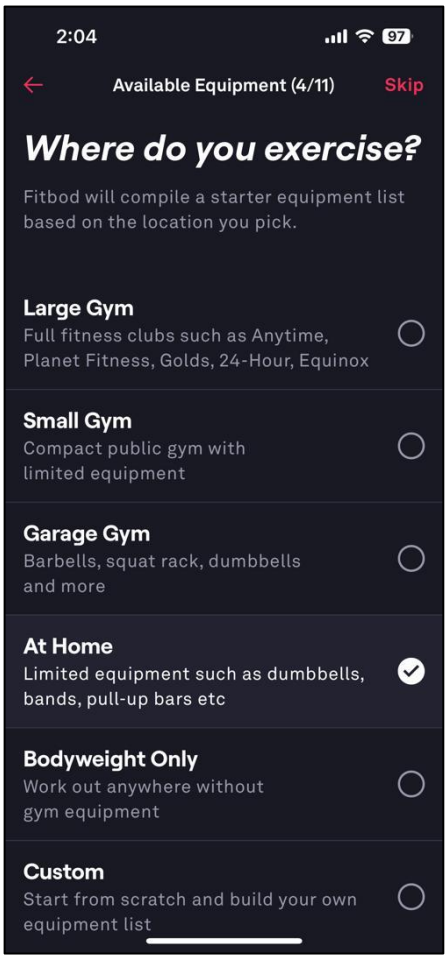
i. Data Input
ii. Workout Filtering
iii. Workout Recommendations
iv. Workout Information
v. Workout Customization
vi. Progress Tracking Report

i. Data Input

Fitbod app has incorporated a user's data input features, where it requires user's to input workout frequency, experience level, fitness goals, workout location, equipment used and body focus area. This features gather user's input and generate a personalized workout solutions that suits best for them. The figure below depicts the user's data input acquired by the app.



(a)



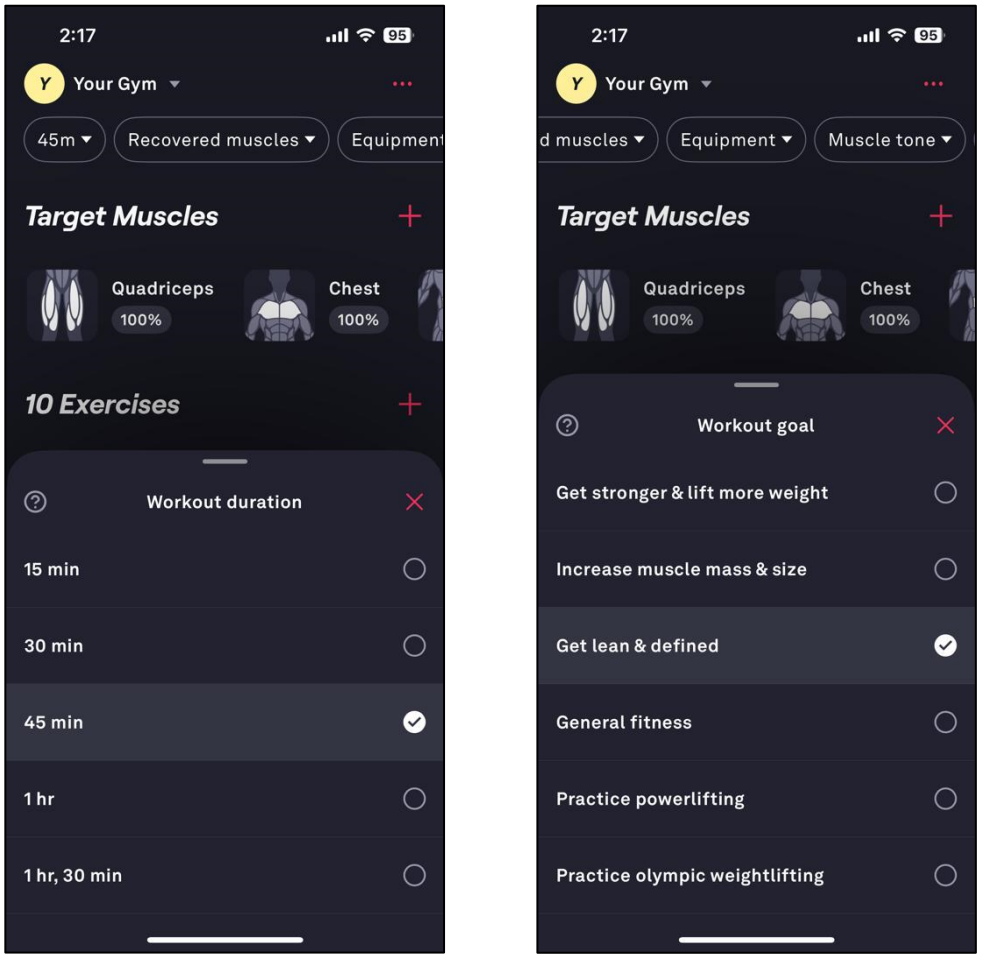
(b)

Figure 2.1(a)(b): User's data input

ii. Workout Filtering

Fitbod offers the workout filtering features, where users can filter and sort the time duration for workout, type of muscle splits, equipment used, bodybuilding workout goal, workout difficulty, exercise type, warm-up and cool-down routine. This features optimizes the best selection for users on their personalised workout solutions. Once workout filtering features are on, it enables users to

view their own customized plans based on their preferences. The figure below shows the workout filtering features that are available in the Fitbod app with workout time duration and workout goals applied.



(a) (b)

Figure 2.2(a)(b): Workout filtering page with duration and goal filter applied

iii. Workout Recommendations

On Fitbod app , after users provide data input , the app will come out with workout recommendation according to user target muscle , as seen in the figure below, where the app will prescribe abs workouts when a user choose abs as their target muscle during the data input process.

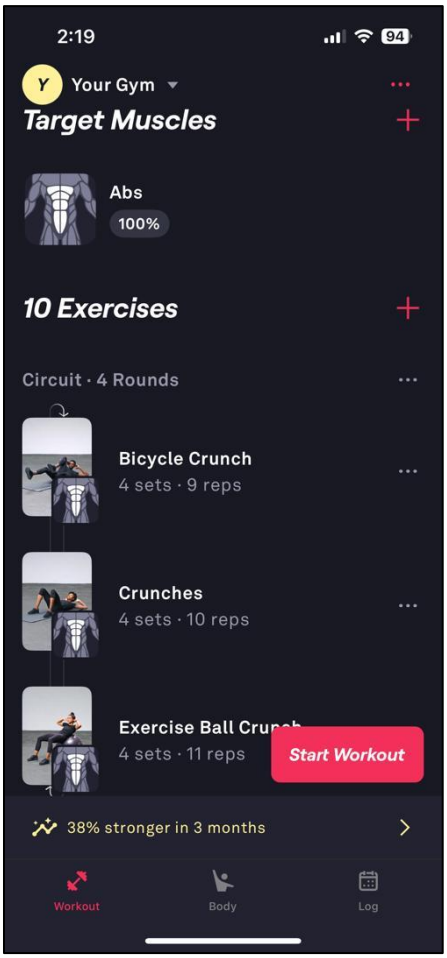
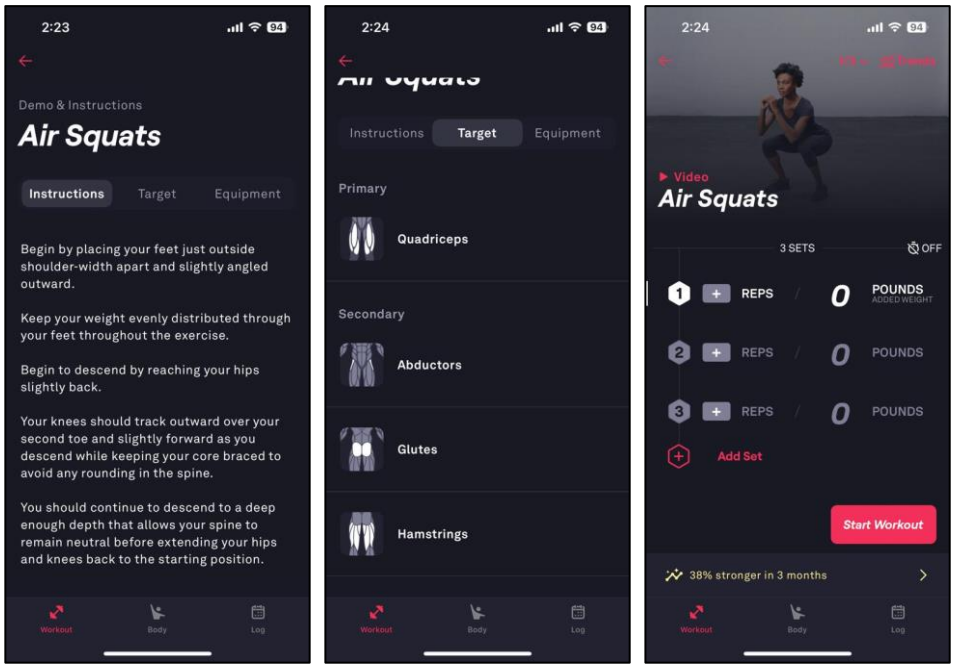


Figure 2.3: Workout recommendation on abs as target muscle

iv. Workout Information

The app has included a workout information section dedicated to the workout information where users can find out all the detailed informations and descriptions of the workout, including workout instructions, videos, primary and secondary targets, equipment used, number of sets, weight required and a timer for users to stay focused and track their progress along the fitness journey. Figure below illustrates the detailed information of a workout session.

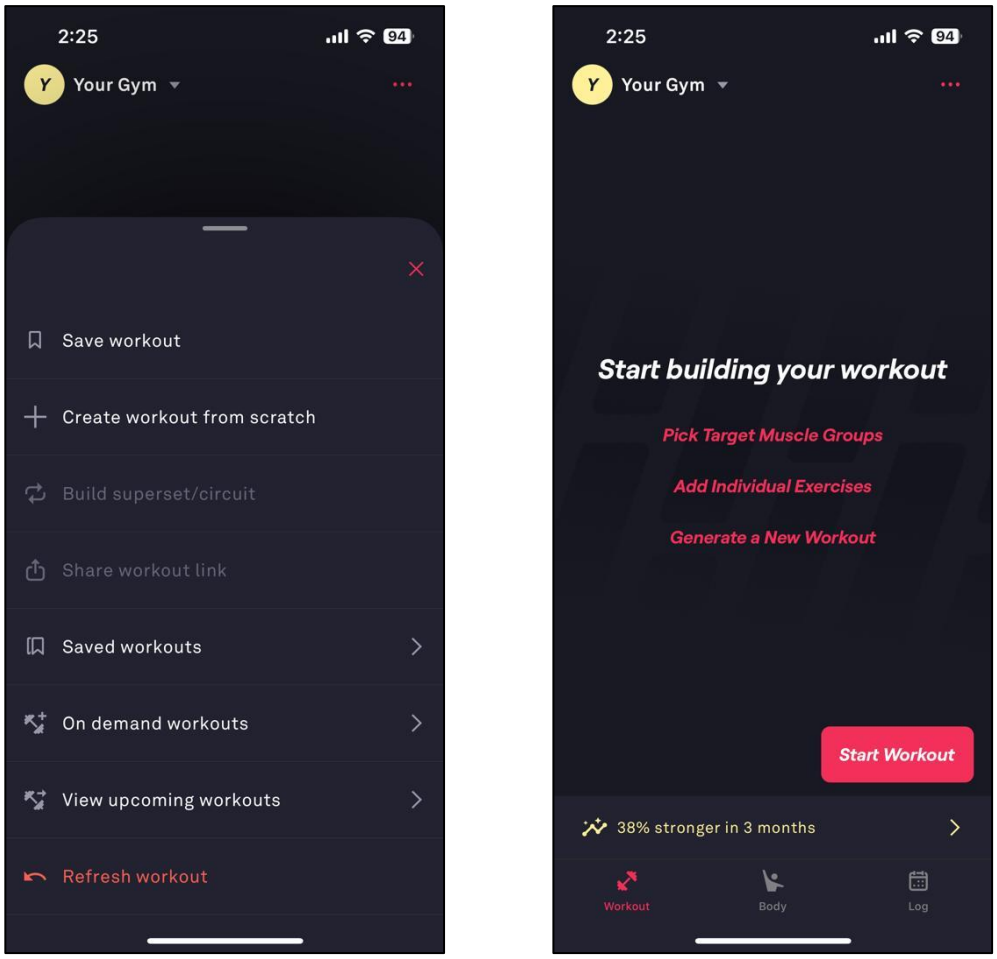


(a) (b) (c)

Figure 2.4(a)(b)(c): Workout information page of a workout session

v. Workout Customization

Fitbod allows users to create workout from stratch where they can customize their workout by picking the muscle group, add individual exercises and generate a new workout according to their own personal taste , as seen in the figure below.



(a) (b)

Figure 2.5(a)(b): Workout customization of app

vi. Progress Tracking Report

The Fitbod app provides extensive details on each workout. As seen in the figure below, it keeps track of user's workout progress over time and provides information on number of workouts, weekly goals and current streak. Users can also monitor the development and results from the calendar , where they can stay motivated along their workout journey. Moreover, it displays the past workouts such as date, type of workout, volume, number of exercises you've accomplished in addition to the amount of calories burned, as illustrated in the figure below.

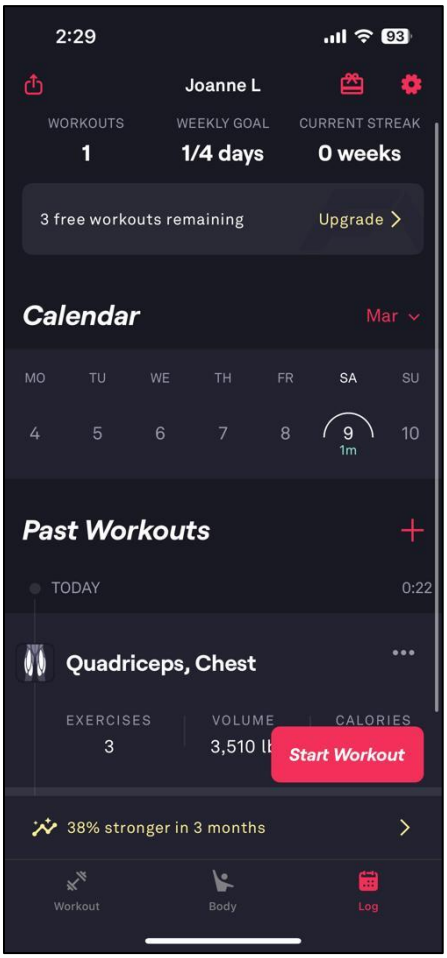


Figure 2.6: Progress tracking report details

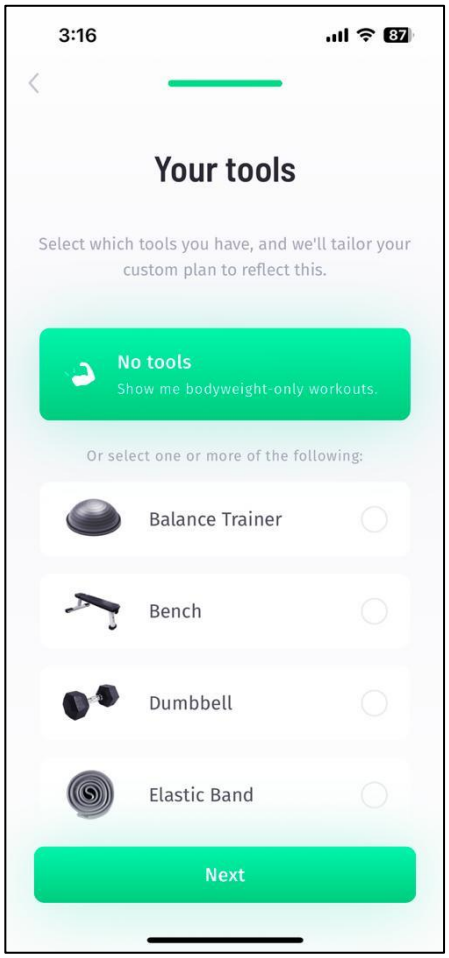
2.2.2 30 Day Fitness-Home Workout

The main focus of 30 Day Fitness , which offers the workout plan that suits and caters to user's need, is emphasize on seeing desired results within 30 days. 30 Day Fitness personalize the plan according to user's data input to generate personalized goals, workout duration and exercise sets. With the help of this fitness app, users can train conveniently at home or at any other convenient location whenever they want. The following is the key features of the the fitness app along with the explanations:

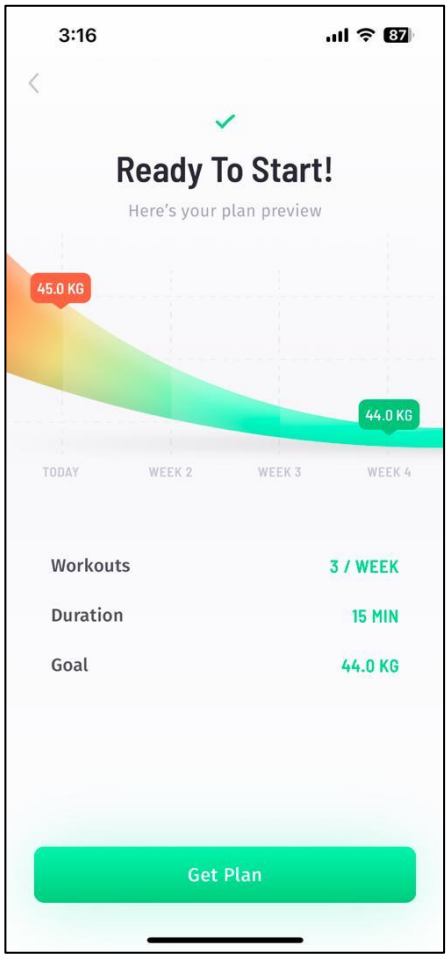
i. Data Input
ii. Workout Filtering
iii. Workout Recommendation
iv. Workout Information
v. Workout Library

i. Data Input

30 Day Fitness offers user's data input features, where it acquires user's name, age, weight, height, gender, desired weight, body area for workout, desired workout goal, activity level, workout day and workout equipment tools. This features gather user's input and generate a customized workout solutions that suits best for them. The data input features where users enter their information to create a personalized workout solutions are shown in the image below.



(a)



(b)

Figure 2.7(a)(b): User's data input to generate personalised workout solutions

ii. Workout Filtering

30 Day Fitness provides the workout filtering features, where users can filter and sort variety of workouts and exercises from the workout library. Users can filter options like body part focus, time duration for workout, workout difficulty

level, equipment used and home-friendly features to browse and view the filtered workouts based on their own preferences. Users might quickly remove the applied filters with a single click in addition to this capability. The figure below shows the workout filtering features that are available in the 30 Day Fitness app with body part focus and workout difficulty level applied.

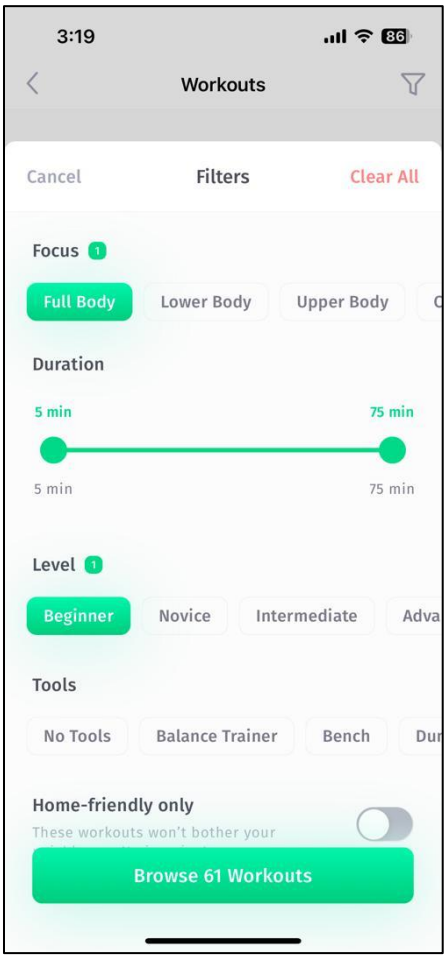


Figure 2.8: Workout filtering page with body part focus and workout difficulty level filter applied

iii. Workout Recommendation

On 30 Day Fitness app, after users provide data input , the app will come out with workout recommendation according to user goals, as seen in the figure below, where the app will prescribe lower body workouts when a user choose lower body as their body part focus during the data input process.

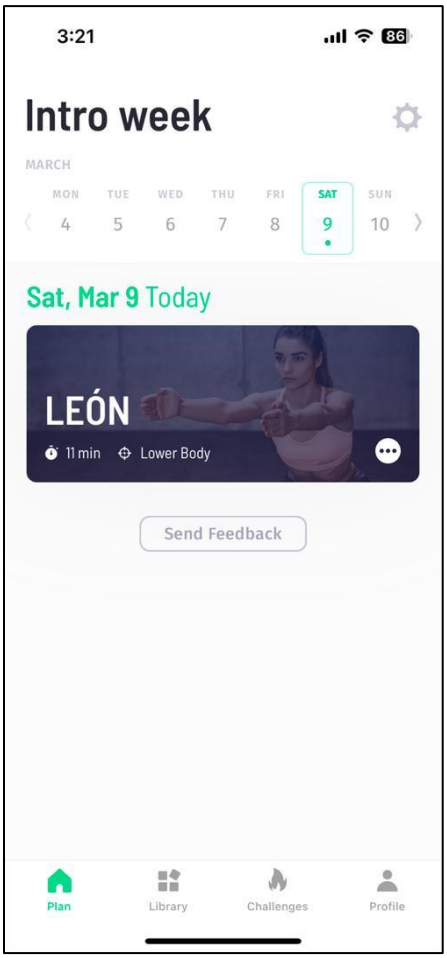
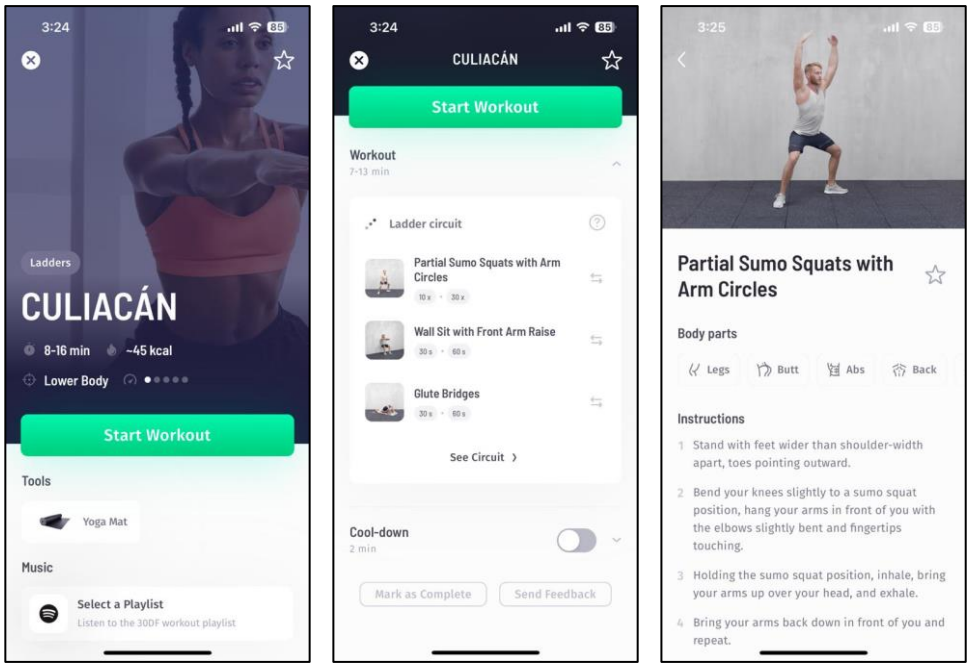


Figure 2.9: Workout recommendation on lower body as body part focus

iv. Workout Information

The app has integrated a workout information section where users can know the detailed informations and descriptions of the workout, such as workout instructions, videos, any equipment used, time duration , calories burnt, number of workouts set and body parts involved to make progress effectively in workout session. With this, it is easier for users to make decisions along their fitness journey. Figure below illustrates the detailed information of a workout session.

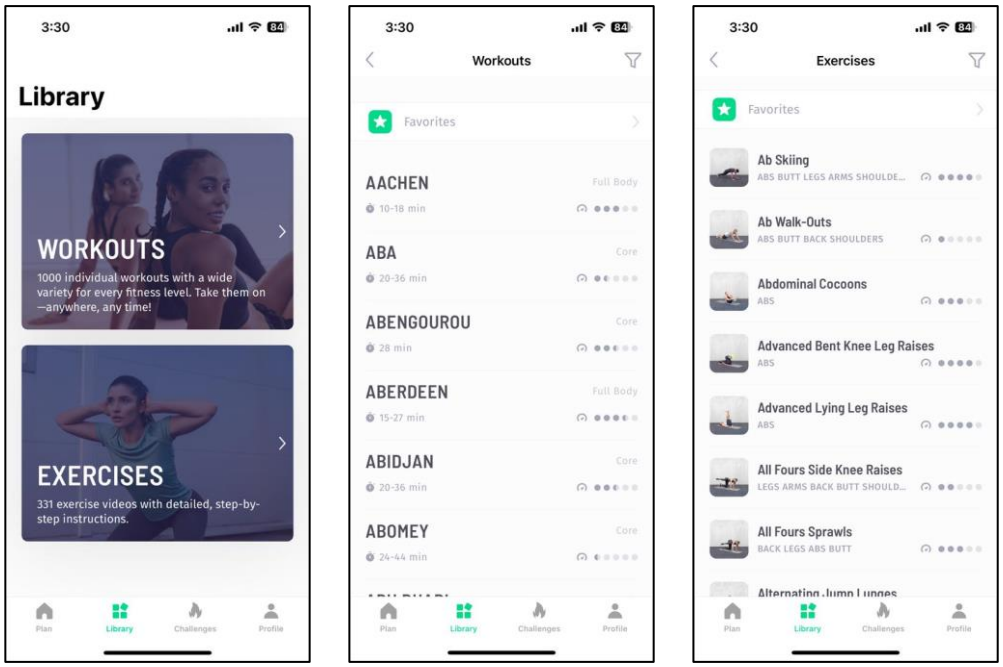


(a) (b) (c)

Figure 2.10(a)(b)(c): Workout information page of a workout session

v. Workout Library

The app library incorporates a wide range of workouts, containing 1000 individual workouts with a wide variety for every fitness level and 331 exercises videos with detailed, step-by-step instructions. Moreover, it offers 30-Day Challenges with in-depth workouts that allows users check in their progress day by day and guarantee a well-initiated fitness journey. The app extensive workout library is shown in the figure below.



(a) (b) (c)

Figure 2.11(a)(b)(c): Workout library page of the app

2.2.3 Fiit : Workouts & Fitness Plans

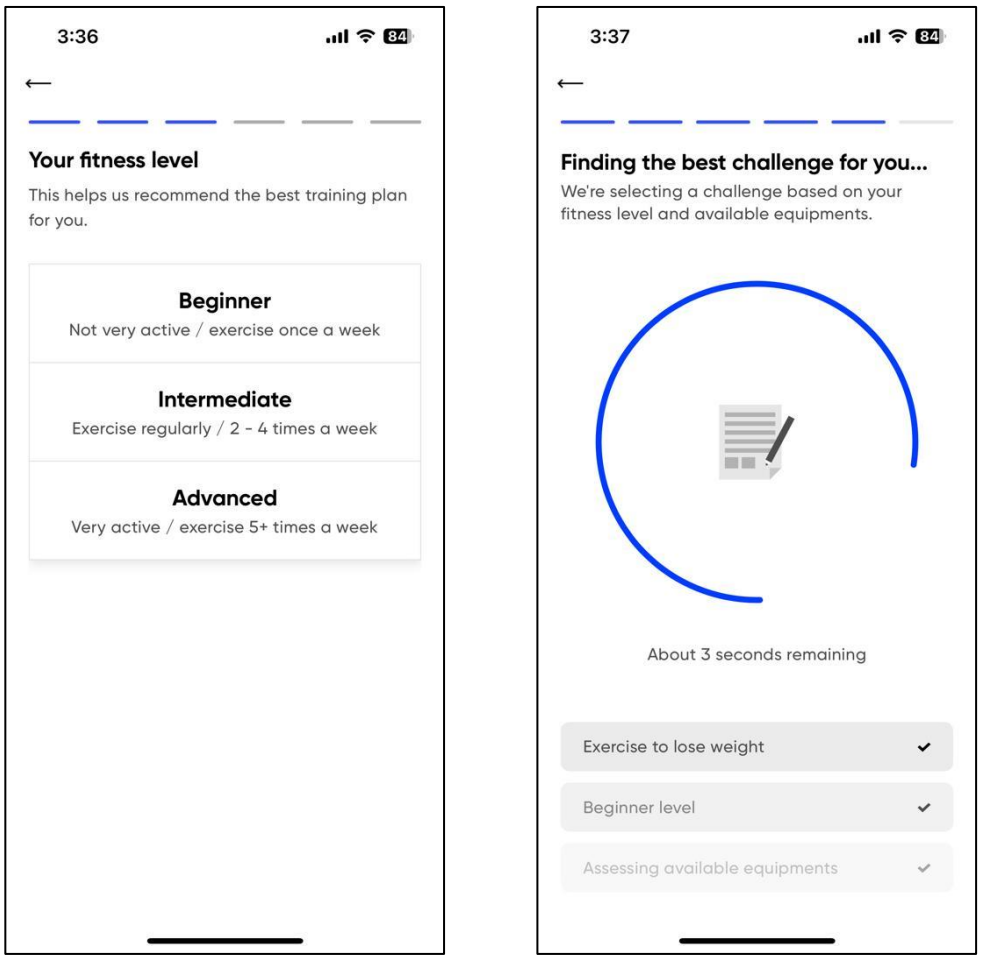
Fiit is an interactive fitness app that contains a comprehensive wide range of hundreds incredible workouts and well-structured training programs, all under the guidance of elite trainers. It is recognized for its outstanding video material of the workout, which includes highly qualified instructors that allows user to recreate the live atmosphere of workout class which enhancing user experiences. The key features of the fitness app enumerated along with the description are shown below:

i. Data Input
ii. Workout Filtering
iii. Workout Library
iv. Workout Information
v. Workout Searching

i. Data Input

Fiit app provides user's data input features, where it acquires user's name, gender, desired workout goal, fitness level and equipment used This features

gather user's input and generate a personalized workout solutions that suits best for them. The figure below shows the data input features where users provide the goals, fitness level and equipment used to generate customized workout solution.



(a) (b)

Figure 2.12(a)(b): User's data input to generate personalised workout solutions

ii. Workout Filtering

Fitt has implemented a workout filtering features, where users can filter and sort classes, plans and trainers according to the variety of factors like duration, workout completed, discipline, class level, target body part, target area and trainers. Next, users can click on apply to turn on the filtering features and view the filtered factors based on their own preferences. Users might quickly remove the applied filters with a single click on reset button in addition to this capability.

The Fiit app's filtering capabilities with duration, workouts and class level are displayed in the figure below.

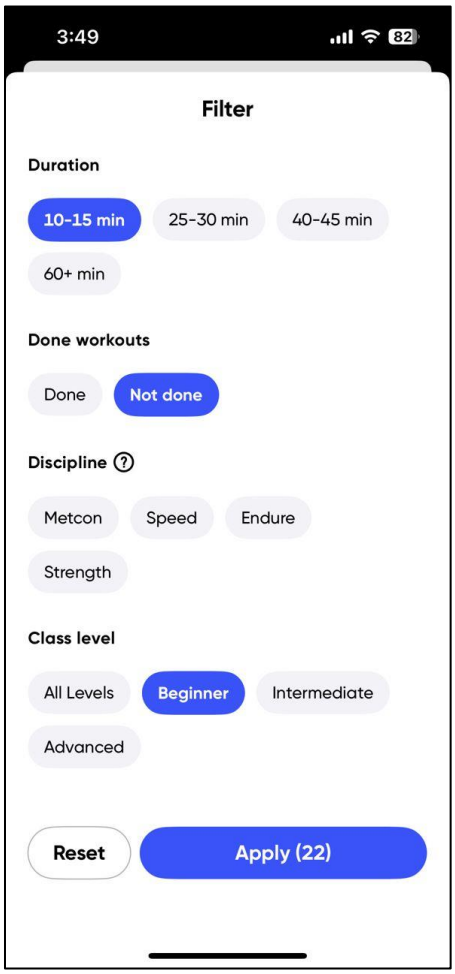


Figure 2.13: Workout filtering page with duration, workouts and class level filter applied

iii. Workout Library

The app has included a wide range of workouts, containing categories of classes and plans with guided trainers. This allows users to progress with their workout day by day under the guidance of trainers and guarantee a well-initiated fitness journey. The app's extensive workout library is displayed in the graphic below.

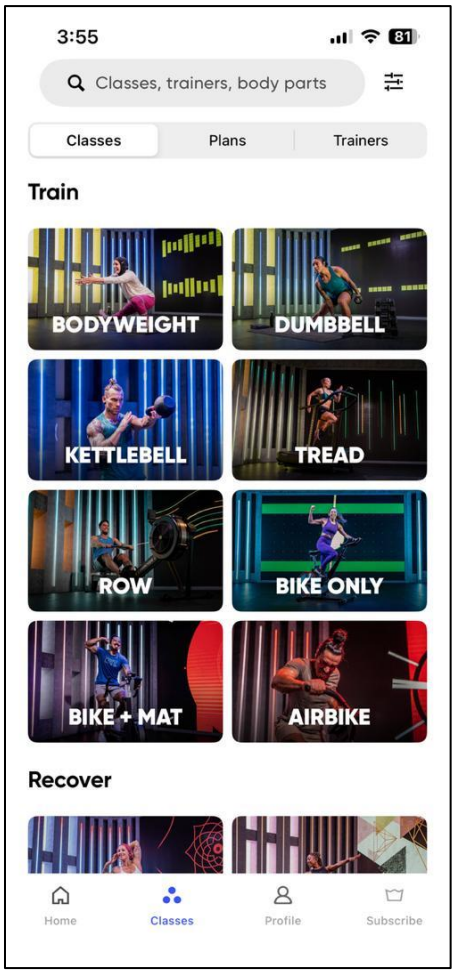
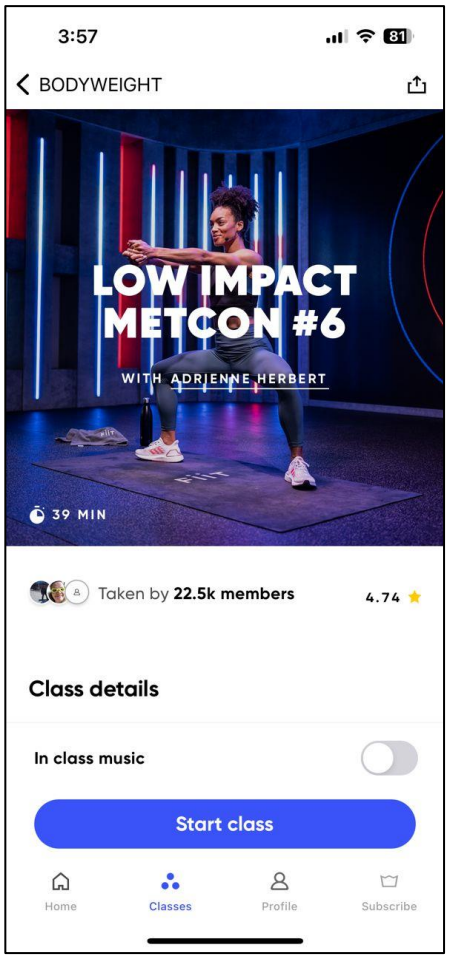


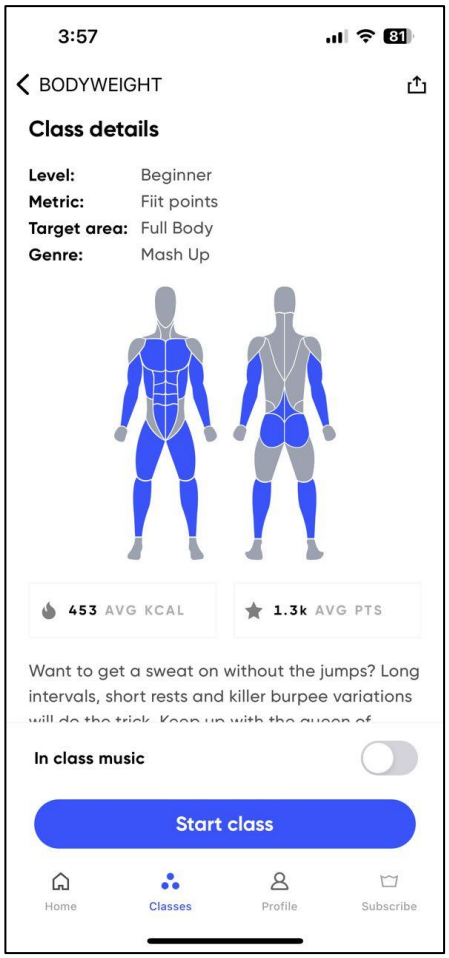
Figure 2.14: Workout library page of the app

iv. Workout Information

The app has integrated a wide range of workout information section where users can access comprehensive informations and descriptions about the workout class including class level, metric, target area, genre, average calories burnt, time duration, workout instructions and videos. Regarding training plans, users get to know the plans outcome, training plan level, time duration, workout instructions and videos. In addition, users get to know the trainers information that aids in decision-making as they progress toward fitness. Figure below illustrates the detailed information of a workout class.



(a)



(b)

Figure 2.15(a)(b): Workout information page of a workout class

v. Workout Searching

Fitt app offers a workout searching features that enable users to find the classes, trainers and body parts they're looking for more quicky and easily by entering relevant terms. Next, the app produces a list of all classes, trainers and body parts with what has been typed in, facilitating user's search for what they need. A figure below illustrates how typing the trainer name "Alex" will reveal a list of Alex Crockford's trainer classes.

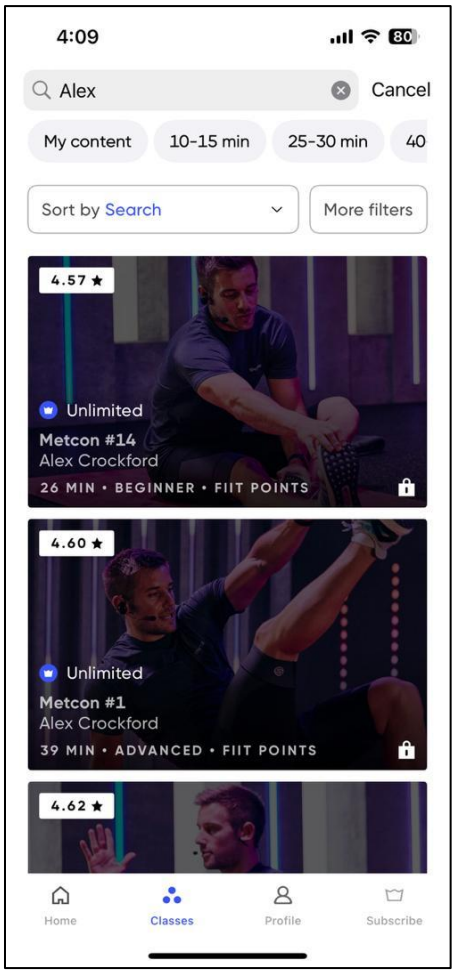


Figure 2.16: Workout searching page with relevant searching keywords "Alex"

2.2.4 Fitness Buddy: Gym Workout Log

The Fitness Buddy app focuses on the ultimate fitness transformation by providing an all-in-one workout and nutrition plan that promotes lean body mass and improves overall health according to user's goals. Additionally, individuals can effortlessly train anytime they want at home or at any other suitable location. Users can achieve convenience, maintain motivation during their fitness journey, and improve experience with the aid of this fitness app. The key features of the fitness app are as follows, accompanied by explanations:

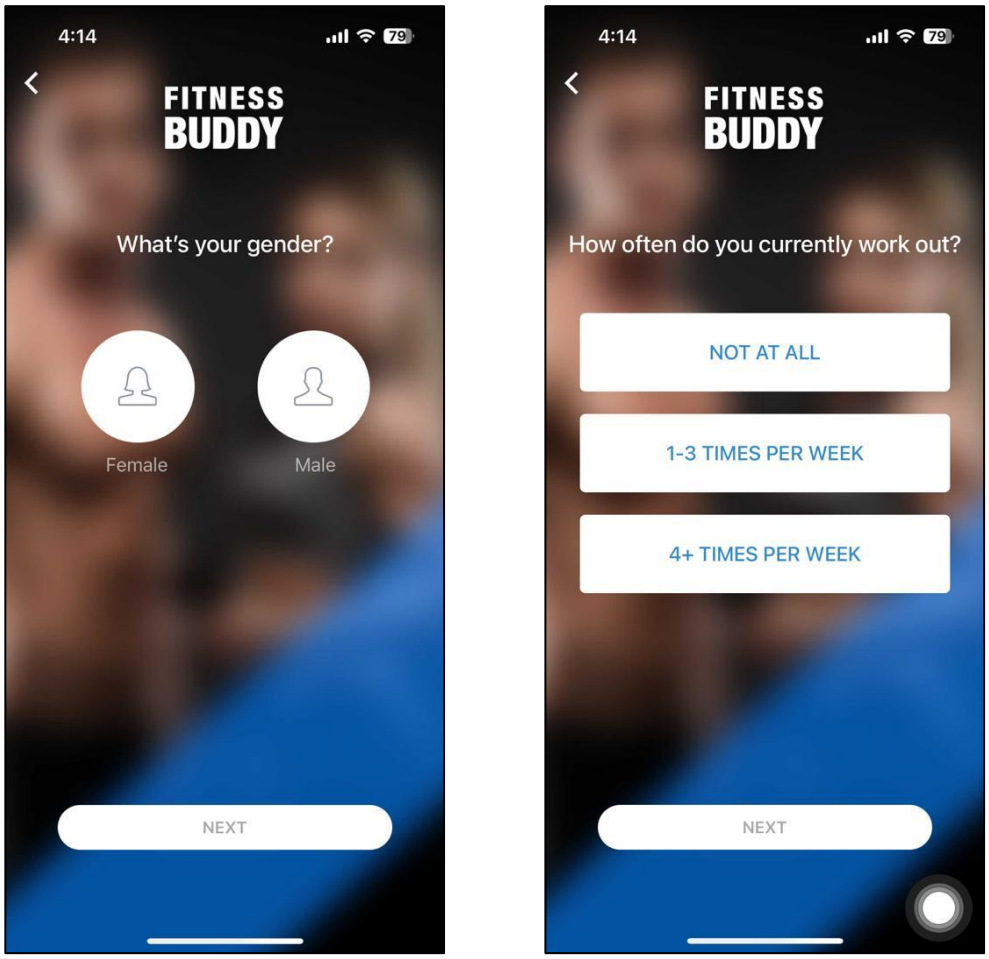
i. Data Input
ii. Workout Recommendation
iii. Workout Information
iv. Workout Library

v. Workout Searching

vi. Workout Customization

I. Data Input

Fitness Buddy app provides user's data input features, where it acquires user's name, gender, workout frequency, workout location, desired workout goal and focus body area This features gather user's input and tailor a personalized workout solutions that is perfect for the users according to their specific needs. The figure below shows the data input features where users provide the gender and workout frequency to generate customized workout solution.



(a) (b)

Figure 2.17: User's data input to generate personalised workout solutions

II. Workout Filtering

Fitness Buddy has incorporated a workout filtering features, where users can filter and sort body part, equipment used and difficulty level such as lower body,

chest, dumbbell, beginner and so on. Next, users can click on apply button to browse and view the filtered options according to their own personal taste. In addition, users can click on cancel button to quickly remove the applied filters. The Fitness Buddy app's filtering capabilities with body part and equipment used are applied and displayed in the figure below.

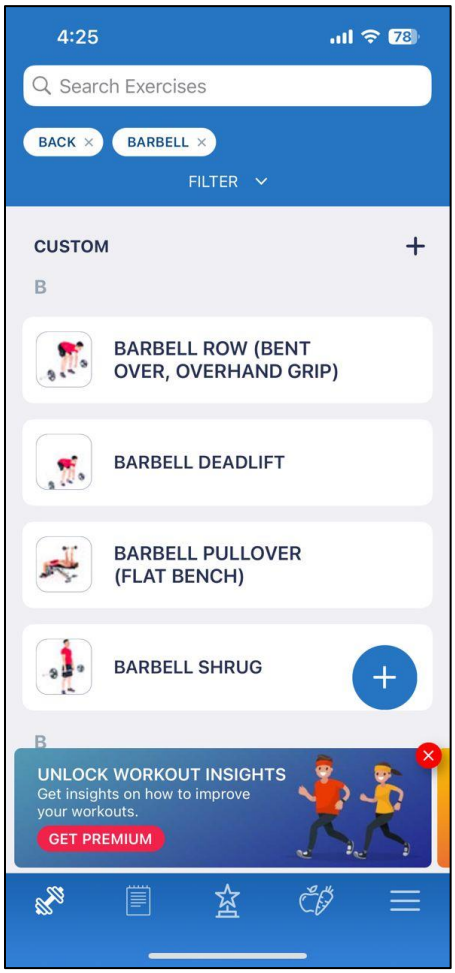


Figure 2.18: Workout filtering page with body part and equipment filter applied

III. Workout Information

The app has integrated a wide range of workout information section where users can access comprehensive informations and descriptions about the workout such as workout name, muscles affected, equipment used and instructions. Figure below illustrates the detailed information of a barbell row workout.

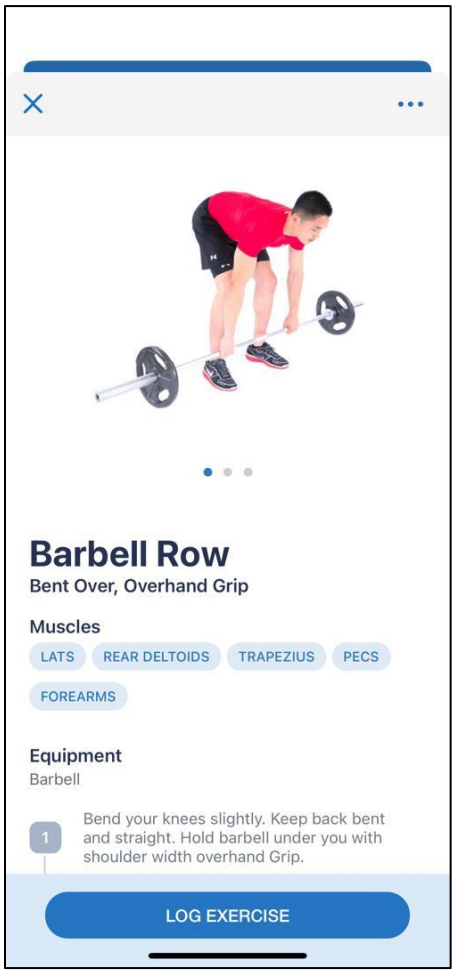


Figure 2.19: Workout information page of a barbell row workout

IV. Workout Searching

Fitness Buddy app offers a workout searching features that enable users to search for exercises they're looking for more quicky and easily by entering relevant keywords. Next, the app will produces a list of exercises with what has been typed in, facilitating user's search for what they need. A figure below illustrates how typing the keyword name "body" will reveal a list of exercises that contains the keyword name.

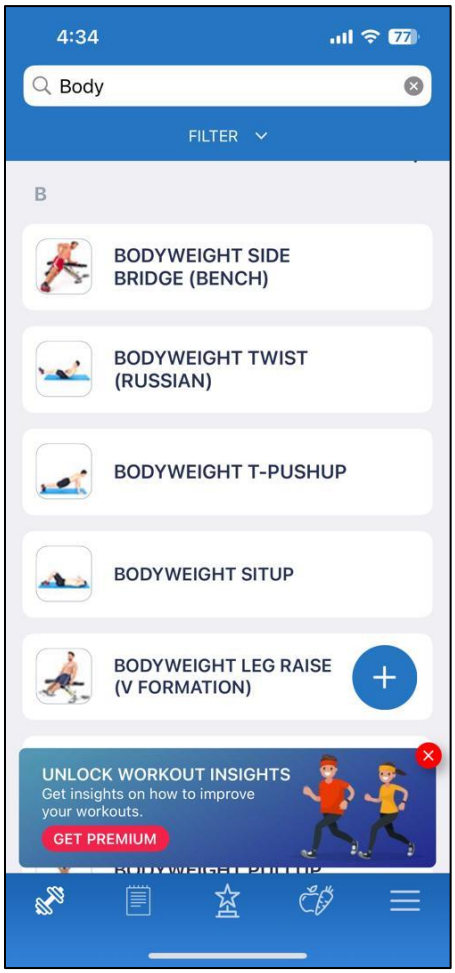


Figure 2.20: Workout searching page with relevant searching keywords "Body"

V. Workout Customization

Fitness Buddy allows users to customize workout where they can add exercise by providing the exercise name, description and selection on parameters like weight and repetition, repetition, weight and duration as well as duration. Once users click on create button, the app will generate a new workout according to their own personal taste , as seen in the figure below.

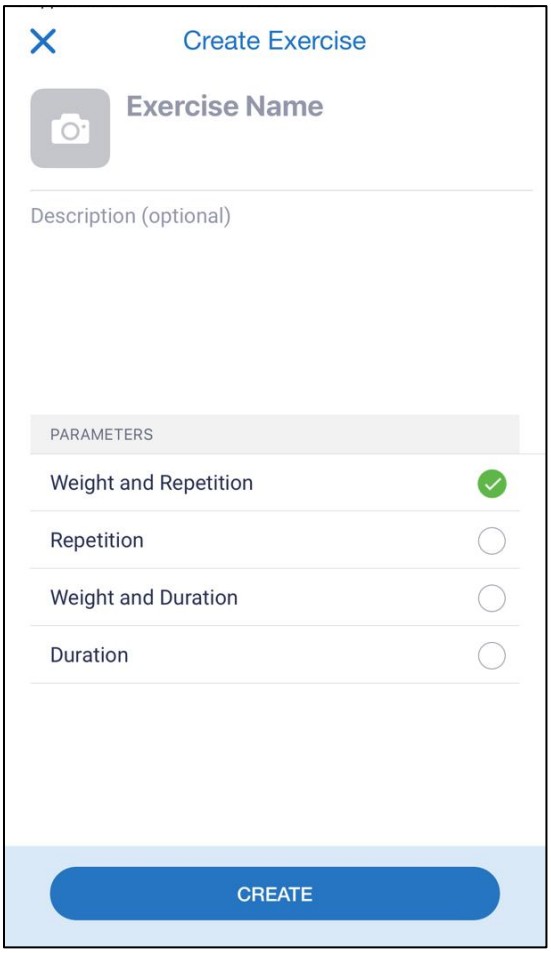


Figure 2.21: Workout customization of app

2.2.5 FitOn

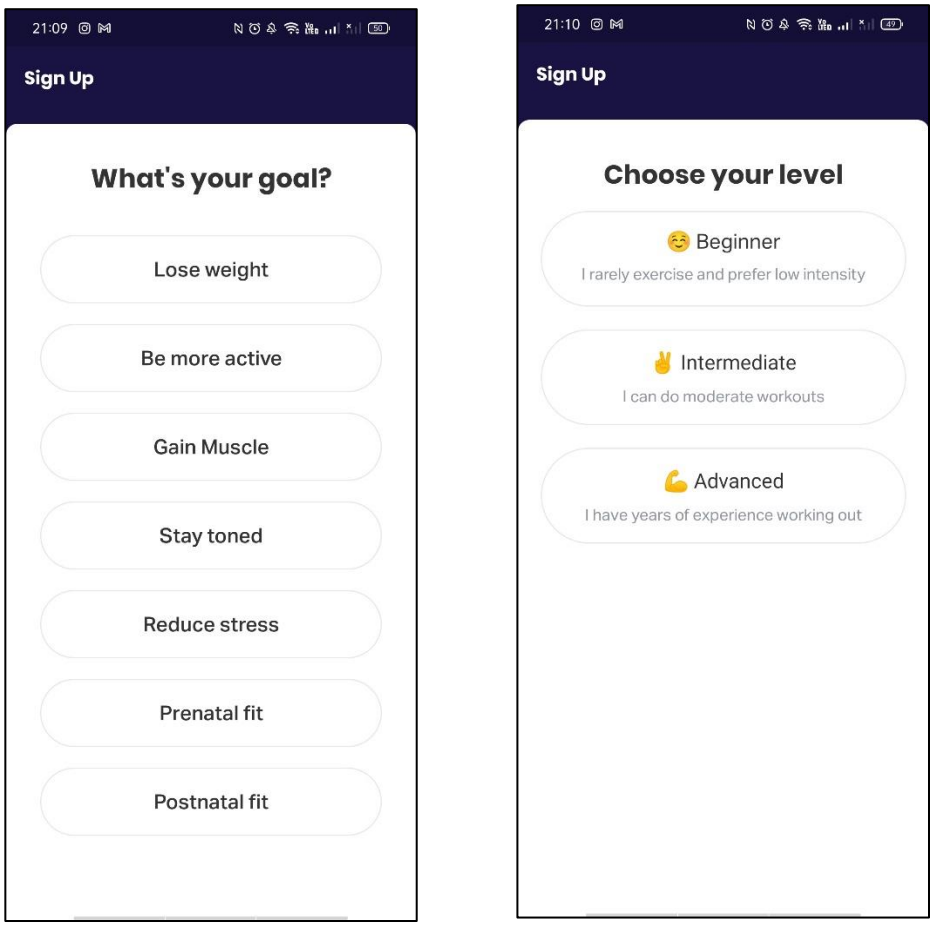
The FitOn app is a top digital wellness platform that offers variety of workouts sessions led by professional trainers with high caliber of video content. FitOn allow individuals to simply workout without any equipment used, which convenient for users to exercise anytime and anywhere. Moreover, FitOn aims to improve convenient workout experience by providing unlimited access on any screen, including android and apple mobile devices. The following are the key features of the fitness app along with the explanations:

i. Data Input
ii. Workout Recommendation
iii. Workout Information
iv. Workout Searching
v. Workout Library

vi. Workout Filtering

i. Data Input

FitOn app requires user to input their data, including gender, date of birth, height, weight, desired workout goals, fitness level, workout frequency and duration. These details input by user will be compiled to generate a personalized workout solutions that caters to user's needs. The figure below depicts the data input features where user provide workout desired goal and fitness level to generate customized workout plan.



(a) (b)

Figure 2.22(a)(b): User's data input to generate personalised workout solutions

ii. Workout Recommendation

FitOn app has incorporated a workout recommendation features, where the app will recommend user on workout plan based on user input that are provided

earlier during data input process. In this figure, the app will prescribe slim down workouts when a user choose lose weight as workout desired goals.

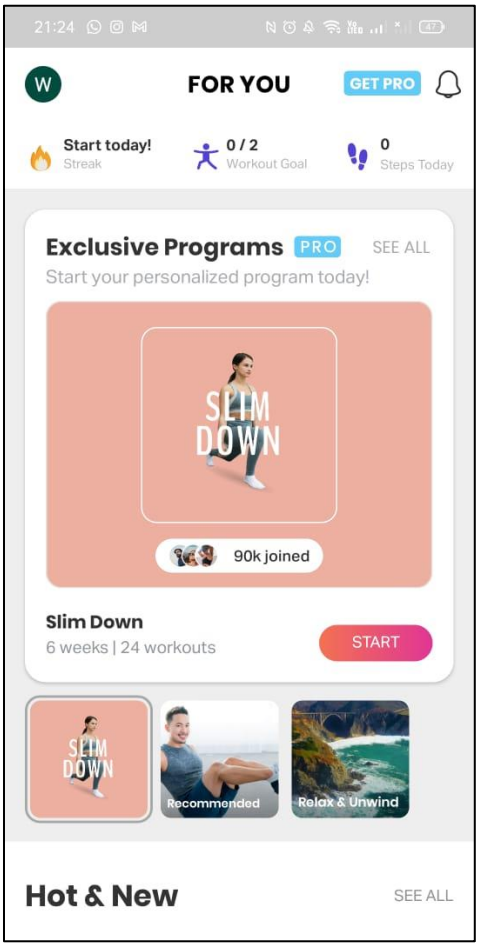


Figure 2.23: Workout recommendation based on workout desired goals

iii. Workout Information

The app has implemented a comprehensive workout information about the workout, such as workout name, trainer name, workout categories, time duration, equipment used and target areas. As show in the figure below, the detailed information of a Quickie Abs are displayed.

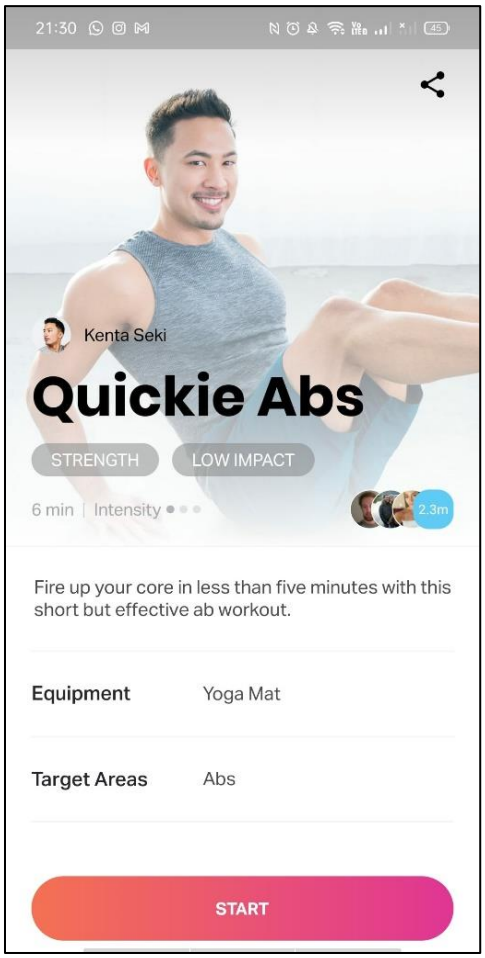


Figure 2.24: Workout information page of a Quickie Abs workout

iv. Workout Searching

FitOn app offers a workout searching features that enable users to search for workouts they're looking for more quicky and easily by entering relevant keywords. Next, the app will produces a list of workouts with what has been typed in, facilitating user's search for what they need. A figure below shows how typing the keyword name "yoga" will reveal a list of yoga sessions that contains the keyword name.

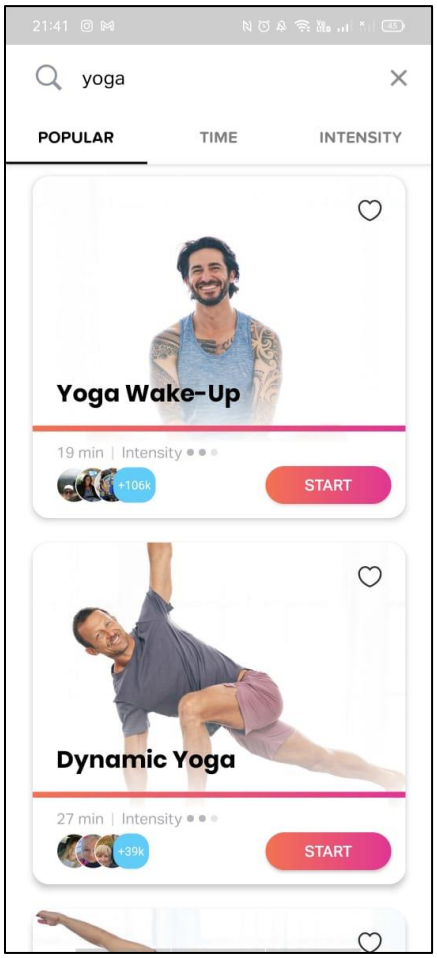


Figure 2.25: Workout searching page with relevant searching keywords "yoga"

v. Workout Library

FitOn app has incorporated a comprehensive range of workouts that are divided into fitness categories. The library contains exercises, each with detailed and high quality of video content that are led by professional trainers. The app extensive workout library is depicted in the figure below.

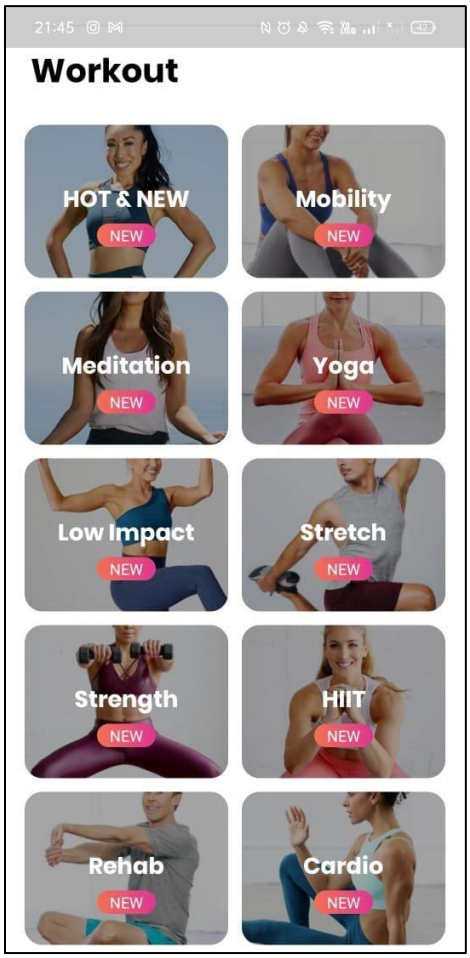
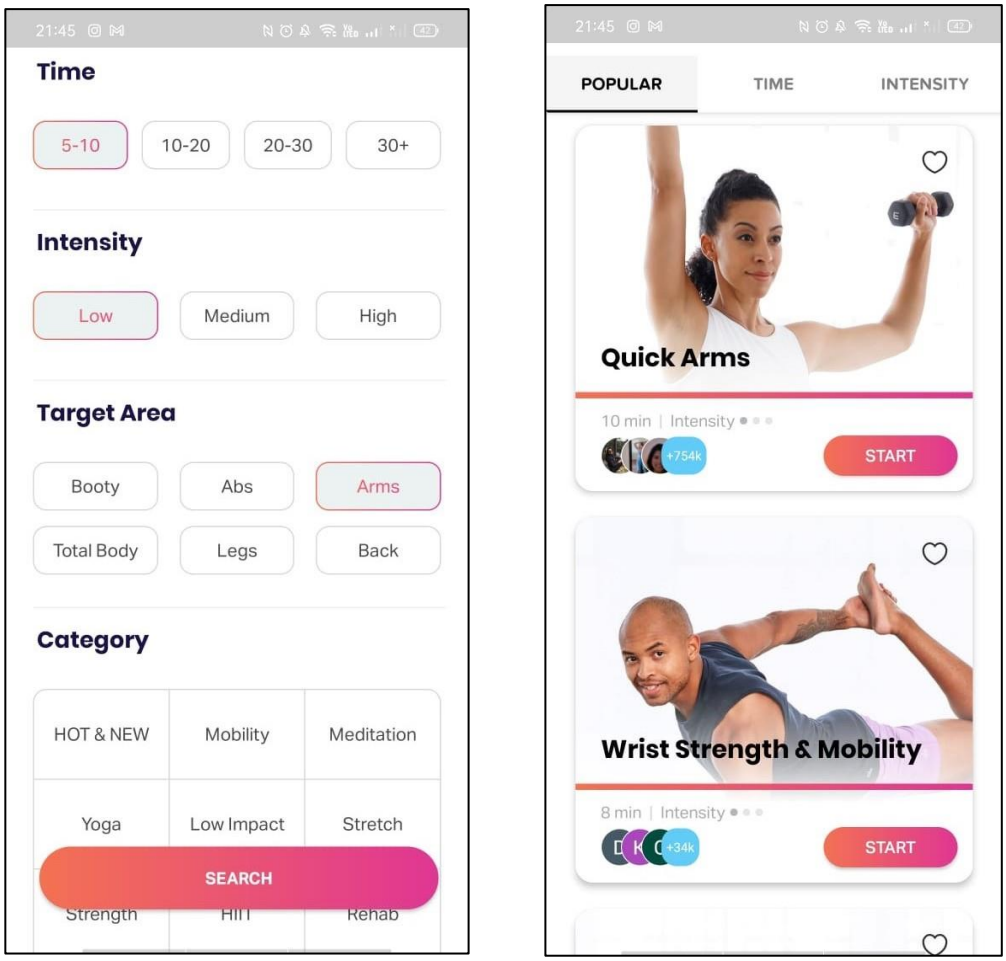


Figure 2.26 : Workout library of the app

vi. Workout Filtering

FitOn app offers a workout filtering features that categories the time, intensity, target area and workout category. Users could easily filter the workouts by selecting the desired choices to obtain their preferred workout plans. As shown in the figure below, how filtering the choices of time, intensity, target area and workout category will affect the output result of filtered workout plan.



(a) (b)

Figure 2.27(a)(b): Workout Filtering page with workout choices applied.

2.2.6 7 Minute Workout

The 7 minute workout app is designed based on high intensity circuit training (HICT) with wide range of exercises, and it takes only 7 minutes. The HICT has claimed that the app has been shown to be the safest, healthiest and most effective ways of building muscle. As the exercises are properly structured by HICT experts, users can achieve their desired fitness outcomes easily with minimal investment on 7 minute workout. The following is the key features of the the fitness app along with the explanations:

i. Workout Library
ii. Workout Information
iii. Progress Tracking Report

iv. BMI Calculator

i. Workout Library

7 Minute Workout app has include various types of exercises that are recommended by HICT experts. The library offers a comprehensice 7 minute exercises that are categorized according to the body componenet. Each of the exercises come with detailed instructions and high quality of video content that are led by professional trainers. The app extensive workout library is shown in the figure below.

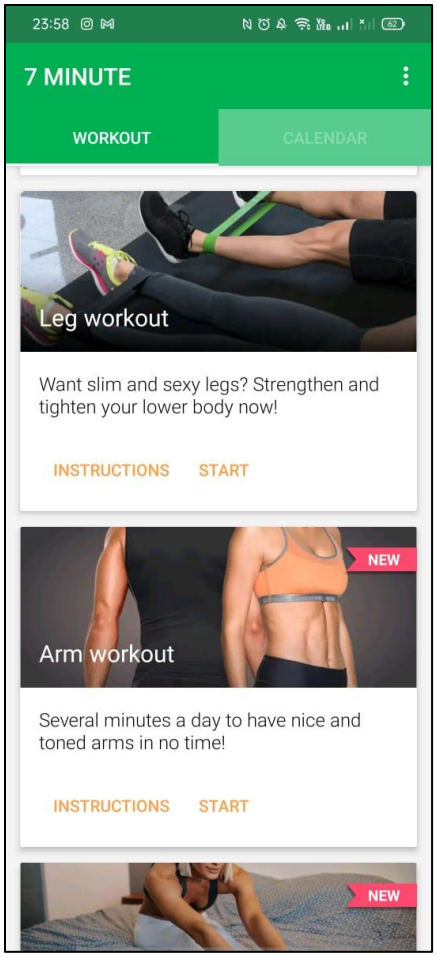


Figure 2.28: Workout Library of 7 Minute Workout app

ii. Workout Information

The app has integrated a wide range of workout information section where users can access comprehensive informations and descriptions about the workout such as workout name, workout instructions and workout videos. Users can click on video button to view workouts in video format instead of reading

instruction manually. Figure below illustrates the detailed information of an abs workout.

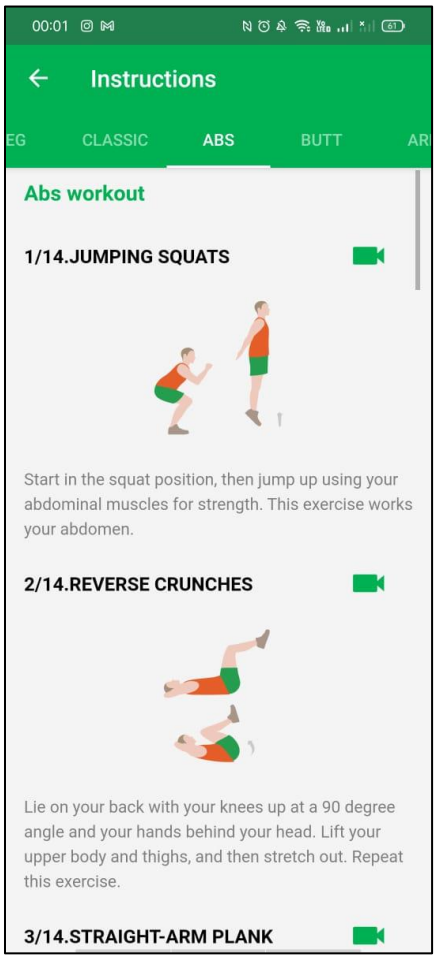
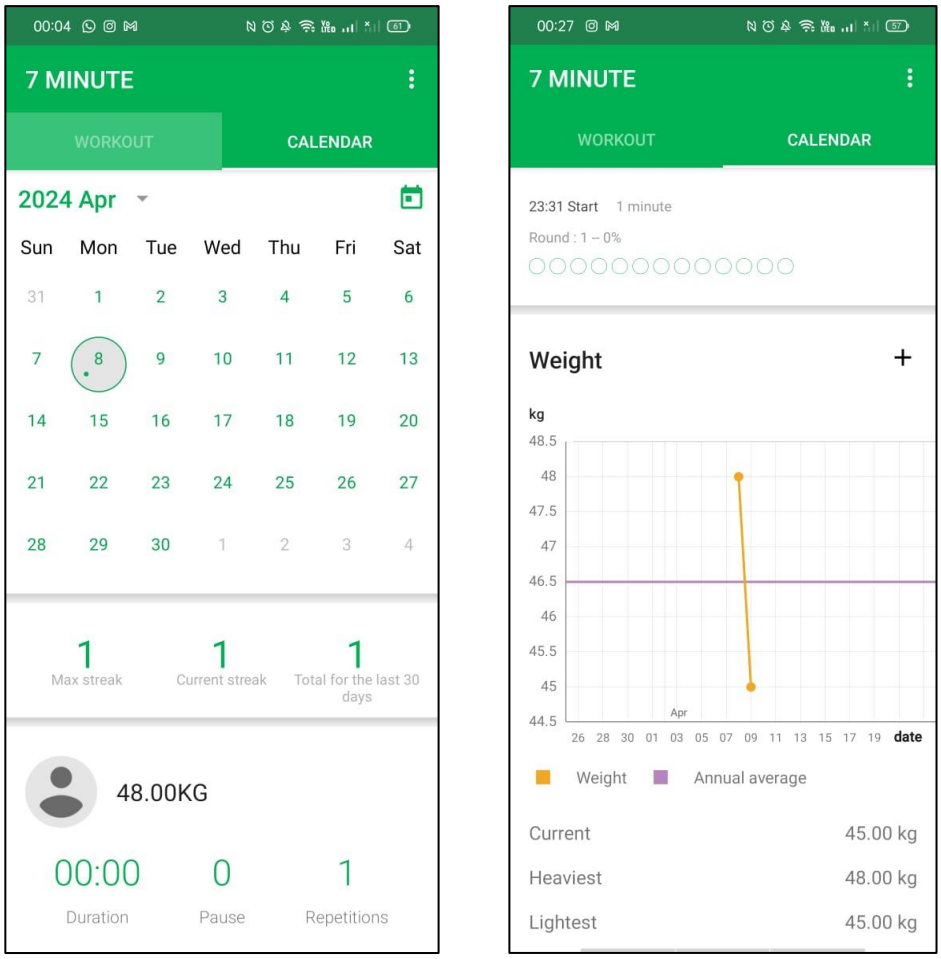


Figure 2.29: Workout information page of an abs workout

iii. Progress Tracking Report

The 7 Minute Workout app provides extensive details on each workout. As depicts in the figure below, it keeps track of user's daily workout progress in a calender, including max and current streak of workout , total number of workout, workout duration and repetitions. Also, users get to track their weight trends to determine their current weight, heaviest weight and lightest weight .Users can a monitor the development and results from the calendar, where they can stay motivated along their workout journey.

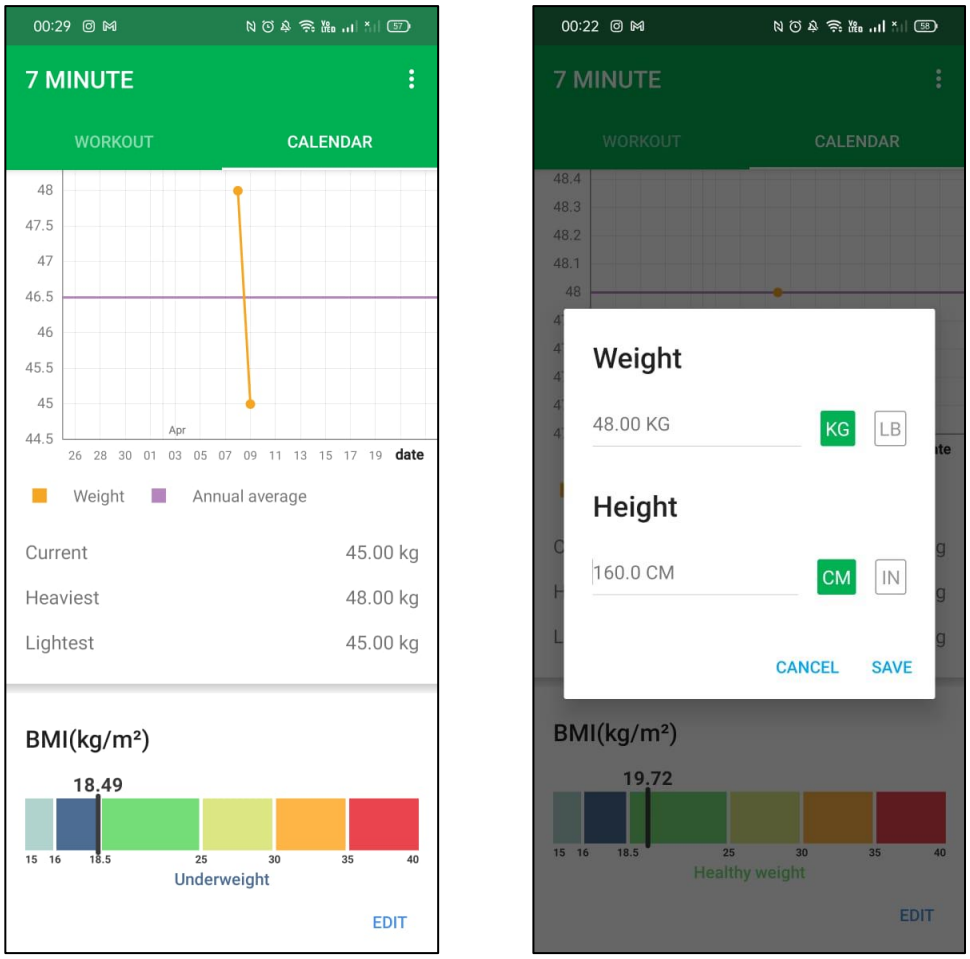


(a) (b)

Figure 2.30(a)(b): Workout progress tracking report details

iv. BMI Calculator

7 Minute Workout app has integrate a BMI calculator which requires user to input their weight and height to calculate their body composition index. This calculator aids to interpret user's body composition index and categorize user into different categories such as underweight, healthy weight or overweight. Hence, users can monitor their body health along with the fitness workout together to achieve a healthy lifestyle. In addition, the calculator is designed with inclusion of different measurement units such as kilograms, pounds, centimeters and inches which increase user experiences by catering to a wider user's needs.



(a) (b)

Figure 2.31(a)(b): BMI Calculator of the app

2.2.7 Home Workout – No Equipment

The Home Workout app is a highly rated app that offers a library of daily workout plans that allow users to maintain their fitness and gain muscle at home without having to visit to the gym. With this app, users can achieve comfortability in their own home without any time constraints. Also, the exercises are designed by experts that are carefully curated to user's goals where they can see a noticeable changes in their body within few weeks. Hence, users can maximize the results under the application which provides motivation during their fitness journey. The key features of the fitness app are as follows, accompanied by explanations:

i. Data Input

ii. Workout Recommendation

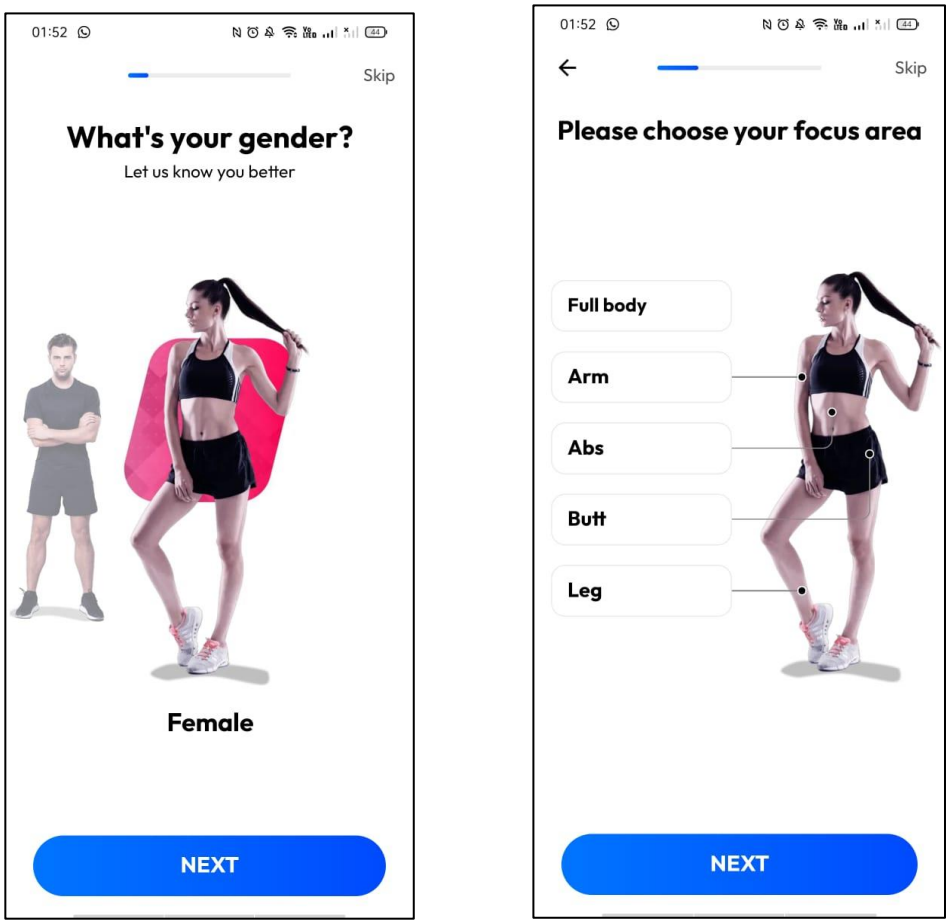
iii. Workout Library

iv. Workout Information

v. BMI Calculator

i. Data Input

Home Workout app offers a data input features, where users need to prompt in the data including gender, body focus area, main goals, weight, height and fitness level. These details will be summarized to create a personalized workout plan according to user preference. The figure below depicts the data input features where user provide gender and body focus area to generate customized workout plan.



(a) (b)

Figure 2.32(a)(b): User's data input to generate personalised workout solutions

ii. Workout Recommendation

On Home Workout app, after users provide data input , the app will come out with workout recommendation according to user goals, as seen in the figure below, where the app will prescribe lower body workouts when a user choose leg as their body part focus during the data input process.

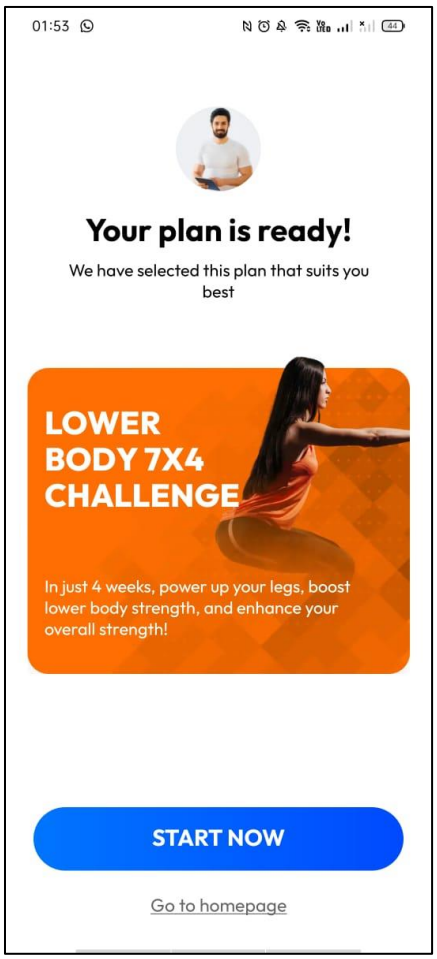


Figure 2.33: Workout recommendation based on workout desired goals

iii. Workout Library

The app has included a wide range of workouts, containing categories of classes according to fitness level with different body focus area. This allows users to progress with their workout according to their choices to maximize the result on workout and guarantee a well-initiated fitness journey. The app's extensive workout library is shown in the graphic below.

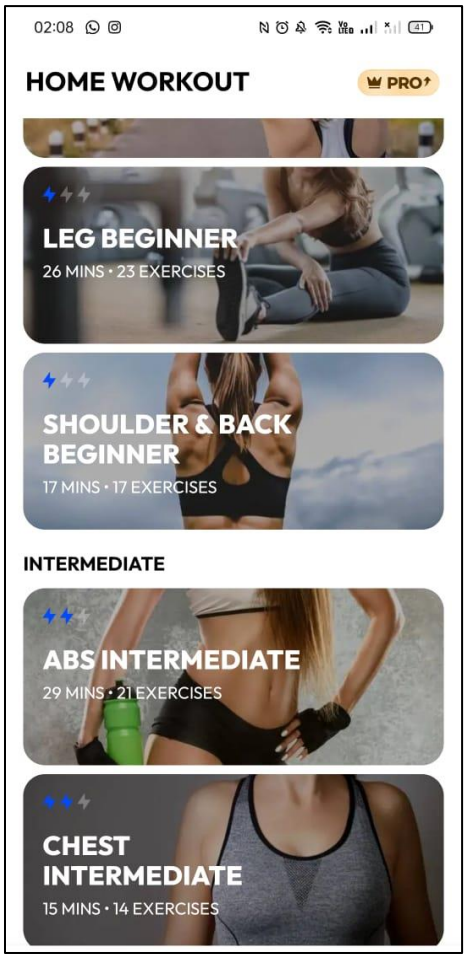
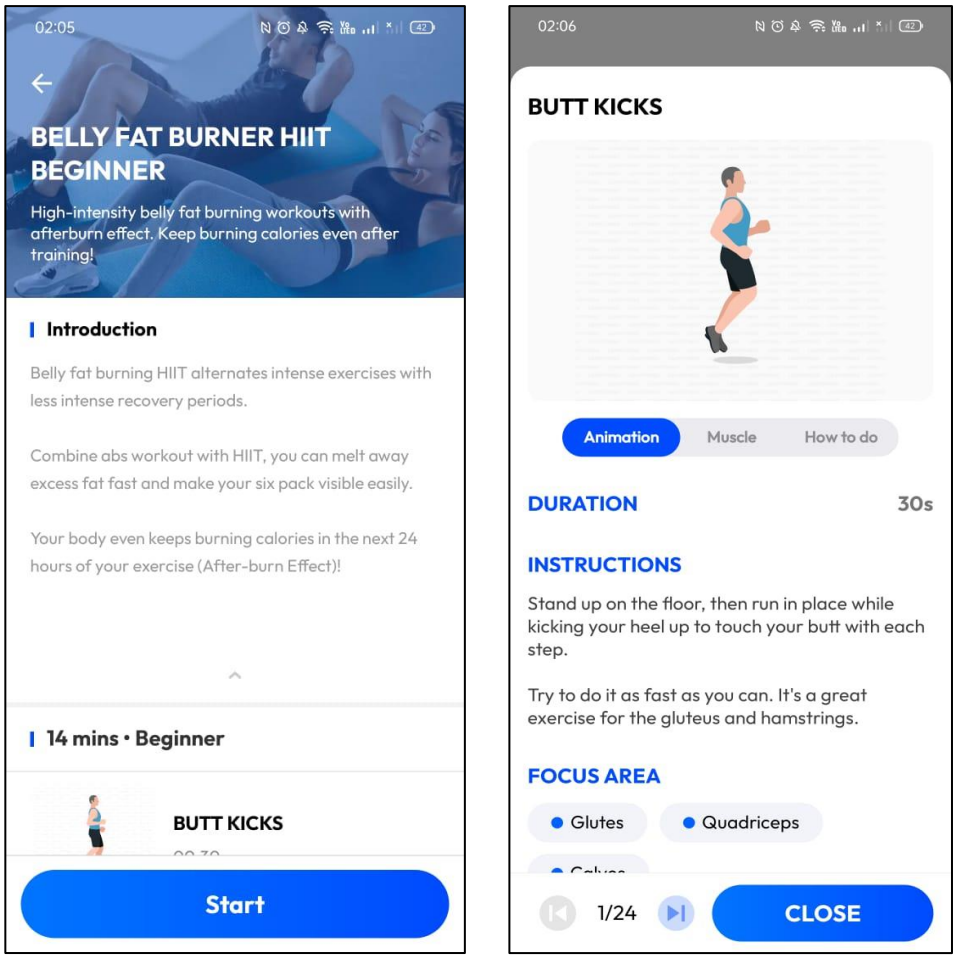


Figure 2.34: Workout library page of the app

iv. Workout Information

The app has integrated a wide range of workout information section where users can access comprehensive informations and descriptions about the workout class including the workout name, introduction and exercises included in the workout. Additionally, users may learn more about the specifics of each activity in the workout class by clicking on each one, which provides information on the exercise's length, written instructions, body emphasis region, and videos. Figure below illustrates the detailed information of a workout class.



(a) (b)

Figure 2.35(a)(b): Workout information page of a workout class

v. BMI Calculator

Home Workout app has integrate a BMI calculator which requires user to input their weight and height to calculate their body composition index. This calculator aids to interpret user's body composition index and categorize user into different categories such as underweight, healthy weight or overweight. Hence, users can monitor their body health along with the fitness workout together to achieve a healthy lifestyle. In addition, the calculator is designed with inclusion of different measurement units such as kilograms, pounds, centimeters and feet which increase user experiences by catering to a wider user's needs.

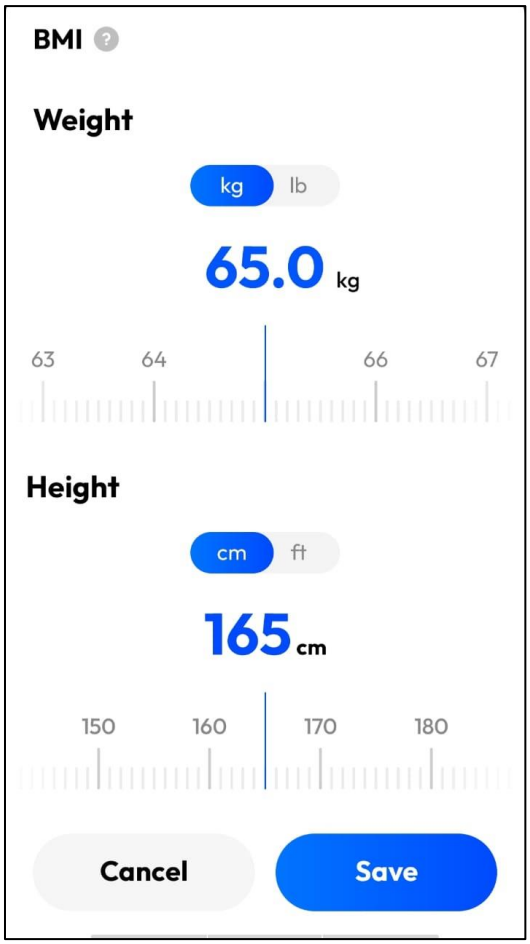


Figure 2.36: BMI Calculator of the app

2.2.8 Comparison between Existing Similar Mobile Applications

App name / Features	Fitbod Workout & Gym Planner	30 Day Fitness – Home Workout	Fiit	Fitness Buddy: Gym Workout Log	FitOn	7 Minute Workout	Home Workout – No Equipment
Data Input	Yes	Yes	Yes	Yes	Yes	No	Yes
Workout Filtering	Yes	Yes	Yes	Yes	Yes	No	No
Workout Recommendation	Yes	Yes	No	No	Yes	No	Yes
Workout Information	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Workout Searching	No	No	Yes	Yes	Yes	No	No
Workout Customization	Yes	No	No	Yes	No	No	No
Workout Library	No	Yes	Yes	No	Yes	Yes	Yes

Progress Tracking Report	Yes	No	No	No	No	Yes	No
AI gym equipment recognition	No	No	No	No	No	No	No
Calories calculator	No	No	No	No	No	No	No
BMI calculator	No	No	No	No	No	Yes	Yes
Nutritional intake information	No	No	No	No	No	No	No

Table 2.1: Comparison between existing similar mobile applications

All in all, each of the applications offered a special and useful features that makes them stand out from other applications. However, there are a few drawbacks of these 7 fitness applications, where the features shown below will be taken into account for implementation when developing the applications. These features includes:

i. AI gym equipment recognition
ii. Calories calculator
iii. Nutritional intake information

2.3 Software Development Methodologies Review

David Young (2013) explained that a software development methodology is a way of overseeing a software development project, where it address matters such as deciding which features to be in the current version, deciding when to release the software, assigning task to people and execute testing. By following the methodology, it assist developers in matching their workflow and characteristics of their products with its functional goals. There are various of software development methodology that comes with its own pros and cons. Hence, choosing the right software development methodology is crucial to maximize the success of project as it creates a solid foundation and guidelines for the whole development process. In this section, the review of 4 methodologies, which are Waterfall, Rapid Application Development (RAD), Prototype and Spiral Model will be conducted before determine on the final methodology.

2.3.1 Waterfall Methodology

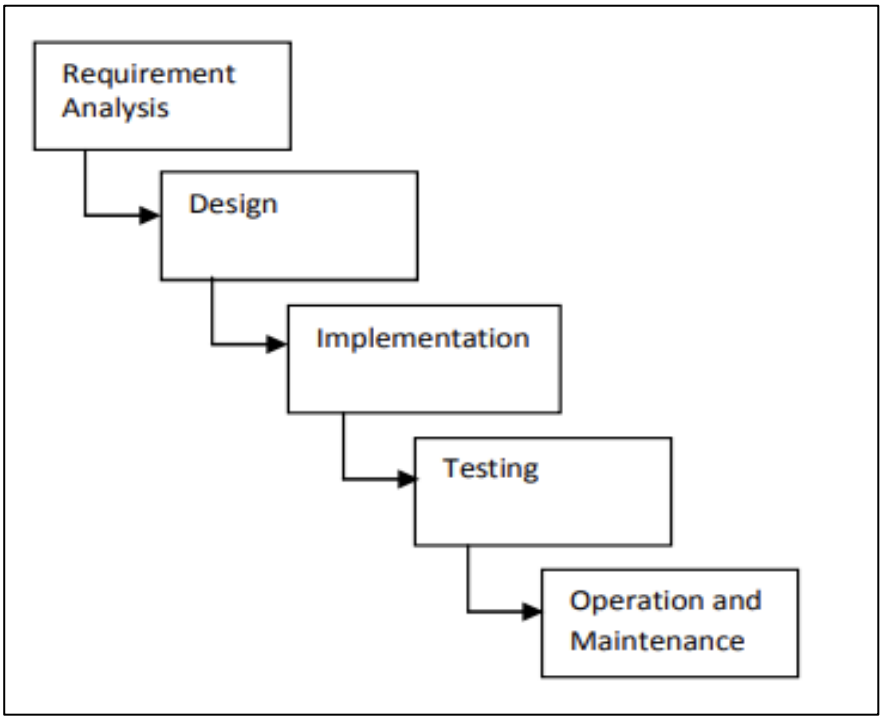


Figure 2.37: Waterfall Model (Adenowo, A.A. and Adenowo, B.A., 2013)

Adel Alshamrani and Abdullah Bahattab (2015) stated that the waterfall model is the ancient and well-known SDLC model. Bassil (2012) mentioned that the waterfall approach is sequential software development process which proceeds in downward direction, much like waterfall. The progress used to flow from one phase to another but only proceeds if each phases are thoroughly defined and completed before moving onto the next phase. Hence, the approach had restricted the process to go back to the previous phase and allow changes in it (Pargaonkar, S. , 2023).

According to the figure 2.37, there are five phases in the waterfall model which comprises of requirement analysis, design, implementation, testing, operation and maintenance, where each phase will have its own duration based on its goals and depends on previous phase to proceed. Balaji and Murugaiyan (2012) affirmed that the defects will be found very late as testing team is not involved from the beginning of the project, but only involved in testing phase. In such case, a defects will not be identified until the testing phase is reached, which leads to a subpar design and lengthy project timeline.

Aditionally, the waterfall approach works well for small software projects when the project manager and the team can simply and precisely plan and the project owner has well specified requirements. The project team can resume work and begin a new phase upon approval.

The following table depicts the advantages and disadvantages of the Waterfall model.

Advantages	**Disadvantages**
Suitable for small projects with simple and well defined goals.	Not suitable for complex and uncertainty high risk projects.
Simple and easy to implement due to rigidity of model.	Not flexible to reverse to previous phase and make changes.
Each phase has defined goals due to well arranged plans and procedure which should be followed. (Kannan, Jhajharia es. el, 2014)	Challenging to establish specifications for everything since the project requirements are always changing. (Kannan, Jhajharia es. el, 2014)
Uses clear structure and do not overlap	Time consuming and higher cost due to the delay of testing until the last phase of project.
Requirements and goals are set early in the project.	No feedback to identify error. (Chandra, V., 2015)

Table 2.2: Advantages and Disadvantages of Waterfall Model

2.3.2 Prototype Methodology

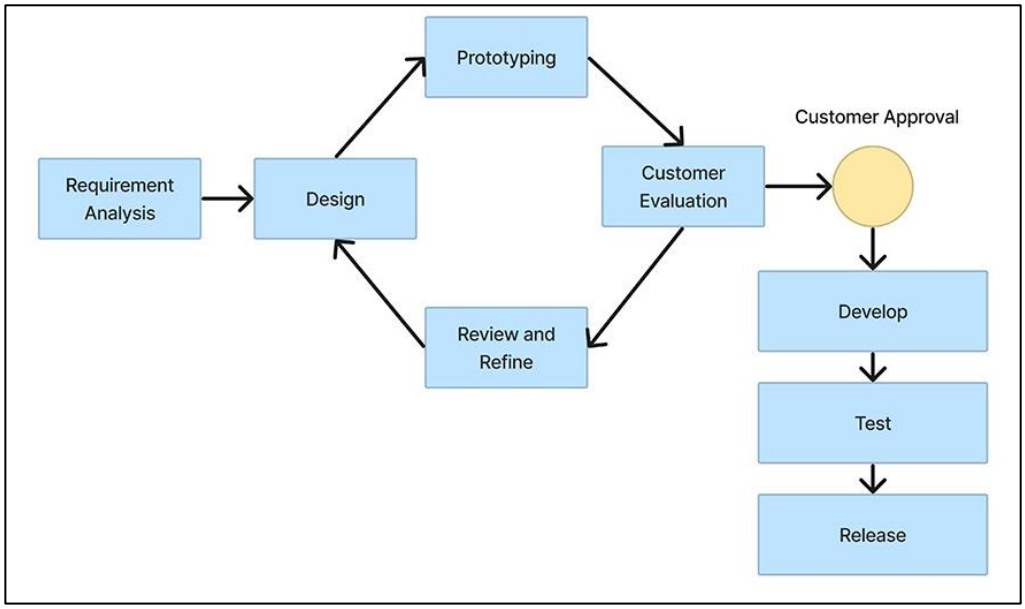


Figure 2.38: Prototyping Model (Anon, 2022)

The Prototyping model is a popular approach in which the process of design, prototyping, customer evaluation, review and refine are carried out concurrently, and they are all repeated in a cycle until the system is completed and an acceptable prototype is achieved (Senarath, 2021). The process is an iterative approach that involve both the user and developer to trial and error by refining the specifications through communication. According to the figure 2.38 above, the initial prototype of the little functional system mock-up will be created and presented to the user where they respond to offer feedback. However, sometimes issue arises when system does not meet the satisfaction of the user. Hence, user will voice out to provide feedback or propose new ideas that can be collected and provided to the developer which are useful in refinement and adjustment to produce a second prototype with more features (Dennis et al., 2014). This cycle of the process doesn't end until the sponsor, analysts, and users concur that the prototype has sufficiently meet the requirements. Once done, a final system is developed and deployed based on the approved final prototype, otherwise, the project will proceed to next phase instead of producing a new prototype.

According to Despa (2014), prototype methodology has claimed to be the best model as the prototypes can be build quickly, and most of the time, the standard practices for programming are ignored. In addition, this methodology

is mainly suitable for large software projects as well as for novel innovations and software projects of this type that have never beem created before (Soobia.et.al.,2019).

The table below shows the advantages and disadvantages of Prototype methodology.

Advantages	Disadvantages
Actively involve users to provide feedback and suggestions on the prototype.	Time consuming as it take several prototypes to meet the client's ultimate needs (Chandra, V., 2015).
Flexible as new requirements can be easily accommodated (Chandra, V., 2015)	Too much variation in requirements causes poor documentation.
Higher users satisfaction.	Users may get confused in the prototypes and actual systems (Senarath, 2021).
Able to detect error in early phase.	May increase the complexity of the system due to addition of requirements.
Quick feedback from users (Senarath, 2021).	Degree of uncertainty in the number of iterations

Table 2.3: Advantages and Disadvantages of Prototype methodology

2.3.3 Rapid Application Development (RAD) Methodology

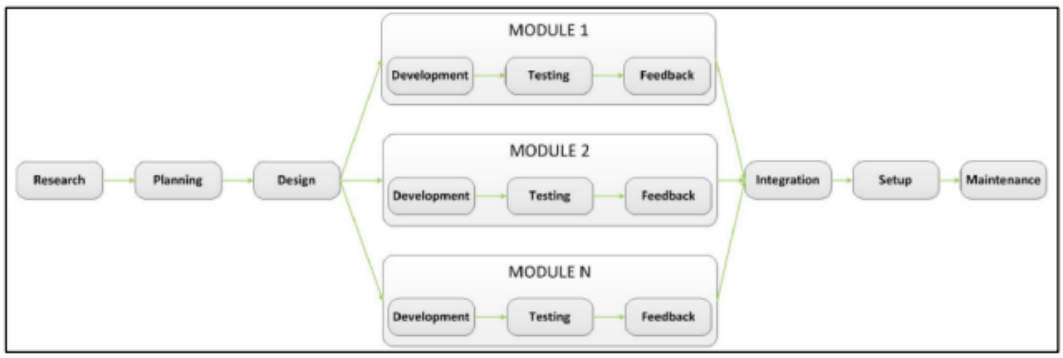


Figure 2.39: Rapid Application Development (RAD) methodology (Despa, 2014)

The Rapid Application Development (RAD) methodology was initially presented by James Martin in 1991. He prioritized quality and speed over cost in the development process in an effort to transcend the constraint of traditional development lifecycle (Berger, Beynon-Davies and Cleary, 2004). Its made to maximize the benefits of robust developing software that has lately advanced. The RAD model is centered on the user involvement and prototyping, essentially condensing all the phases into a short and iterative development cycles. Hence, this approach promotes efficacy and self-correcting, which was viewed as a solution for perceived shortcomings with the traditional lifecycle.

The RAD methodology covers 3 stage and 4 stage cycles. In 4 stage cycles , it comprises of requirement planning, user design, construction and cutover (Fazal, 2020). It also employs that the development cycles are time-boxed and multiple cycles of development can be done simultaneously as depicted in the figure above. By using the notion of prototyping, this approach focus on collecting user requirements and involve them in early testing process. Hence, user may swifly acquire a prototype application from RAD and have a better understanding of the system to recommend changes that will almost certainly bring the system closer to what is required.

Requirements planning and user design are integrated into an iterative activity in the three-stage cycle (Fazal, 2020). The model accelerate task completion without sacrificing customer satisfaction by prioritizing highly iterative design and construction phases over extensive planning. In fact, only up to 30% of the overall work in RAD is devoted to the requirements, specification, and design. Next, the prototyping and construction phases may be repeated until users are satisfied with the requirements before acknowledged as a complete product.

According to Despa (2014), this methodology works well for smaller-scope projects that are created by small teams. However, it might not be as appropriate for larger-scale projects and intricate tasks as the prototype are more expensive when they become harder to extend.

The table below illustrated the advantages and disadvantages of RAD methodology.

Advantages	**Disadvantages**

Actively involve user for feedback which enhance high user satisfaction.	Requires numerous user's input that increase the process complexity.
Reduce cost and mitigate failure risk by including users in early stage of development process (Geambaşu, 2011).	Possible to miss and overlook important requirement due to rapid project iteration pace (Geambaşu, 2011).
Suitable for small teams with small project scope.	Not suitable for large and complex projects.
Able to develop quick iteration that leads to time saving.	Requires skilled and experienced team.
Enhances reusability on prototype.	Lack of scalability in final product if the model is created as a complete application from the start rather than a prototype.

Table 2.4: Advantages and Disadvantages of RAD model

2.3.4 Spiral Model

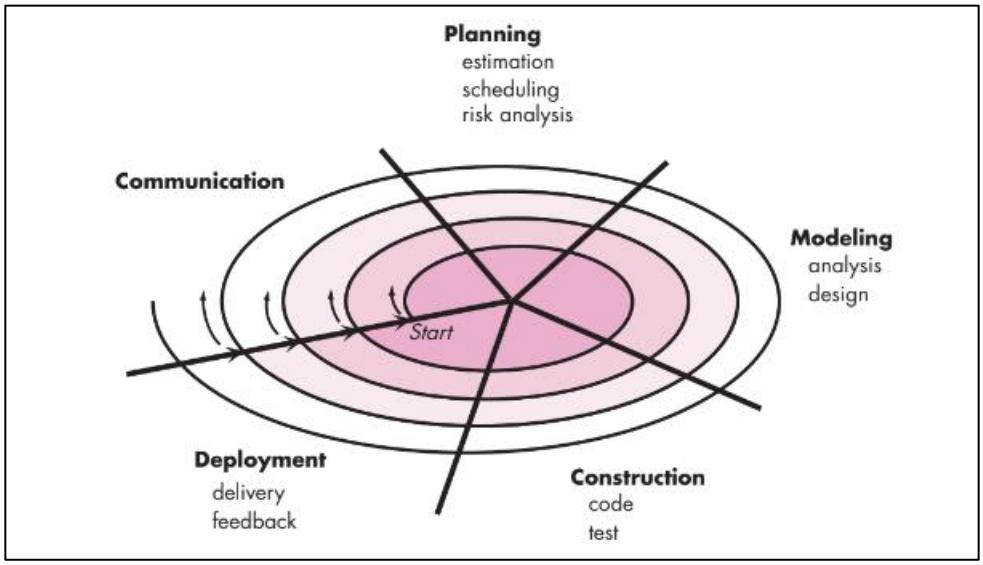


Figure 2.40: Spiral Model (Pressman, 2009)

The spiral model was initially proposed by Boehm that claims to be a crucial model in software development life cycle (Doshi, 2021). The model is represented by a spiral that begins in the center and spirals outward in a clockwise manner, with each round of the spiral representing a stage in the

process (Sommerville, 2015). According to Pressman (2009), a spiral model consist of 5 task regions which comprises of customer communication, planning, modelling, construction and deployment as depicted in the figure above.

By using the spiral approach, the software is produced in a sequence of incremental releases according to the phases. During the early stages, a product specification may be developed after the first circuit of the spiral, prototypes and increasingly more advanced software versions may be developed after consecutive circuits of the spiral. During subsequent iteration, the designed system result in ever more comprehensive versions (Pressman, 2009). The project plan is modified after each transit through the planning region with users evaluation and suggestions until the software is prepared to be installed in the live environment. Furthermore, the model places a strong emphasis on risk analysis and consistently considers a number of options before choosing one to adopt.

Doshi (2021) affirmed that the model is mostly employed in projects that are big, costly and complex to be managed. The model is also proficient in handling the risk which implement the internal project effectively, such as in medium-to-high risk projects and those with complex demands.

The following table listed the advantages and disadvantages of Spiral model.

Advantages	**Disadvantages**
Suitable for big, costly and complex project.	Not suitable for small project due to high cost.
Adjustments can be made based on user feedback and suggestions.	Require an experienced and skilled team to run the model.
Flexible and adapt to changes.	Complex due to multiple iterations in the project.
Manage risk effectively.	More documentation is needed as compared to other model.
Features are introduced methodically and development proceeds in fast pace.	Risk of not meeting project schedule due to uncertain number of phases in the early stage.

Table 2.5: Advantages and Disadvantages of Spiral model

2.3.5 Comparison Between Software Development Methodologies

Methodology/ Characteristics	Waterfall	Prototype	RAD	Spiral
Flexibility in requirement changes	Low	High	High	High
User involvement	Low (only involve in early stage)	High	High	High
Speed of delivering a product	Slow	Fast	Fast	Fast
Experts required	No	Little	Yes	Yes
Risk factor	High	Low	Medium	Low
Team size	Big	Small	Small	Very big
Project scale	Small	Small	Small or medium	Big

Table 2.6: Comparison between software development methodologies

Selecting the correct methodology is essential to a project's effective implementation. To optimize the possible benefits, it is crucial to carefully consider which approach is most appropriate for the project as each has pros and cons of its own.

Hence, prototyping methodology will be selected for this project above alternative approaches for a number of reasons. When it comes to the flexibility to requirement changes, this model is far more flexible than Waterfall methodology. This implies that the requirements can be enhanced or change through a number of iterative process. Other than that, it has a high level of user involvement which guarantees that the users are involved in the project to

provide feedback and suggestions in order to define projects according to their requirements.

Furthermore, the speed of delivering a product for prototype model is in fast pace when compared to other model like Waterfall. In addition, the specialist required for prototype is little and the model is good at handling risk. The Spiral model requires a very big team size and project scale while the prototype methodology only requires small team size and project scale. All in all, we can conclude that prototype methodology is the ideal methodology to be used in this project.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

The methodology implemented which is prototype methodology and project planning will be explained and discussed in this chapter. A Gantt chart and work breakdown structure are also included for outlining and monitoring during the whole development process. In addition, the tools used for developing ths system will be specified in this chapter.

3.2 Software Development Methodology

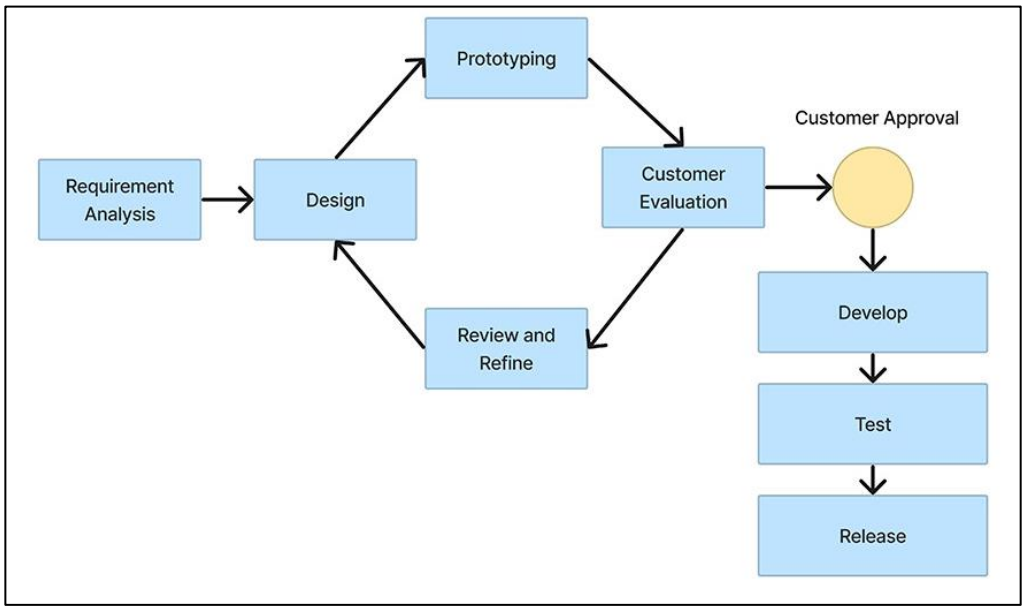


Figure 3.1: Prototype Methodology (Anon, 2022)

According to the figure above, the prototype methodology are made up of 5 main prototype process, which are requirement analysis, the iterative loop of prototype, development, testing and deployment phase. Inside the loop, it consists of 4 iterative phase which are design, prototyping, customer evaluation and review and refine. The successive phases, each with refinement and improved functionality are repeated until user's specified needs are satisfied.

3.2.1 Requirement Analysis

The first step in the prototyping methodology will be the requirement analysis. For this project, a variety of techniques are employed to collect precise and useful data. The techniques used in this project will be literature review to review similar existing mobile application, software development methodology and quantitative methodology like questionnaires to collect insightful feedback from users.

3.2.1.1 Literature Review

A literature review aims to increase understanding by thoroughly examine and analyse the project in order to determine the most effective methods for its advancement. There are 2 literature reviews to be carried out in this project, review similar existing mobile application and software development methodology.

The first literature review is conducted to analyse the 4 similar existing mobile applications such as Fitbod Workout and Gym Planner, 30 Day Fitness-Home Workout, Fiit, Fitness Buddy: Gym Workout Log, FitOn, 7 Minute Workout and Home Workout – No Equipment. Each mobile application's features will be listed and compared to determine the comprehensive features that a fitness application requires and evaluate the shortcomings of rivals.

The second literature review aims to analyse the 4 software development methodologies, including the Waterfall methodology, Prototype methodology, RAD methodology and Spiral methodology. For each of these 4 approaches, the pros and cons are examined in a comparison table created in Chapter 2. By conducting this review, it helps to select the best software development methodology that is most appropriate for the project's needs.

3.2.1.2 Questionnaire

The questionnaire is distributed to 20-40 respondents that comprise of 20 questions. The questionnaire is divided into 4 sections, including personal information, preferences and requirements, usability and features as well as feedback. The division of questions into sections aims to collect user's data and valuable insights in a systematic manner.

3.2.2 Prototype Iteration

This project selects the prototype model as its methodology. There are four phases which includes design, prototyping, customer evaluation and review and refine. Each phase is continually repeated by evaluating and refining until a satisfactory end product is implemented.

3.2.2.1 First Iteration

In the first iteration, the main priorities will be designing the user interface of the mobile application to get an overview of the app framework and architecture on how it should look like. Hence, a low fidelity prototype design will be created for the project to visualise the entire system .

Design:

In the design phase, it focus on designing a user-friendly interface by drafting the design layout for each screen and determine the screen navigation and routing. Hence, the utilization of Axure RP 9 is included in this phase to get a visualisation of the design and dynamic screen flows. Moreover, the database requirement and entity relationship will be created and defined clearly. This is to ensure that the database and user interface are clearly understood and the development process is smoothly progressed.

Prototyping:

The mobile application features that have been designed are developed according to the initial requirements in the first iteration, making sure that each feature functions as intended. In addition, the database is created using MySQL for data management such as updating, storing and retrieval of data.

Customer Evaluation:

The prototype created in the previous phase will be showcased to users for feedback and evaluation. The feedback and review are essential to identify problems, improve and refine the prototype.

Review and Refine:

The feedback and review are recorded down for further analyzation and refinement. Before going on to the next iteration, the prototype is improved based on these comments to make sure the user's needs and satisfaction are met.

3.2.2.2 Second Iteration

The second iteration emphasize on the features needed for implementation. The features are separated into two categories: essential features and optional features. The fundamental CRUD (Create, Read, Update, Delete) activities make up the core features. The supplementary features, on the other hand, offer extra functionality like workout information, workout search, workout recommendation and so on.

Design:

In the design phase, all necessary features in the user interface are listed down and drafted. The overall design of the mobile application interface is further optimized, including layout and contrast to enhance user experience across devices.

Prototyping:

The first prototype will undergo second iteration for additional implementation. All features that have been specified are incorporate during this iteration to make sure each feature functions as intended and is connected with the database.

Customer Evaluation:

The prototype created in the second iteration will be showcased to users for feedback and evaluation. Users will examine and test each functionality of the features and provide feedback on how well the prototype works. The feedback and review are essential to identify problems, improve and refine the second prototype.

Review and Refine:

Similar to the first iteration, the feedback and review are recorded down for further analyzation and refinement. In order to ensure that the needs of the user

are satisfied, the second prototype is updated with the aid of these feedback before moving on to the next iteration.

3.2.2.3 Third Iteration

The third iteration emphasize on the AI-powered gym recognition features for mobile application implementation. This feature will serve as gym equipment recognition function to help user to recognize various types of gym equipment effortlessly.

Design:

The third iteration design process include creating and drafting all necessary features in the AI gym equipment recognition. The data flow between the recognition and database is meticulously designed and constructed to guarantee the ability to recognize the gym equipment correctly.

Prototyping:

This phase includes training the AI gym equipment recognition with training data to ensure it performs as planned.

Customer Evaluation:

The prototype created in the third iteration will be showcased to users for feedback and evaluation. Users will examine and test each functionality of the features and provide feedback on how well the prototype works. The feedback and review are essential to identify problems and requirements that are not met.

Review and Refine:

The feedback and review are recorded down for further analyzation and improvement. The prototype is then undergo further refinement to make sure the user’s needs and improve overall satisfaction.

3.2.3 Develop

The project will proceed to the development stage once the user are satisfied with the final prototype. In this phase, the mobile application will be developed accordance with the final prototype. The mobile application will be developed using tools like Visual Studio Code, Flask, Node.js, Python and React Native.

3.2.4 Test

In this phase, several types of testing will conducted throughout the testing process to evaluate the mobile application's performance.These includes unit testing, system testing and user acceptance testing. The purpose of the software testing is to identify any issue or bugs in the code. The description, data and predicted results are included in the test case to show the result of the testing, either pass or fail. Moreover, 5 users have been chosen to perform user acceptance testing to ensure the mobile application fits the needs and preferences of the user in general.

3.2.5 Release

After completing and passing every testing process, a final report is created to document every aspects of the project's workflow. Once the mobile application is deployed, users can utilize the mobile application in real world for their own use, such as searching for workout and exercises.

3.3 Project Plan

3.3.1 Work Breakdown Structure (WBS)

0.0 Mobile-based Fitness Application with AI-Powered Gym Equipment Recognition Integrated

1.0 Project Planning and Requirements Gathering

- 1.1 Outline Introduction
- 1.2 Identify importance of study
- 1.3 Define problem statement
- 1.4 Define project objective
- 1.5 Determine project solution
- 1.6 Propose project approach
- 1.7 Specify project scope
  - 1.7.1 Identify target users
  - 1.7.2 Define main module covered for each target user
- 1.8 Conduct literature review
  - 1.8.1 Review on similar existing mobile applications
    - 1.8.1.1 Study features of Fitbod Workout & Gym Planner
    - 1.8.1.2 Study features of 30 Day Fitness-Home Workout
    - 1.8.1.3 Study features of Fiit
    - 1.8.1.4 Study features of Fitness Buddy:Gym Workout Log
    - 1.8.1.5 Study features of FitOn
    - 1.8.1.6 Study features on 7 Minute Workout
    - 1.8.1.7 Study features on Home Workout – No Equipment
    - 1.8.1.8 Comparison between the similar existing mobile applications
  - 1.8.2 Research on software development methodologies
    - 1.8.2.1 Specify the characteristics, pros, and cons of Waterfall Model
    - 1.8.2.2 Specify the characteristics, pros and cons of Prototype Methodology
    - 1.8.2.3 Specify the characteristics, pros, and cons of Rapid Application Development Methodology

1.8.2.4 Specify the characteristics, pros, and cons of Spiral Methodology

1.8.2.5 Comparison between software development Methodologies

1.9 Methodology and work plan

1.9.1 Determine the most applicable methodology

1.9.1.1 Define each phase of the methodology

1.9.2 Create Work Breakdown Structure (WBS)

1.9.2.1 Identify tasks and activities

1.9.3 Create Gantt Chart

1.9.3.1 List down the tasks and activities

1.9.3.2 Set start-date and end-date

1.10 Conduct fact finding analysis

1.10.1 Construct Questionnaire

1.10.2 Distribute Questionnaire in Google forms

1.10.3 Perform data analysis for questionnaire

1.11 Identify project development tools

1.12 Requirement Specifications

1.12.1 Determine the functional requirements

1.12.2 Determine the non-functional requirements

2.0 System Design

2.1 Use Case Modeling

2.1.1. Illustrate use case diagram

2.1.1.1. Define use cases

2.1.2 Compile use case description table

2.2 Design Modeling

2.2.1 Illustrate context diagram

2.2.2 Illustrate Level 0 data flow diagram

2.2.3 Illustrate activity diagrams

2.3 System Architecture Design

2.3.1 Illustrate Architecture Design

2.3.2 Architecture Design Explanation

2.4 AI Gym Equipment Recognition Architecture Design

2.4.1 Illustrate AI gym equipment recognition architecture design

2.4.2 AI gym equipment recognition architecture explanation

2.5 Rough Prototyping

2.5.1 UI design

2.5.2 Define routing between UI

3.0 Protype Implementation

3.1 First Iteration

3.1.1 Design UI prototype details of each page

3.1.2 Get User Feedback

3.1.3 Refine prototype based on user feedback

3.2. Second Iteration

3.2.1. Construct prototype with main functions

3.2.1.1 Authentication Function

3.2.1.2 Workout Recommendation Function

3.2.1.3 Workout Searching Function

3.2.1.4 Workout Filtering Function

3.2.1.5 Workout Customization Function

3.2.1.6 Workout Data Input Function

3.2.1.7 Workout Progress Tracking Report Function

3.2.1.8 Workout Library Function

3.2.1.9 BMI and Calorie Calculator Function

3.2.2.0 Nutrition Function

3.2.2 Get User Feedback

3.2.3 Refine prototype based on user feedback

3.3. Third Iteration

3.3.1 Construct prototype with AI Gym Equipment Recognition

3.3.1.1 Prepare training data

3.3.1.2 Develop recognition model

3.3.1.3 Implement AI Gym Equipment Recognition Functionality

3.3.2 Get User Feedback

3.3.3 Improve prototype based on user feedback

4.0 Development

4.1 Develop all module and functions along with its routings

4.2 Develop the authentication function for users

4.3 Develop AI gym equipment functions

4.4 Integrate AI gym equipment into the mobile application

4.5 Build Database For All Functions

4.6 Refine and reduce code duplication

4.7 Improve overall UI design

5.0 Testing

5.1. Conduct Unit Testing

5.1.1. Create unit test cases

5.2. Conduct System Usability Testing

5.2.1. Create usability test scenario for users

5.2.2. Calculate and document the results

5.3. Conduct User Acceptance Testing

5.3.1. Create user acceptance test template for users

5.5. Rearrange Documentation

6.0 Improvement

6.1 Improve mobile application

6.2 Prepare presentation slides

3.3.2 Gantt Chart

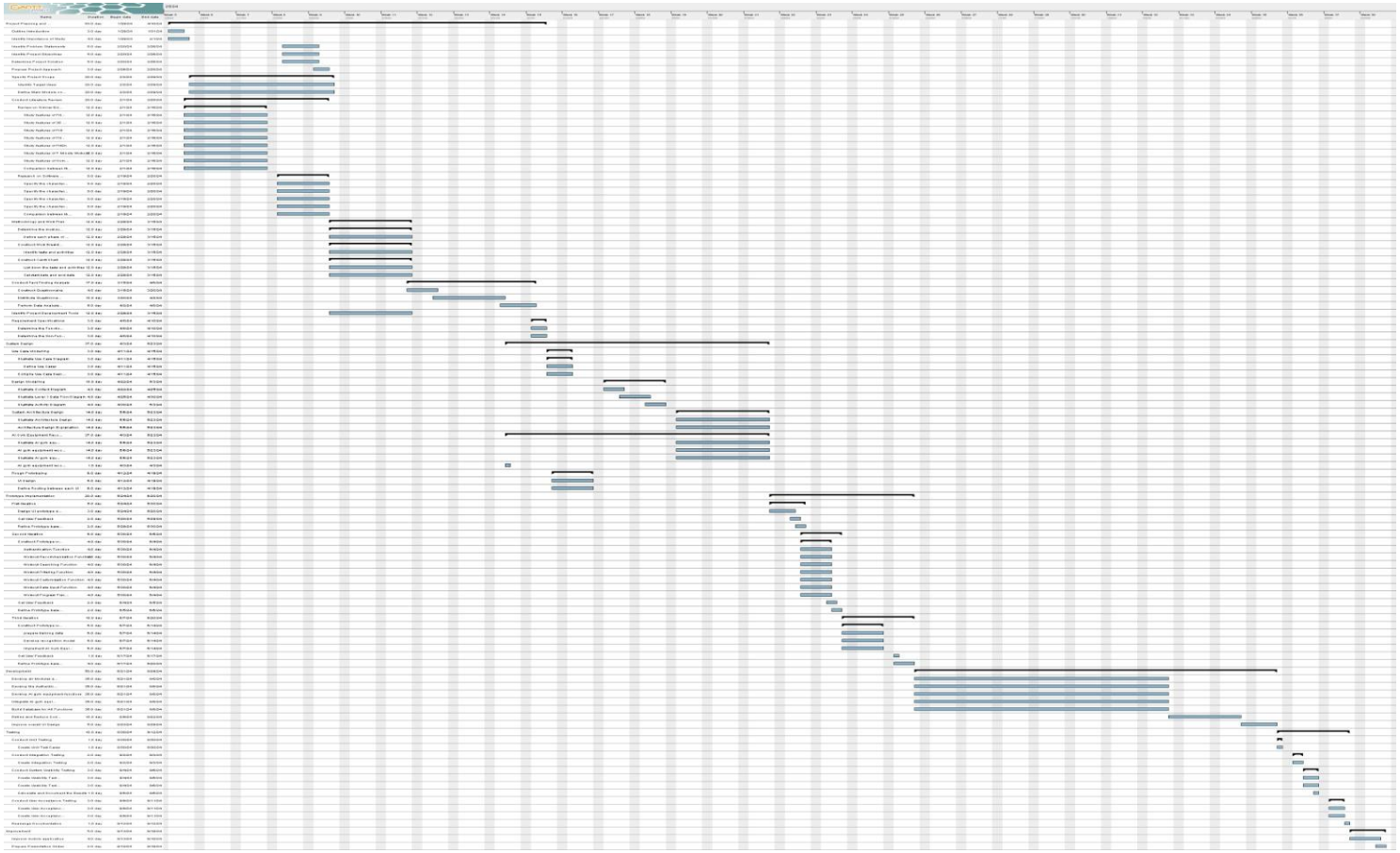


Figure 3.2: Gantt Chart

3.4 Project Development Tools

This project will make use of a number of development tools, including React Native, Python, Node.js, Flask, Visual Studio Code, Expo Go and PhpMyAdmin. These tools aim to facilitate the development process and optimize the performance of mobile application.

3.4.1 React Native

React Native is an open-source software development framework that are created by Meta (formerly Facebook) for developing Android mobile applications. Additionally, JavaScript is the core language of React Native and comes with a JavaScript library for creating user interfaces. It offers a fundamental collection of native, platform-neutral components including View, Text, and Image that correspond to the native user interface building blocks of the platform. React components allows developers to increase work efficiency by using JavaScript and React's declarative UI paradigm to communicate with native APIs and wrap pre-existing native code. Lastly, React Native support a rapid refresh features where developers may receive immediate feedback on code modifications such as UI adjustments.

3.4.2 Python

Python is a high-level, straightforward and easy to understand language which is very popular among developers. Due to its simple basic syntax, Python promotes readability and increase productivity since there is no complexity of code. Moreover, this language has a wide range of library and promote code reuse due to its module and packages availability

3.4.3 Node.js

Node.js is an open-source and cross-platform JavaScript runtime environment. It allows developers to create dynamic and scalable network applications by utilizing JavaScript for server-side and client-side code. Moreover, Node.js eliminates the need to manage thread concurrency, utilizes event-driven architecture which allows it to handle several requests at once with great efficiency.

3.4.4 Flask

Flask is a lightweight framework that allow developers to create app effortlessly with the help of Flask libraries. It is widely used in mobile application integration to handle AI related tasks via HTTP request due to its simplicity and flexibility.

3.4.5 Visual Studio Code

Visual Studio Code is a source code editor that allow developers to refine and optimize code for building applications. It supports a comprehensive range of programming languages and framework. In this project, visual studio code is used to create and edit source code for the mobile application.

3.4.6 Expo Go

React Native will collaborate with the Expo Go to launch the native mobile application on Android devices. With Expo Go, developers can exposed to the screen, navigation, user interface and component of the mobile application. Moreover, it allows developers to view changes in their code immediately, making it a crucial tool for quick mobile development.

3.4.7 PhpMyAdmin

PhpMyAdmin is a free software administration tools that is designed to manage and work on databases such as MySQL and MariaDB. It supports numerous operations such as CRUD and queries including column, table and relations that allow users to execute. In this project, MySQL and phpMyAdmin are used to manage the databases.

CHAPTER 4

PROJECT INITIAL SPECIFICATION

4.1 Introduction

This chapter mainly devoted to the project specification which covers both functional and non functional requirements, illustration of use case to show the interaction between user and system as well as use case descriptions to explain how users will carry out tasks on the system in detailed.

4.2 Facts Finding

In this section, the results of data analysis were discussed according to the data collected from questionnaires. This survey included 40 participants in total with age categories of teenagers, adults and seniors, comprising of different gender.

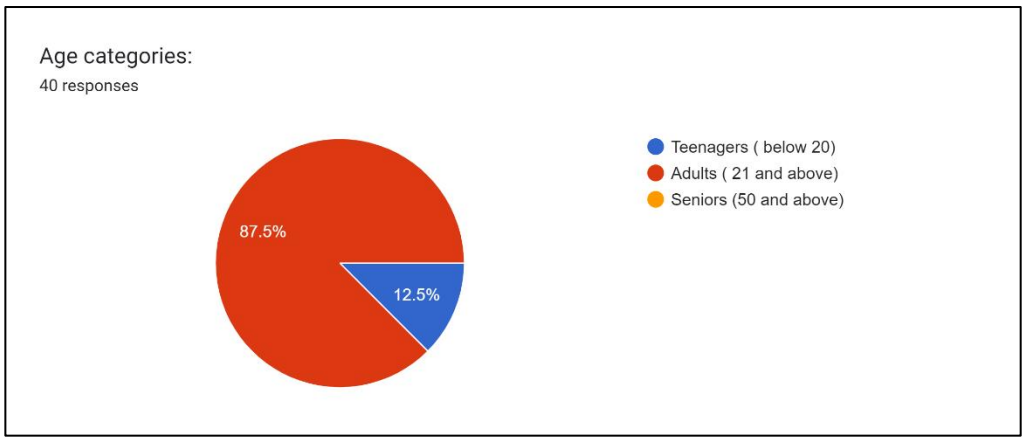


Figure 4.1: Age categories of respondents

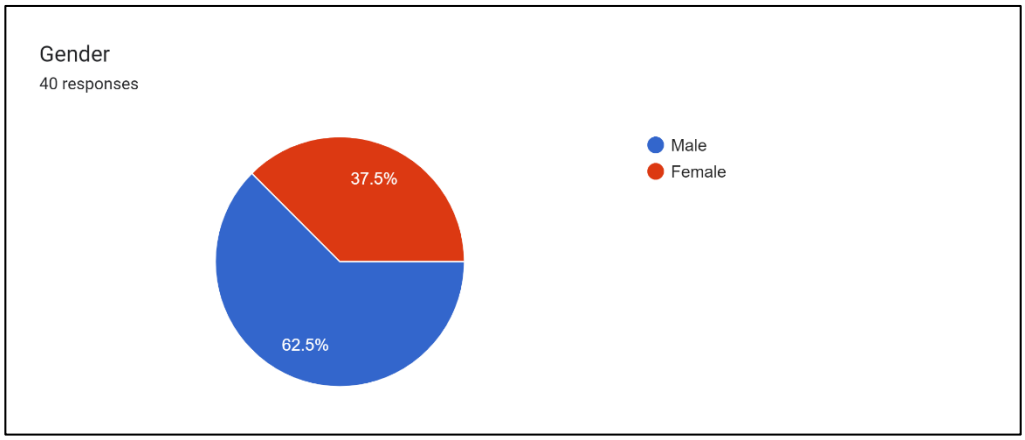


Figure 4.2: Gender of respondents

There are 2 questions to gather respondent's personal information. As shown in the figure 4.1, 87.7% of the respondents fall under age categories of adults with age 21 and above, 12.5% under teenagers with age below 20. None of the respondents fall under senior categories with age 50 and above. Among these respondents, 62.5% were male respondents and 37.5% were female respondents as seen in the figure 4.2.

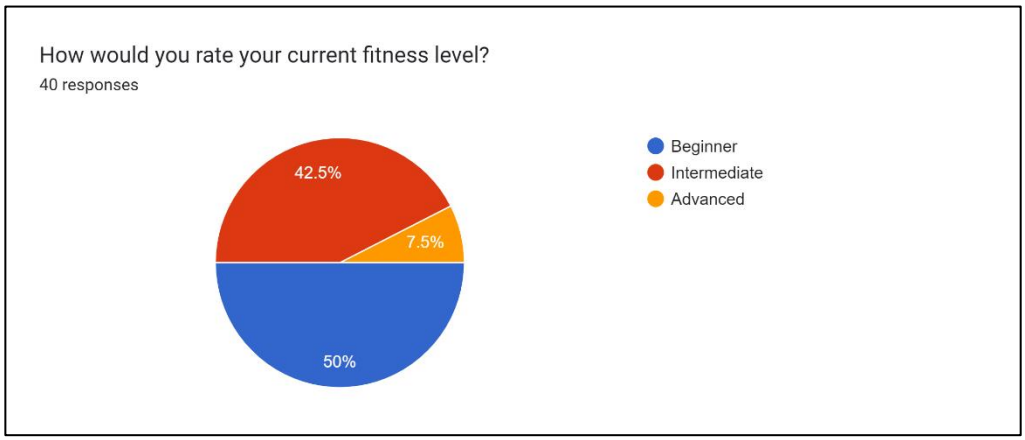


Figure 4.3: Fitness level of respondents

The third to twelfth questions of the survey mainly focused on the preferences and requirements of the respondents towards the application. The third question aimed to determine respondent's current fitness levels that were categorized into beginner, intermediate and advanced. According to figure 4.3, it was evident that the majority of the respondents (50 %) were at beginner level, while 42.5 % of the respondents were at intermediate level , and only 7.5% were at advanced level. As a result, it could be seen that beginner level was the current fitness level among all the respondents.

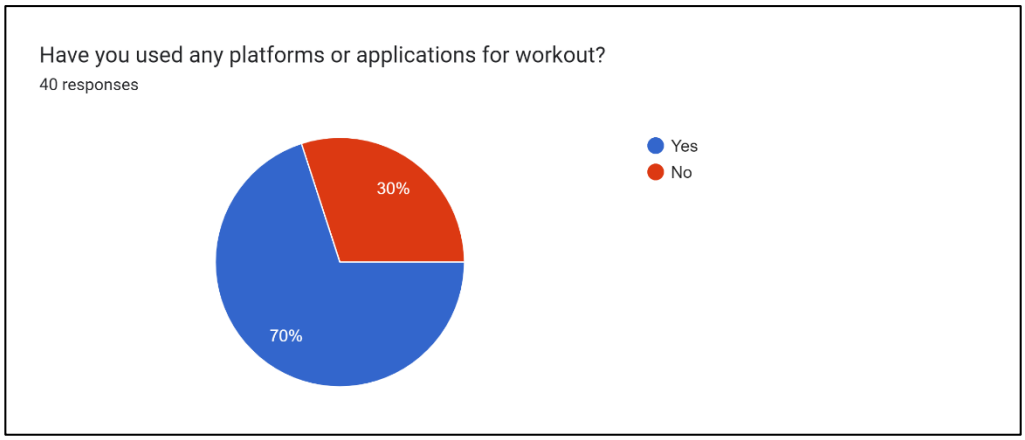


Figure 4.4: Platform or application used for workout among respondents.

By referring to figure 4.4, 70% of respondents claimed that they had used any platform or applications for workout purposes while the rest of 30% of respondents did not utilize any of them. Hence, it could be seen that application or platform utilization is essential for respondents in working out.

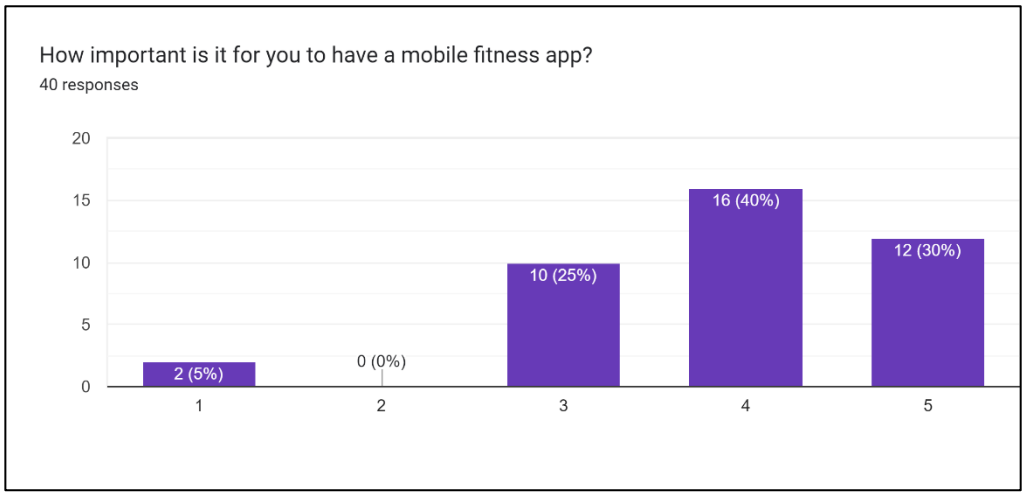


Figure 4.5: Importance of mobile fitness app for respondents.

The next question aimed to determine the importance of utilizing mobile fitness application among respondents. Based on the figure 4.5, the majority respondents (40%) rated the importance at the scale of 4 to express the importance of having mobile fitness app for workout purposes. The scale was followed by 12%, 10%, 2% and 0% of respondents which believed that the

mobile fitness application was not important at all. Therefore, it could shown that mobile fitness app was important for respondents.

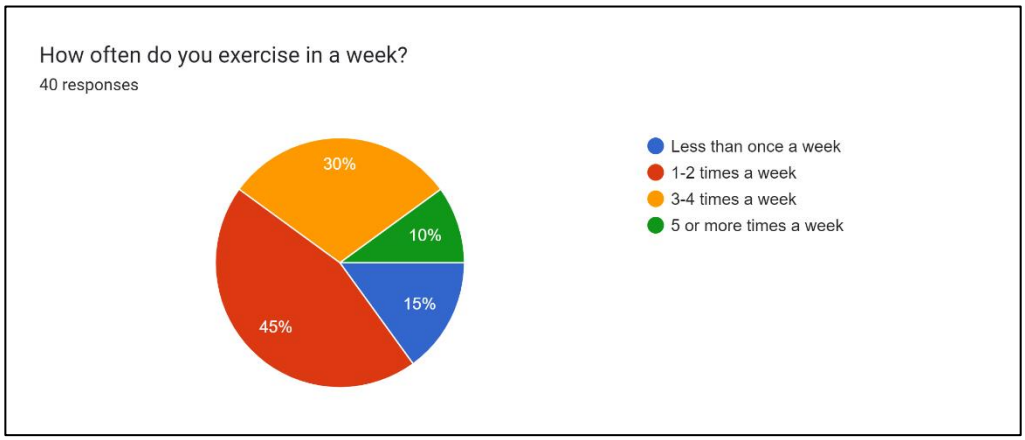


Figure 4.6: Frequency of exercise weekly among respondents.

The sixth question mainly aimed to gather information on repondents exercises frequency in a week. By referring to the result shown in figure 4.6, it was seen that 45.5% of respondents exercise 1-2 times a week, followed by 30% of respondents that exercises 3-4 times a week, 15% of respondents that exercised less than once a week and 10% of respondents that exercised 5 or more times a week. Hence, the conclusion made shown that the majority of the respondents prefer exercise 1-2 times a week.

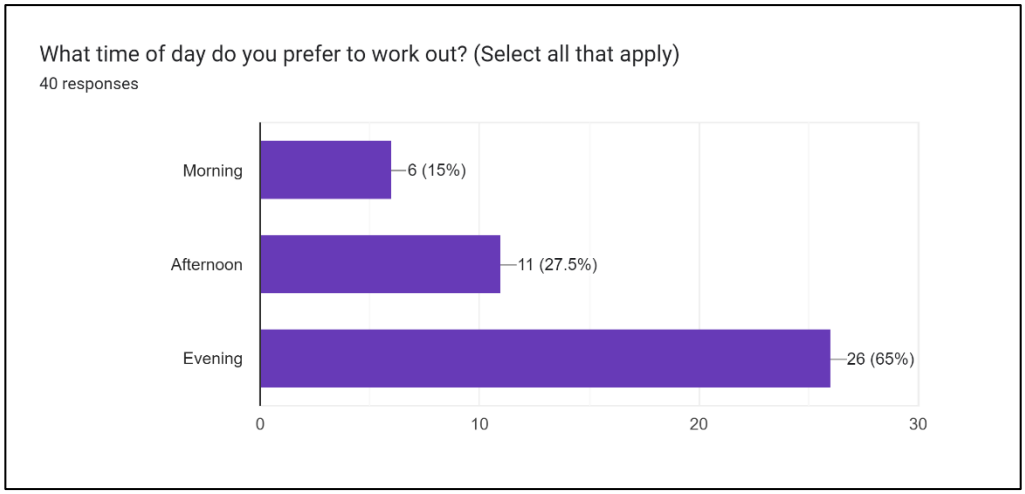


Figure 4.7: Preferred day to workout among respondents.

The next questions aimed to determine the preferred time to workout among respondents. According to the results in figure 4.7, the majority of

respondents (65%) voted evening as their preferred time for workout, followed by afternoon and morning, which took up only 27.5% and 15% of the respondents. Hence, it could be summarised that respondents preferred evening as their workout day.

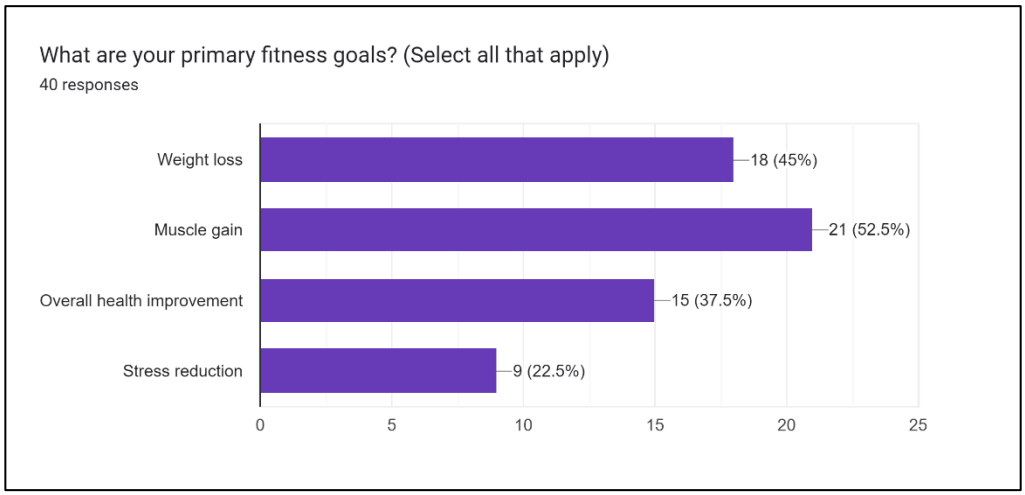


Figure 4.8: Primary fitness goals of respondents.

The following questions were mainly focused on the primary fitness goals of respondents. According to figure 4.8, it could be observed that 52.5% of the respondents consider muscle gain as their primary fitness goals, following by 45% of weight loss , 37.5% of overall health improvement and 22.5% of stress reduction. Therefore, it could be concluded that the muscle gain was the main option to be considered as primary fitness goals among respondents.

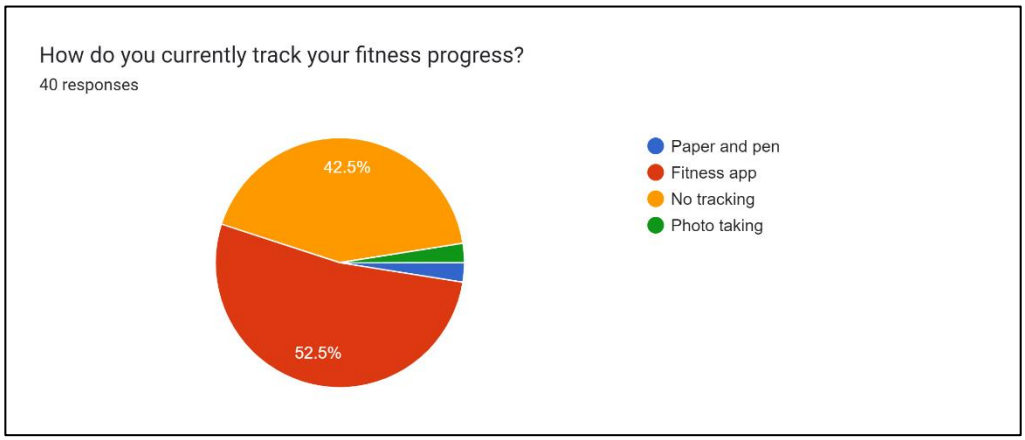


Figure 4.9: Fitness progress tracking method within respondents.

This question aimed to determine respondent's preferred fitness progress tracking method to track their fitness progress. As seen in figure 4.9, 52.5% of respondents indicated that they used fitness app as their fitness progress tracking method, while 42.5% of respondents reported that they did not track their progress. Moreover, none of the respondents utilized the photo tracking method and paper and pen method to track their fitness progress.

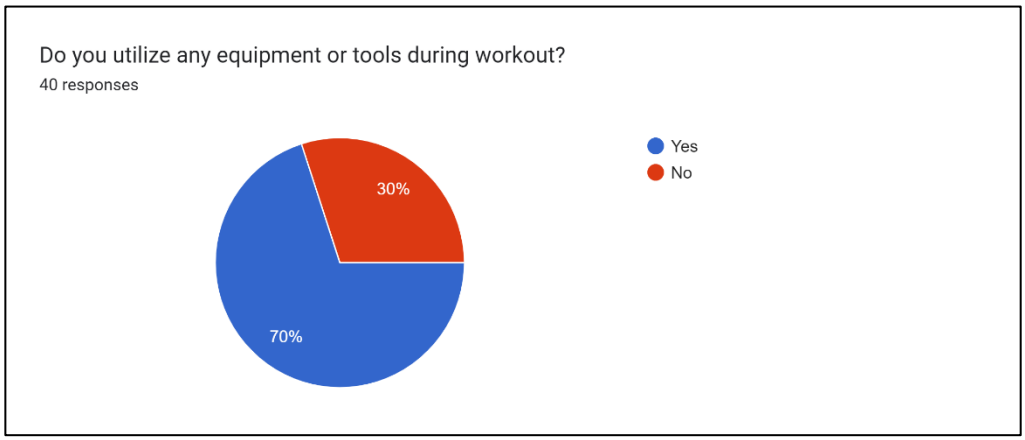


Figure 4.10: Utilization of equipment or tools during workout among respondents.

The subsequent question focused on indicating respondents on any utilization of equipment or tools during their workout sessions. As depicted in figure 4.10, 70% of respondents reported utilizing equipment or tools during workout session while 30% do not. Hence, it was apparent that most of the respondents workout with equipment or tools.

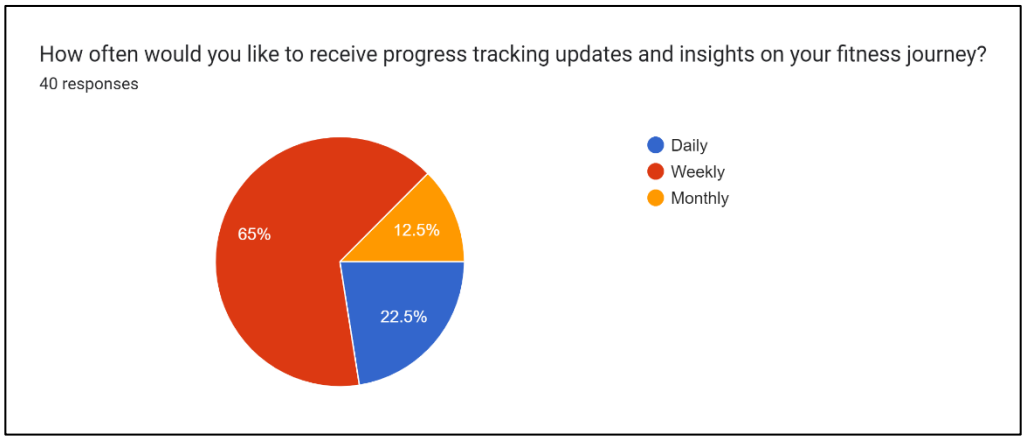


Figure 4.11: Progress tracking updates and insights among respondents.

The next question indicated on how often would respondents like to receive progress tracking updates and insight along the fitness journey. As depicted in figure 4.11, 65% of respondents choosed weekly, follow by 22.5% of daily and 12.5% of monthly. As a result, it was reported that weekly progress tracking updates and insight would be the most optimum choice among respondents.

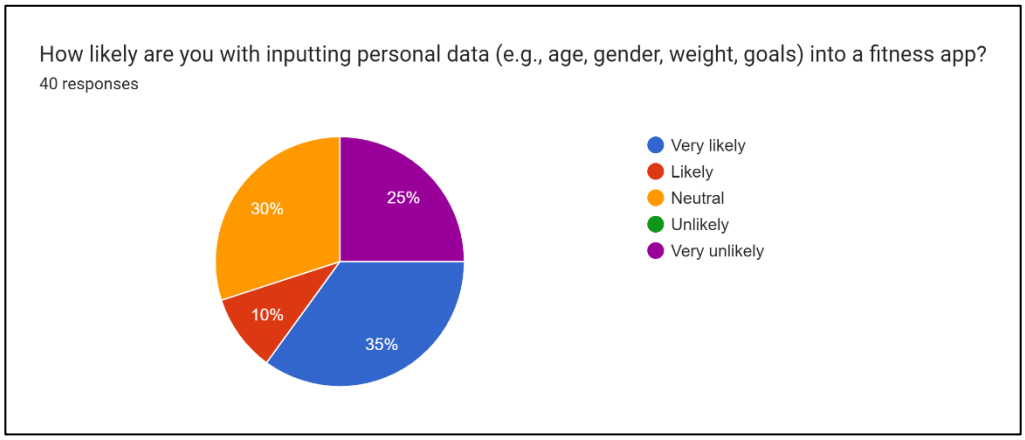


Figure 4.12: Personal data input among respondents.

The last question of this section focused on how likely respondents would be by inputting personal data into a fitness application. According to figure 4.12, it had reported that majority of the respondents (35%) choosed very likely, followed by 30% of neutral, 25% of very unlikely and 10% of likely. None of the respondents choosed unlikely. Therefore, this indicated that respondents were likely to input personal data into fitness application for a more accurate results.

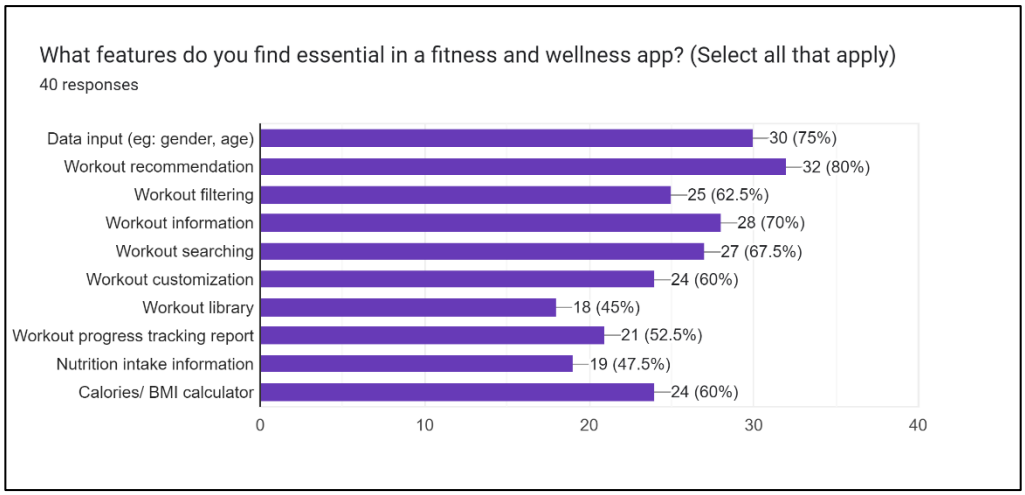


Figure 4.13: Essential features in fitness application among respondents.

The last section of the survey contained five questions that was mainly focused on the usability and functionality of the application. Based on figure 4.13, it was apparent that the most essential features among the respondents were the workout recommendation, followed by data input, workout information, workout searching, workout filtering, workout customization and calories/BMI calculator, workout progress tracking report, nutritional intake information and lastly workout library.

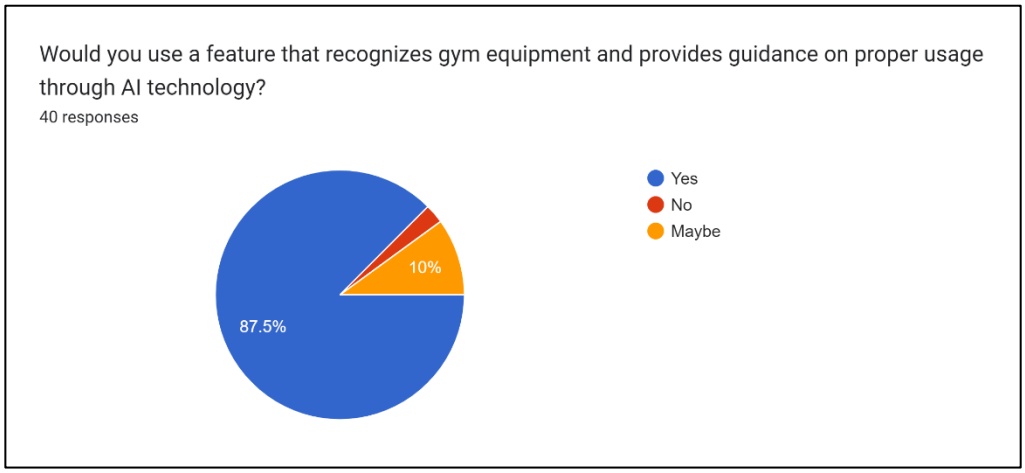


Figure 4.14: AI gym equipment recognition features utilization within respondents.

The subsequent question aimed to gather respondent's feedback on the utilization of AI gym equipment recognition features. As shown in figure 4.14, 87.5% of respondents voted that they would utilize this feature that aided in

recognizing gym equipment and providing guidance through AI technology .10% of respondents voted maybe and 2.5% of respondents voted no. As a result, this could be concluded that the majority of the respondents were willing to accept and utilized this AI gym equipment recognition feature during their workout session.

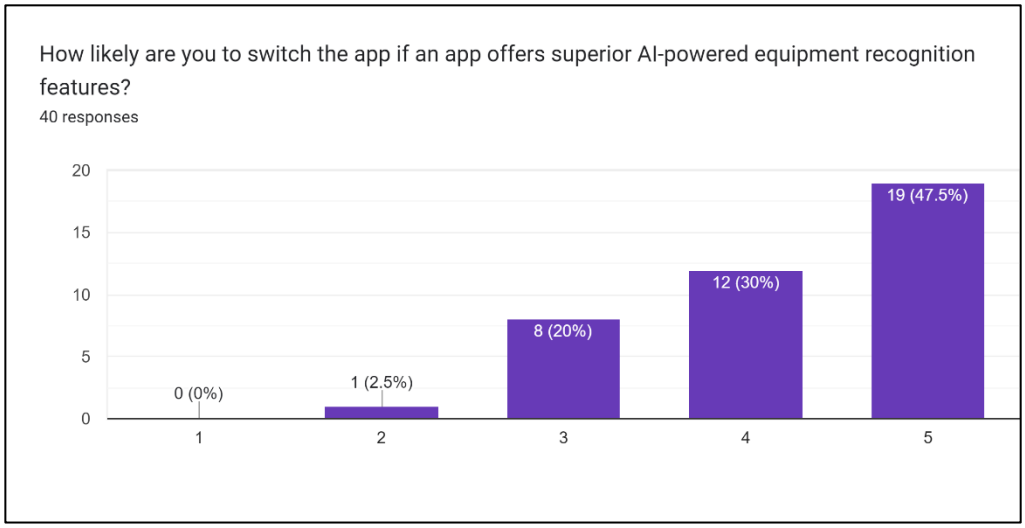


Figure 4.15: App switching

The next question focused on how likely the respondents would be in switching the app if an app offered superior AI-powered equipment recognition features. By referring to the results in Figure 4.15, it could be reported that 47.5% of the respondents rated very likely at the scale of 5, followed by 30%, 20%, 2.5% and 0% . Hence, this indicated that respondents were very likely to switch app due to the superior AI gym equipment recognition features.

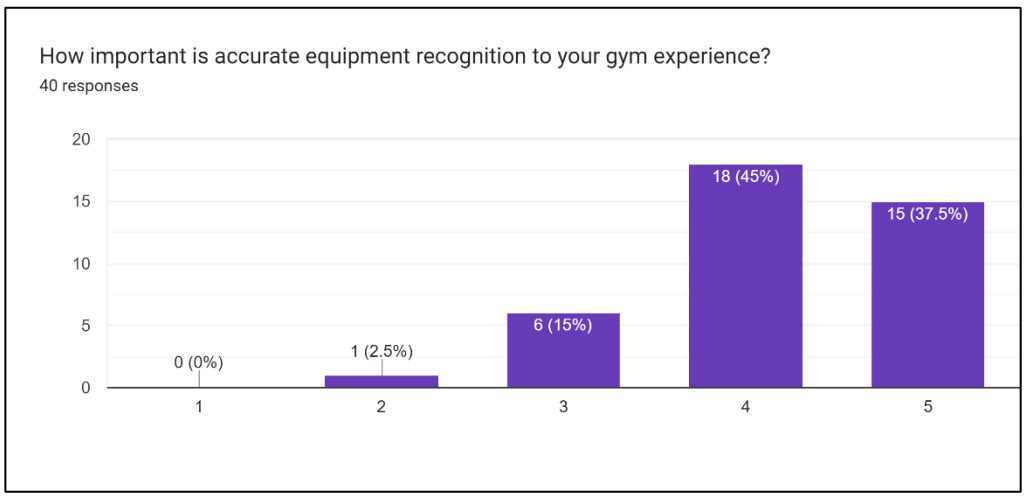


Figure 4.16: Importance of accuracy in gym equipment recognition among respondents.

This question aimed to collect respondent's information on the importance of gym equipment recognition accuracy. As shown in the figure 4.16, it is seen that 45%% of the respondents rated very important at the scale of 4, followed by 37.5%, 15%, 2.5% and 0%. Therefore, this proves that accuracy in gym equipment recognition was important to respondent's gym experience.

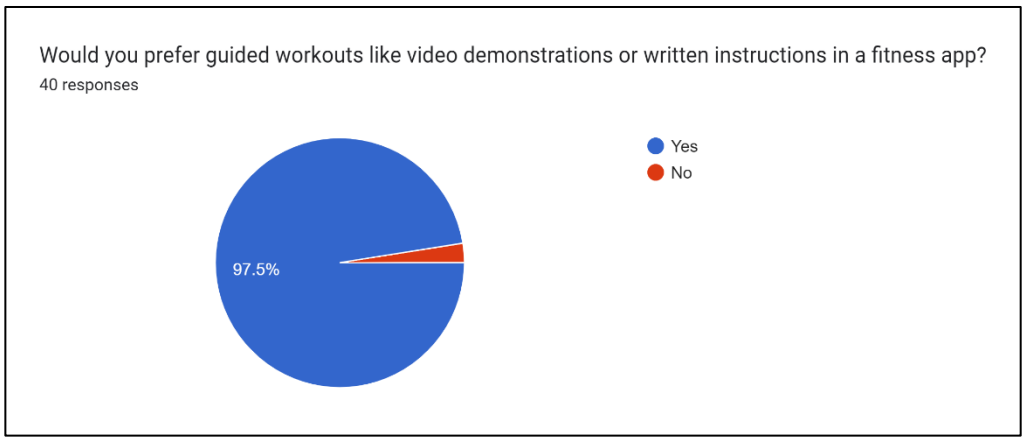


Figure 4.17: Availability of guided workouts among respondents.

The last question of this section aimed to determine whether respondents would like to prefer guided workouts like video demonstrations or written instructions in the fitness app. As depicted in the figure 4.17, 97.5% of respondents prefer the availability of guided workouts while the rest of 2.5% do not prefer. As a result, this indicated that the guided workout should exist in the fitness app to ease respondents in workout out effectively.

4.3 Functional Requirements

The functional requirements for mobile fitness application are listed as below:

1. The app shall allow existing users to log in with their email and password.
2. The app shall allow new users to sign up for an account by using name, email, password, gender and age.

3. The app shall allow users to enter data input for personalized workout solutions.
4. The app shall generate personalized workout solutions as workout recommendation to user based on user input.
5. The app shall allow user to filter and display filtered workout results based on various criteria.
6. The app shall display detailed workout information to user.
7. The app shall allow users to create, edit or delete workout customization.
8. The app shall provide access to user on workout library with comprehensive exercises and workouts.
9. The app shall allow users to search functionality with relevant keywords.
10. The app shall track and display user's workout progress over time to generate progress report to user.
11. The app shall allow users to input their data by using BMI calculator in order to calculate body composition index.
12. The app shall allow users to input their data by using calories calculator in order to calculate daily calories needs.
13. The app shall display nutritional intake information on daily recommended nutrient intake to users.
14. The app shall display user profile information.
15. The app shall allow users to edit data input in user profile.

The functional requirements for AI gym equipment recognition:

1. The gym equipment recognition shall display name, guidance or instructions to user after scanning.

4.4 Non- Functional Requirements

The non-functional requirements for mobile fitness application are outlined as below:

1. Integrity: The app shall validate user input by display error message if incorrect input format was given.
2. Usability: The app shall have an appealing, user-friendly, simple-to-use, and understandable interface as well as a clear navigation.

3. Performance: The app shall be quick and highly responsive upon user request.
4. Reliability: The app shall be operated continuously over time without any crashing or interruption.
5. Flexibility: The app shall allow multiple users to access simultaneously.

4.5 Use Case Diagram

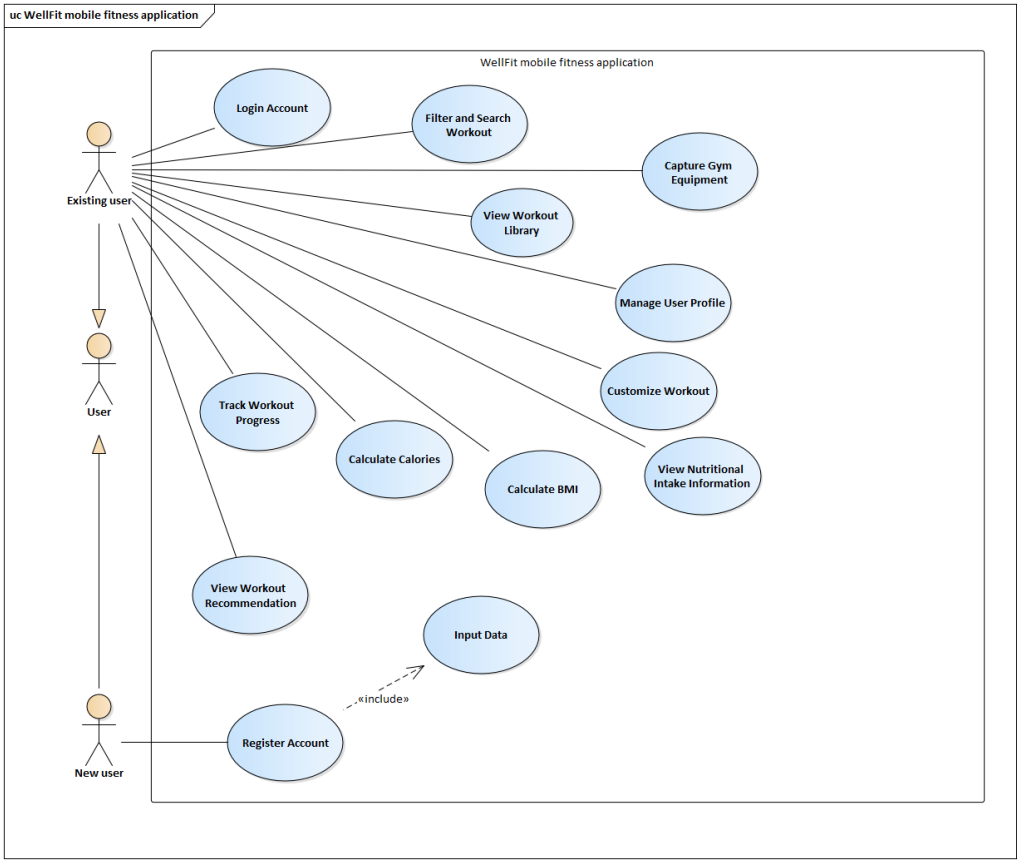


Figure 4.18: Use Case Diagram

4.6 Use Case Description

Use Case Name: Register Account	ID: UC01	Importance Level: High
Primary Actor: New User	Use Case Type: Detailed, Essential	

Stakeholders and Interests: New user –wants to register their account on the mobile fitness application.
Brief Description: This use case describes how new user register an account via the mobile fitness application.
Trigger: When a new user wants to register for new account in the mobile fitness application.
Relationships: Association: New User Include: Input Data Extend: - Generalization: -
Normal Flow of Events: 1. A new user clicks on the "Register" button at the navigation page of the application to register a new account. 2. The system allow new users to enter required information such as name, email, password, gender and age for creating a new account. 3. The system validates on the input from the user. 4. Once the user input is validated, the system creates the user account based on the user information provided. 5. The system redirects users to data input page with their account registered.
Sub-flows: None
Alternative/Exceptional Flows: 1a. The system displays an error message if the account is already exists or input for name, email or password is invalid.

Table 4.1: Use Case Description – Register Account

Use Case Name: Login Account	ID: UC02	Importance Level: High

Primary Actor: Existing User	Use Case Type: Detailed, Essential
Stakeholders and Interests: Existing user – Users with existing account wants to login to their account on the mobile fitness application.	
Brief Description: This use case describes how existing user login to the account via the mobile fitness application.	
Trigger: The existing user wants to login to their account in the mobile fitness application.	
Relationships: Association: Existing User Include: - Extend: - Generalization: -	
Normal Flow of Events: 1. The system allow existing users to login to their accounts by clicking on the "Login" button displayed on the navigation page of the application. 2. The system allow existing users to enter required information such as email and password for login to the application. 3. The system validates on the user inputs. Perform subflow 3.1. 4. When the account is validated successfully, the system allows users to login to the application. 5. The user is allowed to access the home page with their account logged in.	
Sub-flows: 3.1 The system will search for user email and password in the database. 3.2 The system retrieves user records from the database.	

Alternative/Exceptional Flows: 1a. The system displays an error message if the account already exists or input for name, email or password is invalid.

Table 4.2: Use Case Description – Login Account

Use Case Name: Input Data	ID: UC03	Importance Level: High
Primary Actor: New User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user –New users who wants to input their data into the mobile fitness application.		
Brief Description: This use case describes how new users provide personal input data into the mobile fitness application for personalized workout solutions.		
Trigger: When a new user wants to input data into the mobile fitness application.		
Relationships: Association: New User Include: - Extend: - Generalization: -		
Normal Flow of Events: 1. The system prompt user to provide input for each details such as workout goals, body focus area, gender, duration, frequency and so on. 2. The system validates on the user inputs. Perform subflow 2.1. 3. The system generate a workout solution after user input based on user preferences. 4. The user is then directed to the home page to view personalized workout.		

Sub-flows: 2.1 The system will store user input in the database. 2.2 The system will retrieve user records to generate workout personalization.
Alternative/Exceptional Flows: 1a. The system displays an error message and does not proceed to next data input if the user do not provide data input for the required fields.

Table 4.3: Use Case Description: Input Data

Use Case Name: View Workout Recommendation	ID: UC04	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user –Existing users who wants to view personalized workout recommendation after data input.		
Brief Description: This use case describes how existing user can view personalized workout recommendation after data input.		
Trigger: The existing user wants to view personalized workout recommendation after data input.		
Relationships: Association: Existing User Include: - Extend: - Generalization: -		
Normal Flow of Events: 1. The system prompt existing user to workout recommendation section after data input. 2. The system allow existing user to view personalized workout recommendation after user input.		

3. The user perform the selected workout based on the workout recommendation in the mobile fitness application.
Sub-flows: None
Alternative/Exceptional Flows: None

Table 4.4: Use Case Description: View Workout Recommendation

Use Case Name: View Workout Library	ID: UC05	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user –Existing users who wants to view workout library to explore various types of exercises.		
Brief Description: This use case describes how existing user can view workout library to explore various types of exercises in mobile fitness application.		
Trigger: The existing user wants to view workout library that contains comprehensive workout exercises.		
Relationships: Association: Existing User Include: - Extend: - Generalization: -		
Normal Flow of Events: 1. The application presents the user with comprehensive workout exercises. 2. The application allow user to browse through the workout library to select specific workout. 3. The application displays detailed workout information such as instructions, motion guidance and descriptions about the selected		

workout exercises in workout library. 4. The user may perform workout based on selected workout exercises from the library.
Sub-flows: None
Alternative/Exceptional Flows: None

Table 4.5: Use Case Description – View Workout Library

Use Case Name: Filter and Search Workout	ID: UC06	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user –Existing users who wants to view information of a certain workout.		
Brief Description: This use case describes how existing users filter and search workout in the mobile fitness application.		
Trigger: The existing user wants to view workout information in the mobile fitness application.		
Relationships: Association: Existing User Include: - Extend: - Generalization: -		
Normal Flow of Events: 1. The user provide workout keywords on the searching textbox. 2. The user applies filters to the workout list. Perform subflow 2.1. 3. The user submit the empty search filter textbox. Perform subflow 3.1.		

4. The system displays a list of relevant workouts based on the search and filter options.
Sub-flows: 2.1 Users select workout choices to be filtered by clicking on the “Apply” button. 3.1 If user wish to remove the filter, user may click on “Reset” button to remove applied filter.
Alternative/Exceptional Flows: The system prompt an error message if no relevant workout is found.

Table 4.6: Use Case Description – Filter and Search Workout

Use Case Name: View Nutritional Intake Information	ID: UC07	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user – Existing users who want to view the nutritional intake information in the mobile fitness application.		
Brief Description: This use case describes how existing users view the nutritional intake information.		
Trigger: The existing user wants to view nutritional intake information as daily nutritional intake in the mobile fitness application.		
Relationships: Association: Existing User Include: - Extend: - Generalization: -		
Normal Flow of Events: 1. The user clicks on the “Nutrition” button to view the detailed information.		

2. The system redirects user to the nutritional intake information page after user clicks on the button. 3. The system displays the detailed information of the nutritional intake.
Sub-flows: None
Alternative/Exceptional Flows: None

Table 4.7 : Use Case Description – View Nutritional Intake Information

Use Case Name: Customize Workout	ID: UC08	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user – Existing users who want to customize workout according to their own preferences.		
Brief Description: This use case describes how existing users customize the workout solutions.		
Trigger: The existing user wants to customize the workout according to their own preferences in the mobile fitness application.		
Relationships: Association: Existing User Include: - Extend: - Generalization: -		
Normal Flow of Events: 1. The user clicks on the “Customize” button to customize workout. 2. The system redirects user to workout customization page after user clicks on the button. 3. The system displays users with option to customize their workout		

solutions. 4. The users create specific exercises and adjust parameters for the workout customization. 5. The user finalizes and save the customized workout by clicking on "Save" button. Perform subflow 5.1
Sub-flows: 5.1The customized workout is saved into the database.
Alternative/Exceptional Flows: If users wish to edit their customized workout, they may modify the saved workout for adjustment. If users wish to delete their customized workout, they may click on "Delete" button.

Table 4.8: Use Case Description - Customize Workout

Use Case Name: Calculate BMI	ID: UC9	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user – Existing users who want to calculate their body composition index with BMI calculator.		
Brief Description: This use case describes how existing users calculate the BMI in the mobile fitness application.		
Trigger: The existing user wants to calculate their body composition index in the mobile fitness application.		
Relationships: Association: Existing User Include: - Extend: - Generalization: -		

Normal Flow of Events: 1. The system prompts user to choose measurement unit for weight and height. 2. The system prompts user to input their weight and height. 3. The user enters their weight and height into the BMI calculator. 4. The system calculates the user's BMI according to user's weight and height input. 5. The system displays the BMI chart with user's BMI index value and BMI category.
Sub-flows: None
Alternative/Exceptional Flows: 1a. If users wish to edit their weight and height, they may modify the value to recalculate the new BMI. 1b. The system will prompt error message if user's weight and height input are invalid.

Table 4.9: Use Case Description - Calculate BMI

Use Case Name: Calculate Calories	ID: UC10	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Existing user – Existing users who want to calculate their daily calories intake with calories calculator.		
Brief Description: This use case describes how existing users calculate the daily calories intake in the mobile fitness application.		
Trigger: The existing user wants to calculate their daily calories intake in the mobile fitness application.		

Relationships: Association: Existing User Include: - Extend: - Generalization: -
Normal Flow of Events: 1. The system prompts user to choose measurement unit for weight and height. 2. The system prompts user to input their weight and height, age, gender and activity level. 3. The user enters their input into the calories calculator. 4. The system calculates the user's daily calories needed according to user's input provided. 5. The system displays users with the total daily calories intake information.
Sub-flows: None
Alternative/Exceptional Flows: 1a. If users wish to edit their input for gender, weight, height, age and activity level, they may modify the value to recalculate the new daily calories intake. 1b. The system will prompt error message if user's input are invalid.

Table 4.10: Use Case Description - Calculate Calories

Use Case Name: Track Workout Progress	ID: UC11	Importance Level: High
Primary Actor: Existing User	Use Case Type: Detailed, Essential	

Stakeholders and Interests: Existing user – Existing users who want to track their workout progress over time.
Brief Description: This use case describes how existing users track and monitor their workout progress over time in mobile fitness application.
Trigger: The existing user wants to track their workout progress over time in mobile fitness application.
Relationships: Association: Existing User Include: - Extend: - Generalization: -
Normal Flow of Events: 1. The users click on "Workout Progress" button to redirect users to the workout progress tracking page. 2. The system allows users to view the workout progress tracking. Perform subflow 2.1 3. The system provide analyse according to user's data and generates report . 4. The user view their workout progress details that are displayed in metrics.
Sub-flows: 2.1 User's data is stored and retrieved from the database for analyzation and report.
Alternative/Exceptional Flows: None

Table 4.11: Use Case Description - Track Workout Progress

Use Case Name: Capture Gym Equipment Image	ID: UC12	Importance Level: High

Primary Actor: Existing User	Use Case Type: Detailed, Essential
Stakeholders and Interests: Existing user – Existing users who want to capture gym equipment image.	
Brief Description: This use case describes how existing users capture gym equipment image for AI recognition.	
Trigger: The existing user wants to capture gym equipment image.	
Relationships: Association: Existing User Include: - Extend: - Generalization: -	
Normal Flow of Events: 1. The system allow users to capture gym equipment image by clicking on the "Camera" button. 2. The system analyse and detect equipment according to user's scanning to provide gym equipment name. 3. The users view the gym equipment that are displayed in the system after recognition is successful.	
Sub-flows: None	
Alternative/Exceptional Flows: The system will prompt a message if invalid detection of gym equipment.	

Table 4.12: Use Case Description - Capture Gym Equipment Image

Use Case Name: Manage User Profile	ID: UC13	Importance Level: High

Primary Actor: Existing User	Use Case Type: Detailed, Essential
Stakeholders and Interests: Existing user – Existing users who want to manage the view user profile information in the mobile fitness application.	
Brief Description: This use case describes how existing users manage the user profile information.	
Trigger: The existing user wants to manage user profile information in the mobile fitness application.	
Relationships: Association: Existing User Include: - Extend: - Generalization: -	
Normal Flow of Events: 1. The user clicks on the "User Profile" button to view the detailed information. 2. The system redirects user to the user profile information page after user clicks on the button. 3. The system displays the detailed information of the user profile. 4. If users wish to edit user profile information, perform sub flow 4.1.	
Sub-flows: 4.1 The users click on the input for user profile information to make changes. 4.2 The system updates on the relevant information after users click on "Save" button.	
Alternative/Exceptional Flows: 1a. The system will prompt error message if user's input are invalid.	

Table 4.13: Use Case Description – Manage User Profile

4.7 User Interface (UI) Prototype

This section defines the UI interface design of the prototype, note that although all the features are shown in the figure, but their functionality is still in the early stages of development. This prototype's main goal is to create the user interface and allow users to navigate between them. The features and their functionality were implemented in the next revisions.

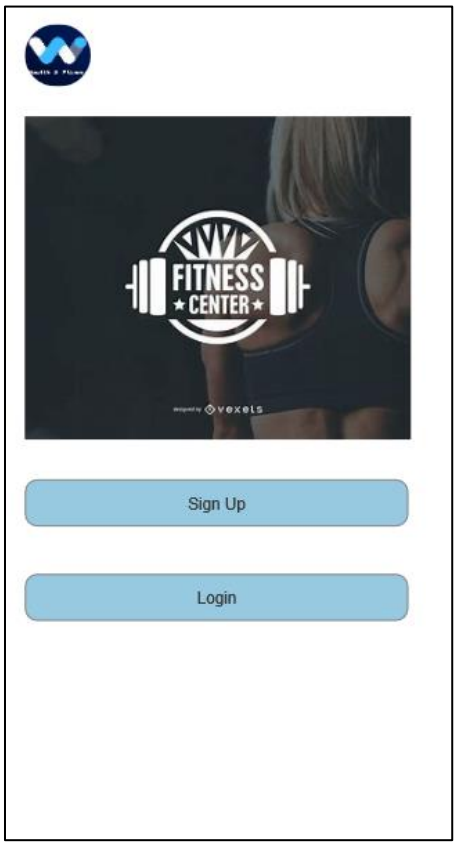


Figure 4.19: Prototype of Main Page UI

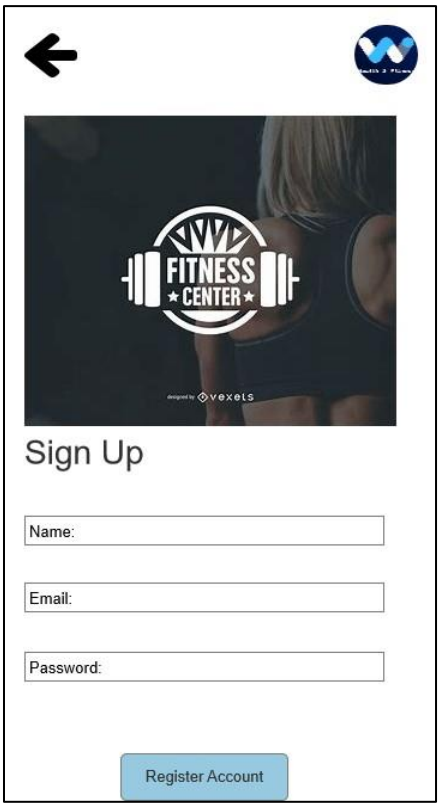


Figure 4.20: Prototype of Sign Up Page UI

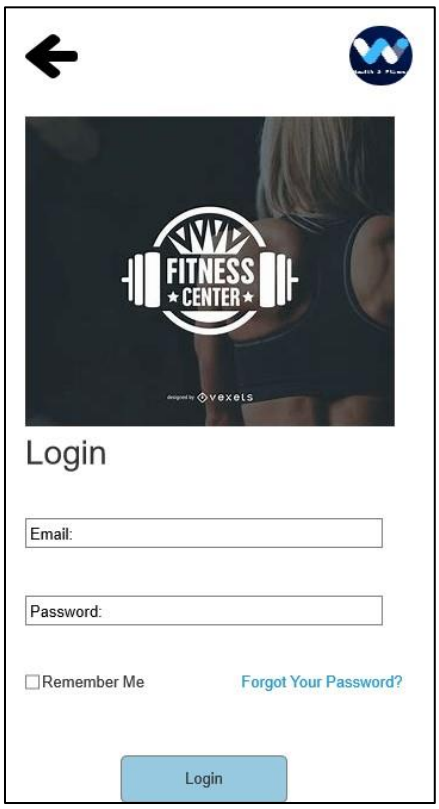
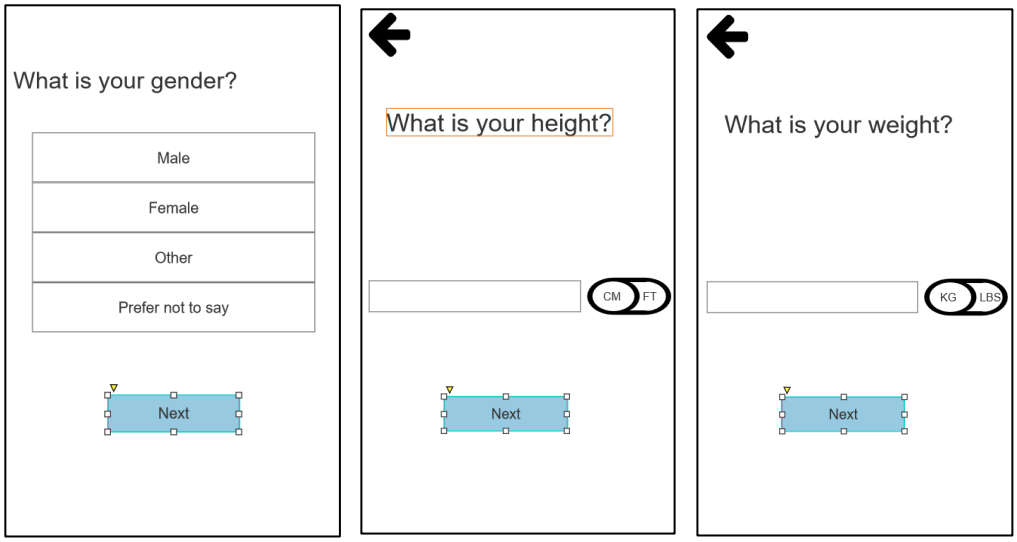
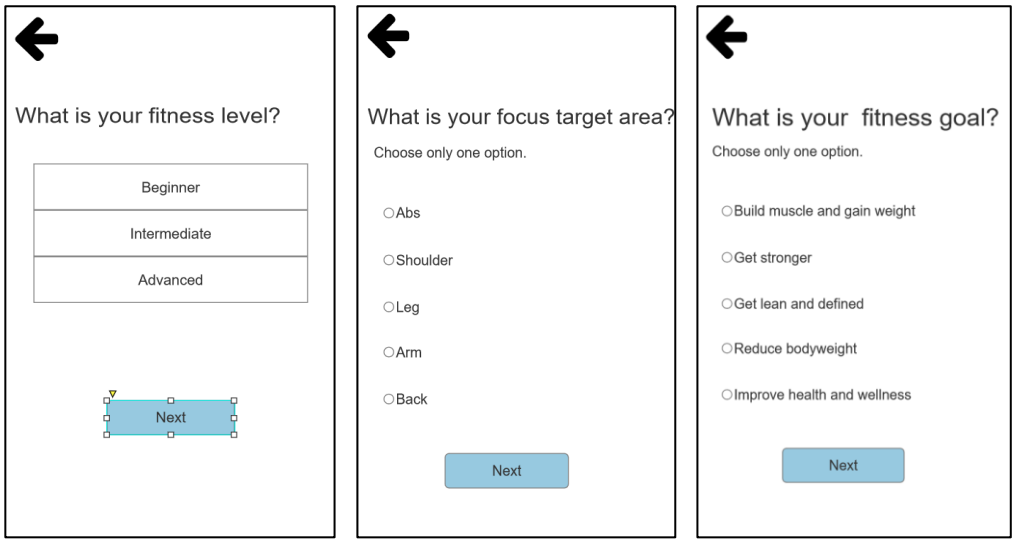


Figure 4.21: Prototype of Login Page UI



(a) (b) (c)



(d) (e) (f)

Figure 4.22(a)(b)(c)(d)(e)(f): Prototype of Data Input Pages UI

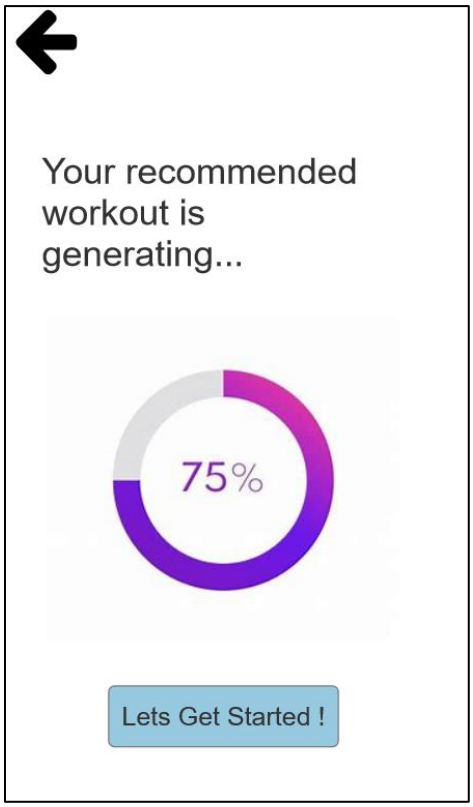


Figure 4.23: Prototype of Recommendation Workout Generation based on Data Input Page UI

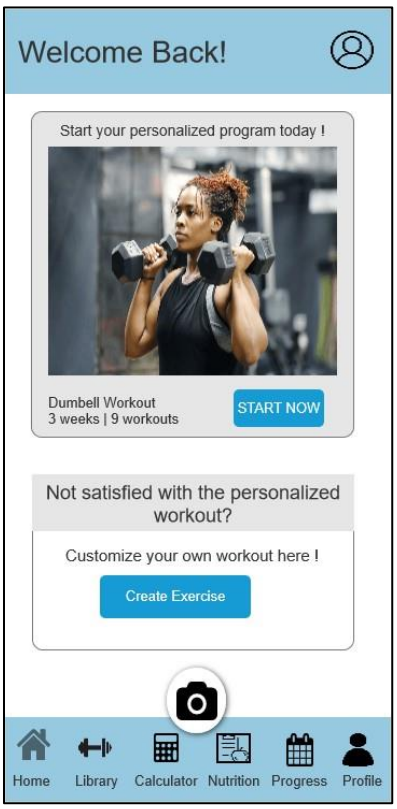
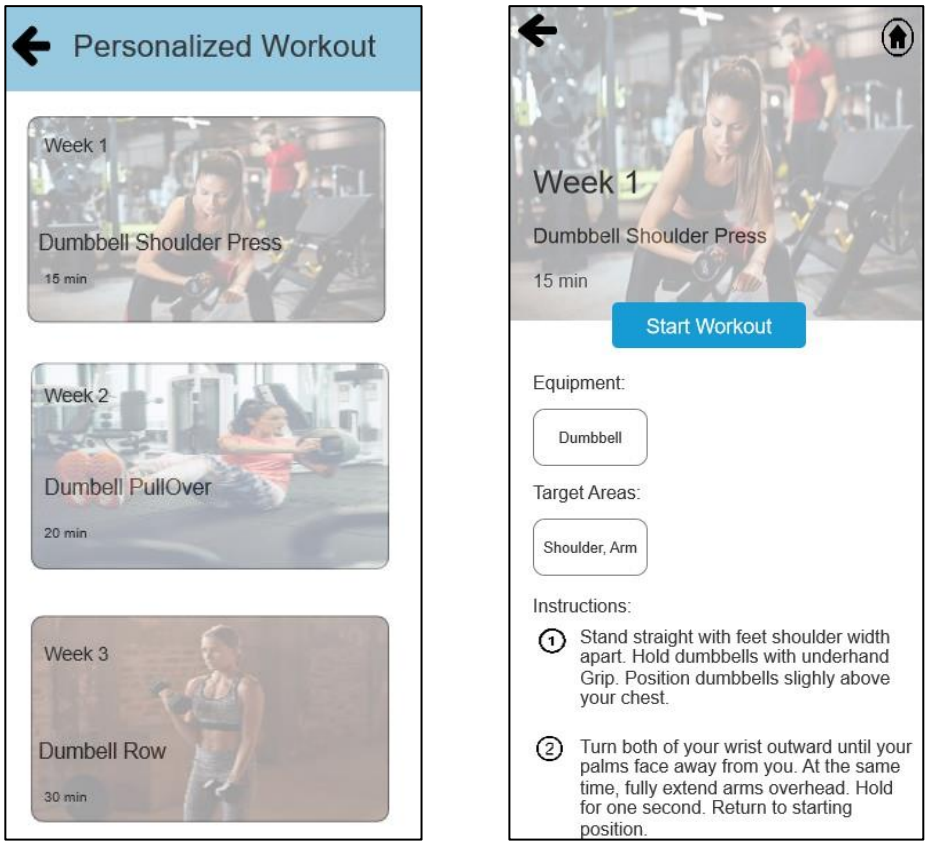


Figure 4.24: Prototype of Home Page UI



(a) (b)

Figure 4.25(a)(b): Prototype of Workout Recommendation Pages UI

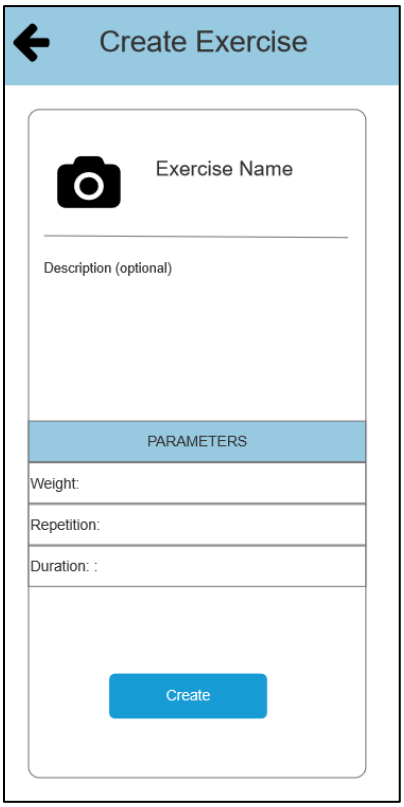
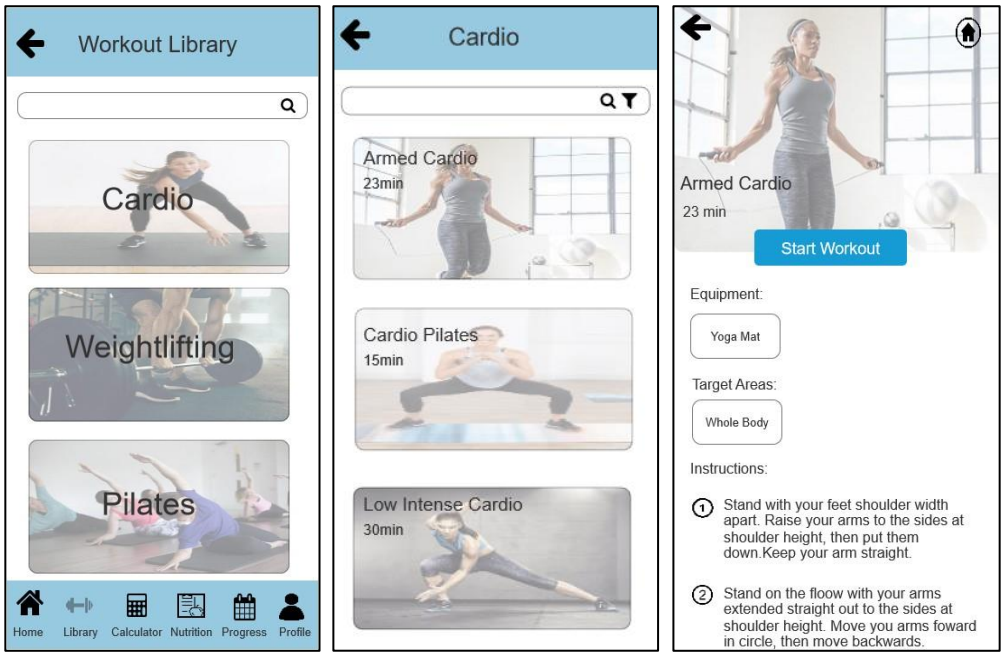


Figure 4.26: Prototype of Workout Customization Page UI



(a) (b) (c)

Figure 4.27(a)(b)(c): Prototype of Workout Library Pages UI

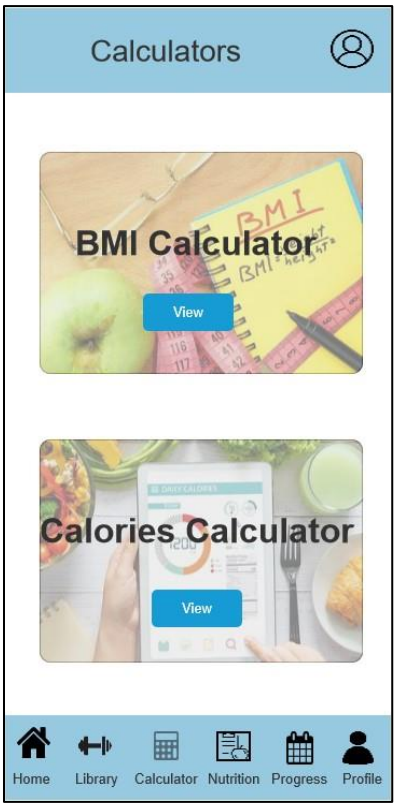
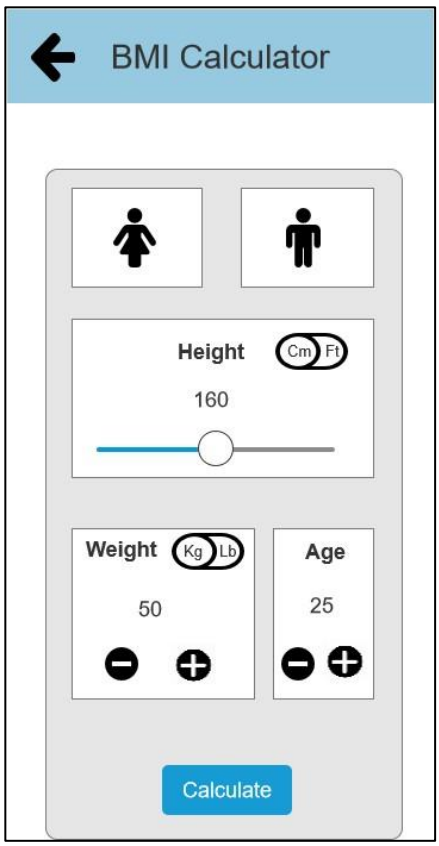
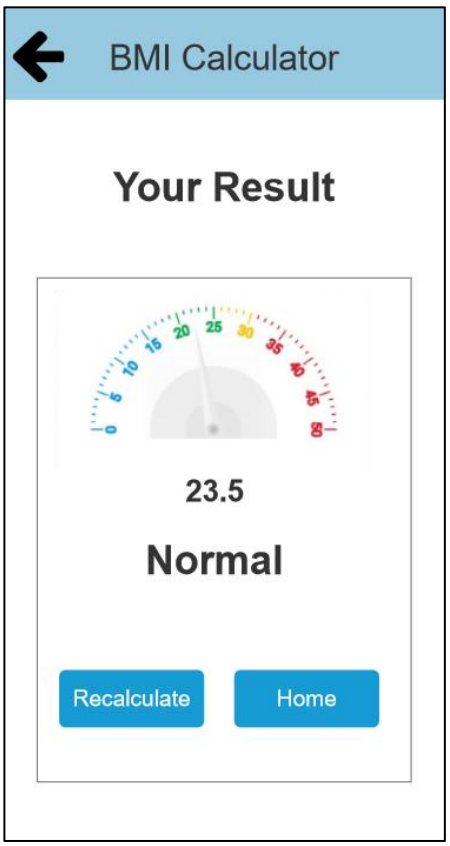


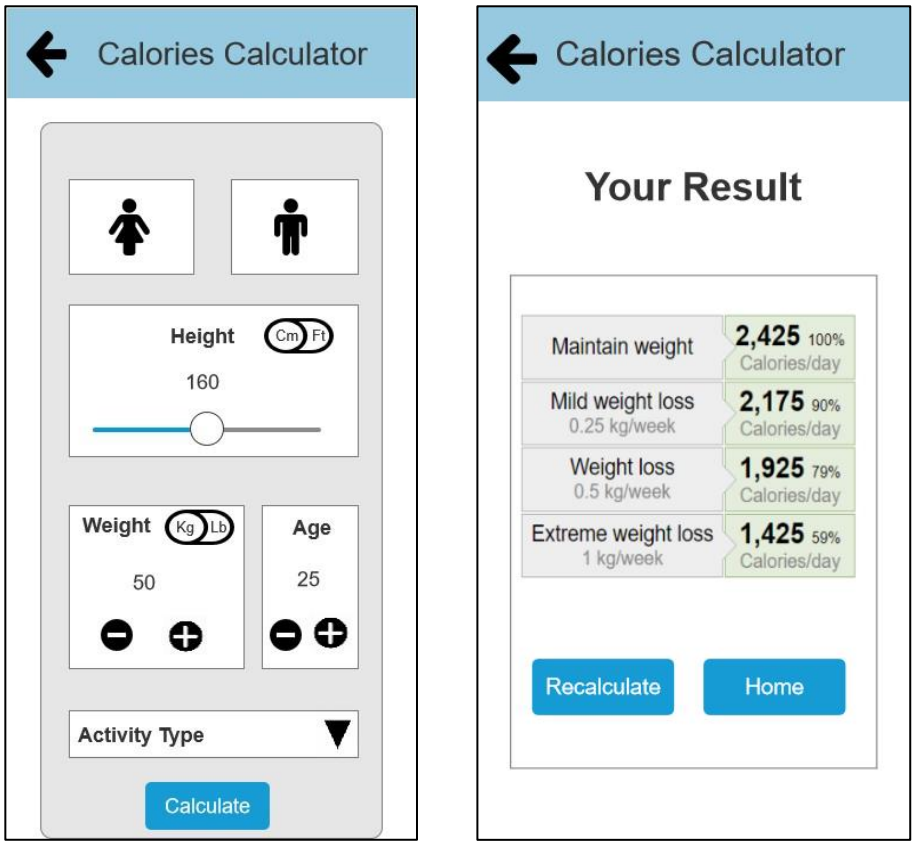
Figure 4.28: Prototype of Calculator Page UI





(a) (b)

Figure 4.7.29(a)(b): Prototype of BMI Calculator Pages UI



(a) (b)

Figure 4.7.30(a)(b): Prototype of Calories Calculator Pages UI

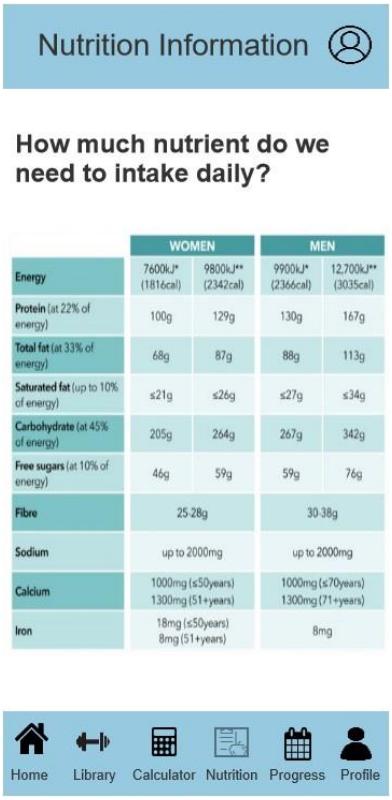


Figure 4.31: Prototype of Nutrition Intake Information Page UI

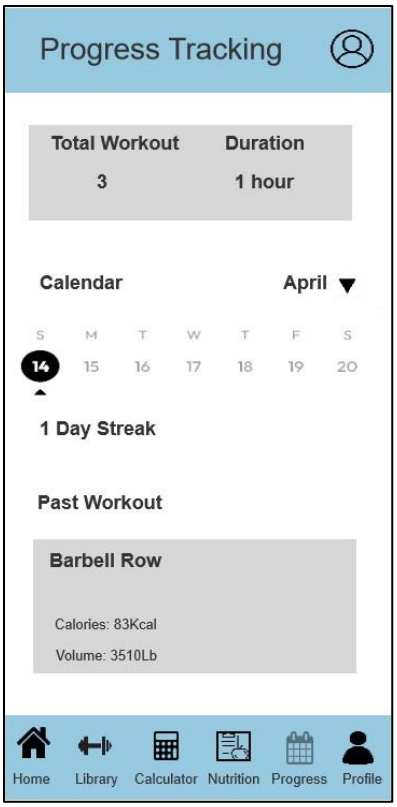
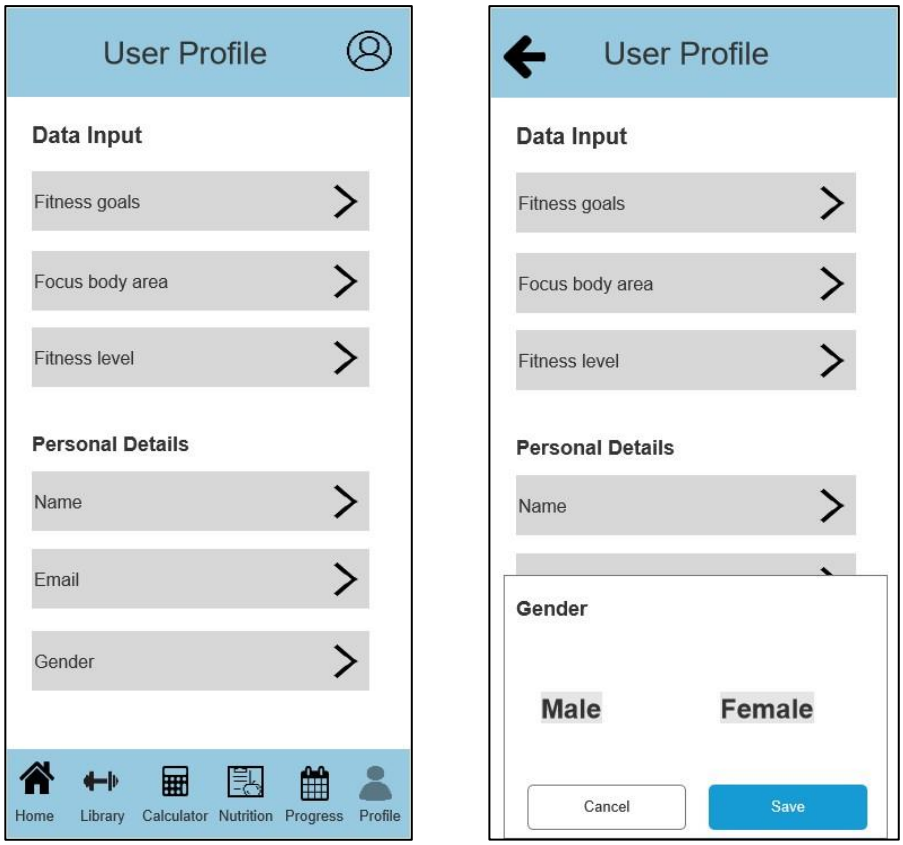
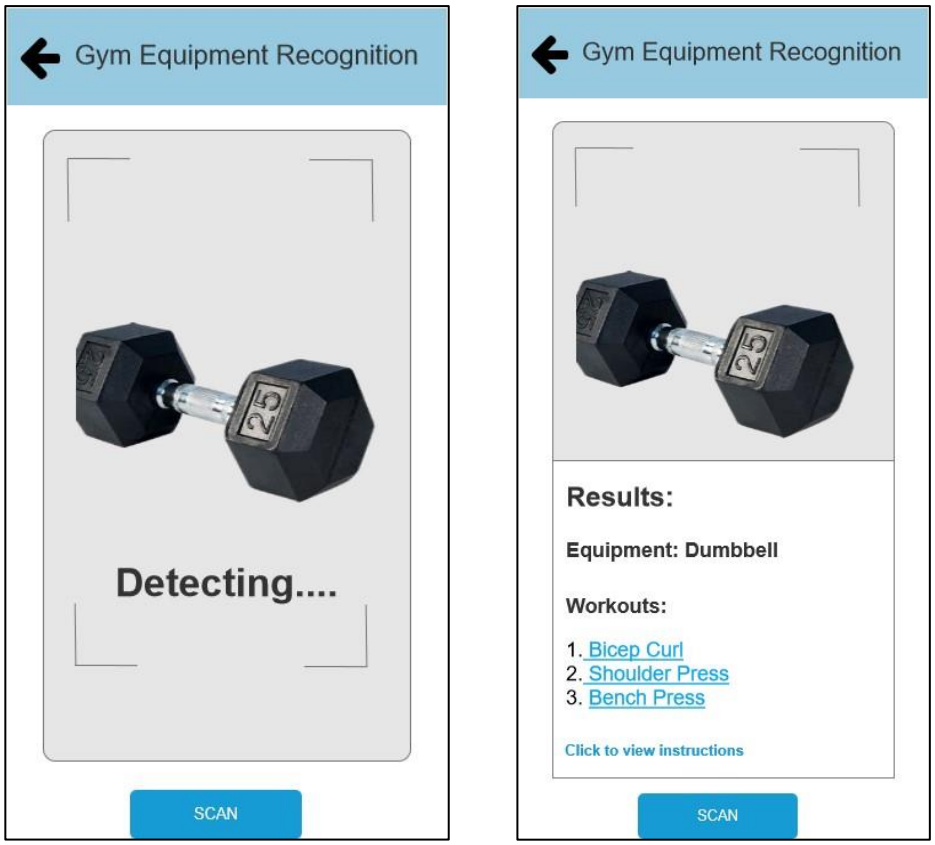


Figure 4.32: Prototype of Progress Tracking Page UI



(a) (b)

Figure 4.33: Prototype of User Profile Page UI



(a) (b)

Figure 4.34: Prototype of AI Gym Equipment Recognition Pages UI

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

This chapter covers the system design which includes both the system architecture and database design, illustration of the data flow diagram in different levels and user interface design on the application.

5.2 System Architecture Design

This project makes use of a three-tier architecture in the system architecture design. It comprises of presentation layer, business logic layer and data layer. The topmost layer is the presentation layer, which handles user interface interaction by gathering and displaying information to the business logic layer for processing. Following that will be the business logic layer, which performs data processing by applying business rules and logic and computes to the data layer. The data will be stored and managed in the data layer upon the requests from the business logic layer by interacting with the database. This architecture provides substantial benefits in terms of scalability as it can be scaled without affecting the other tiers. Moreover, it ensures strong security because of its concern separation, which allows each layer to be managed independently while maintaining its own security protocol. By putting this architecture into practice, the project helps to create a strong basis for the development of fitness mobile applications. Figure 5.1 depicts the three-tier architecture of the system design.

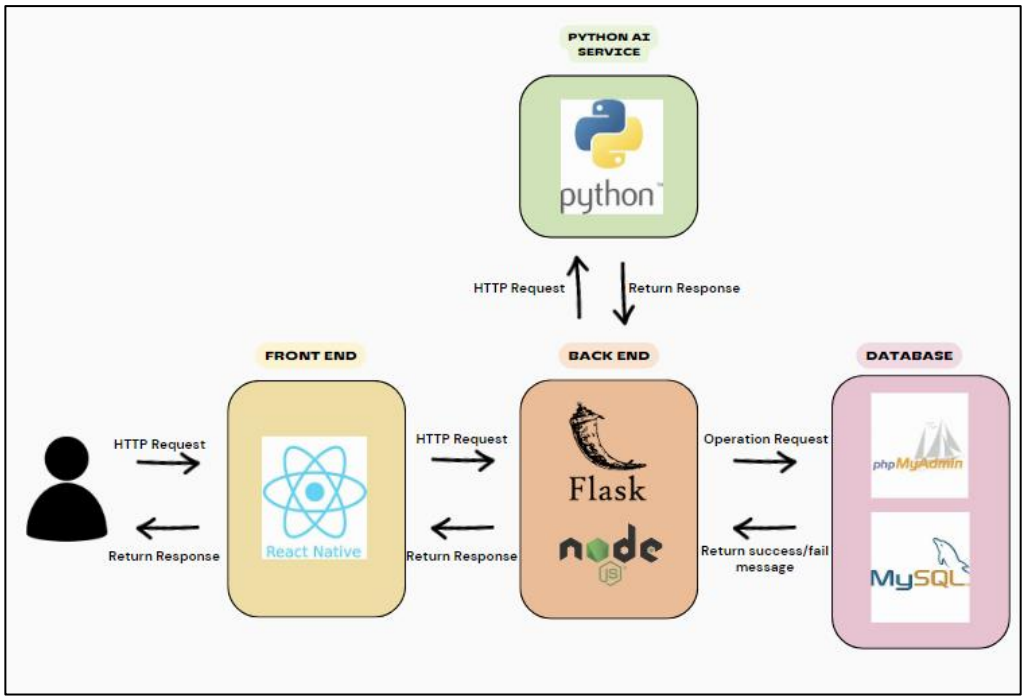


Figure 5.1: System Architecture Diagram

The fitness application's user interface is designed and developed using the React Native framework. Users interact seamlessly with the application by providing input to the back end and receiving a response as output. Both Node.js and Flask are the back-end services in the project that are utilized to handle and manage HTTP requests as well as facilitate interaction between Python AI services, databases, and React Native frontends.

The AI integration on gym equipment recognition is handled by Flask, while the Python AI service is responsible for interpreting the images captured by users and sending back the results to Flask and Node.js.Within the user interface, users can exposed to the name of scanned gym equipment by initiating a dynamic interaction with AI model to receive workout instructions based on scanned gym equipment. The AI model and the back-end services work seamlessly to ensure the successful implementation of AI recognition in the application.

For user authentication, Node.js is in charge of the validation part, making sure that only authorized users are able to access the application’s features during login and sign up. When the API sends a query, Node.js will reply to the data if it is successful or unsuccessful by providing error messages.

Last but not least, the data layer, which is mostly controlled by MySQL and PhpMyAdmin serves as a data storage, management and modification for user credentials and workout library. This layer is dedicated to communicate seamlessly with Node.js and Flask for data processing and returning the data to be displayed in the user interface. Moreover, it guarantees data security and scalability across the application, ensuring the optimal data management performance of the project.

5.3 Database Architecture Design

In this section, the entity relationship diagram is illustrated below to show the database architecture design.

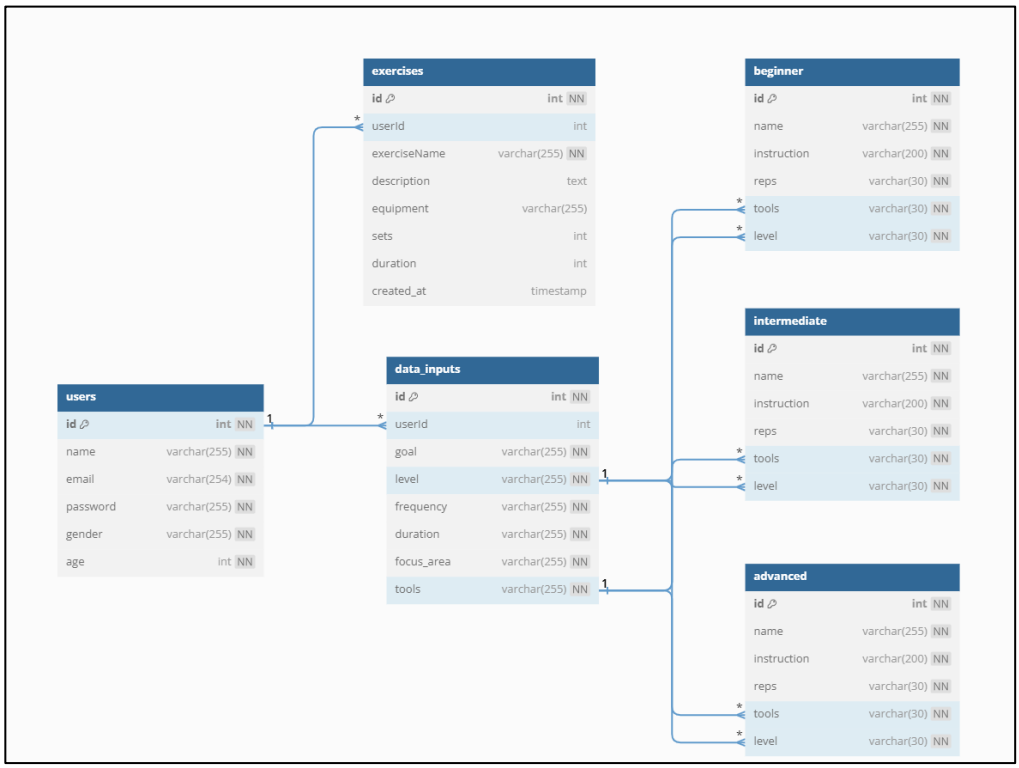


Figure 5.2: Database Architecture Design

Table Name	Description
users	Stores user details such as name, email, password, gender and age.

data_inputs	Stores data input from user, including the goal, level, frequency, duration, focus area and tools.
exercises	Stores the exercise details such as exercise name, description, equipment, set, and duration.
beginner	Stores the beginner workout details, including name, instruction, repetitions, tools and level.
intermediate	Stores the intermediate workout details, including name, instruction, repetitions, tools and level.
advanced	Stores the advanced workout details, including name, instruction, repetitions, tools and level.

Table 5.1: Database Architecture Design Table

5.4 Data Flow Diagram

5.4.1 Context Diagram

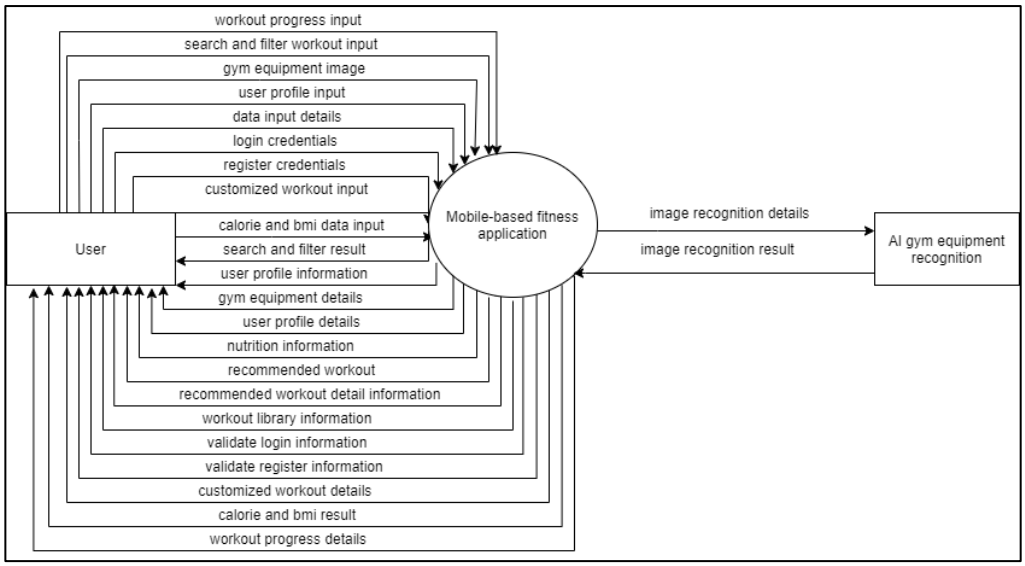


Figure 5.3: Context Diagram

5.4.2 Level 0 Data Flow Diagram

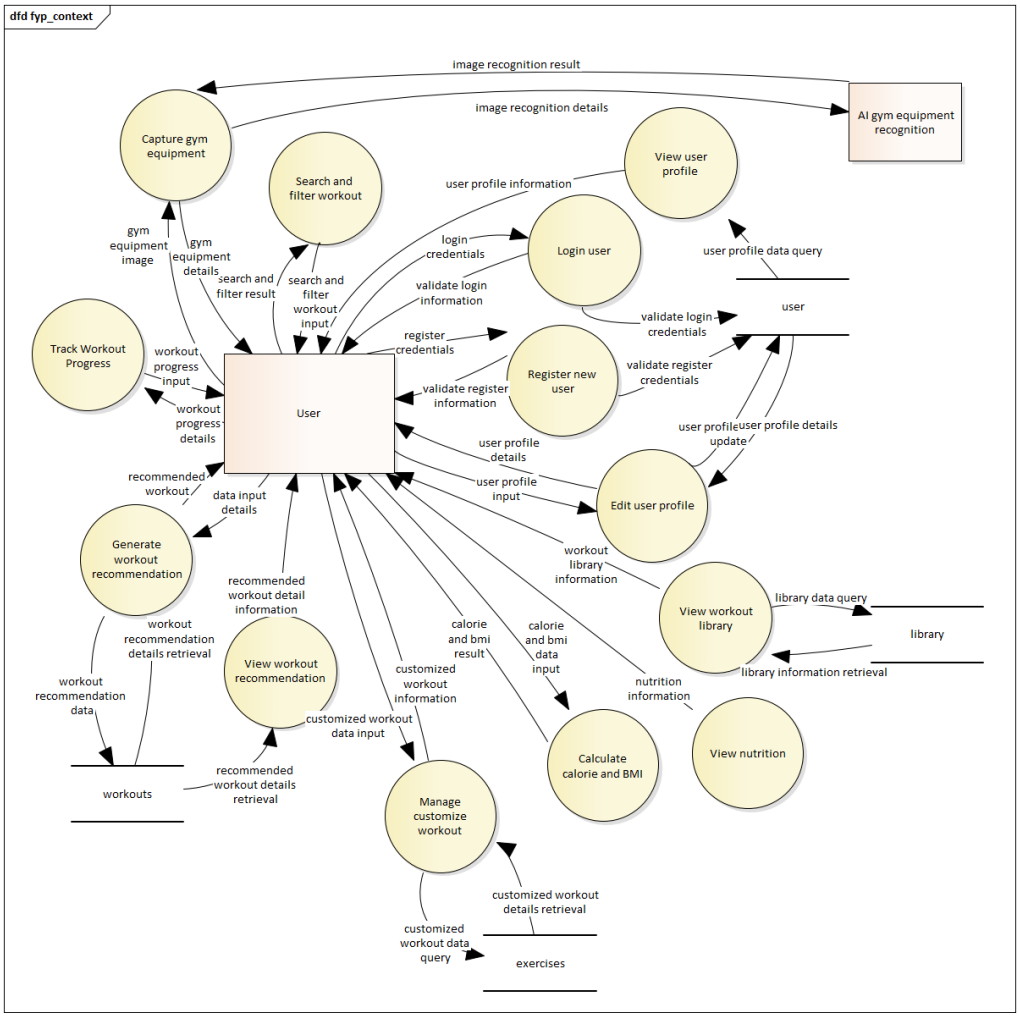


Figure 5.4: Level 0 Data Flow Diagram

5.4.3 Activity Diagram

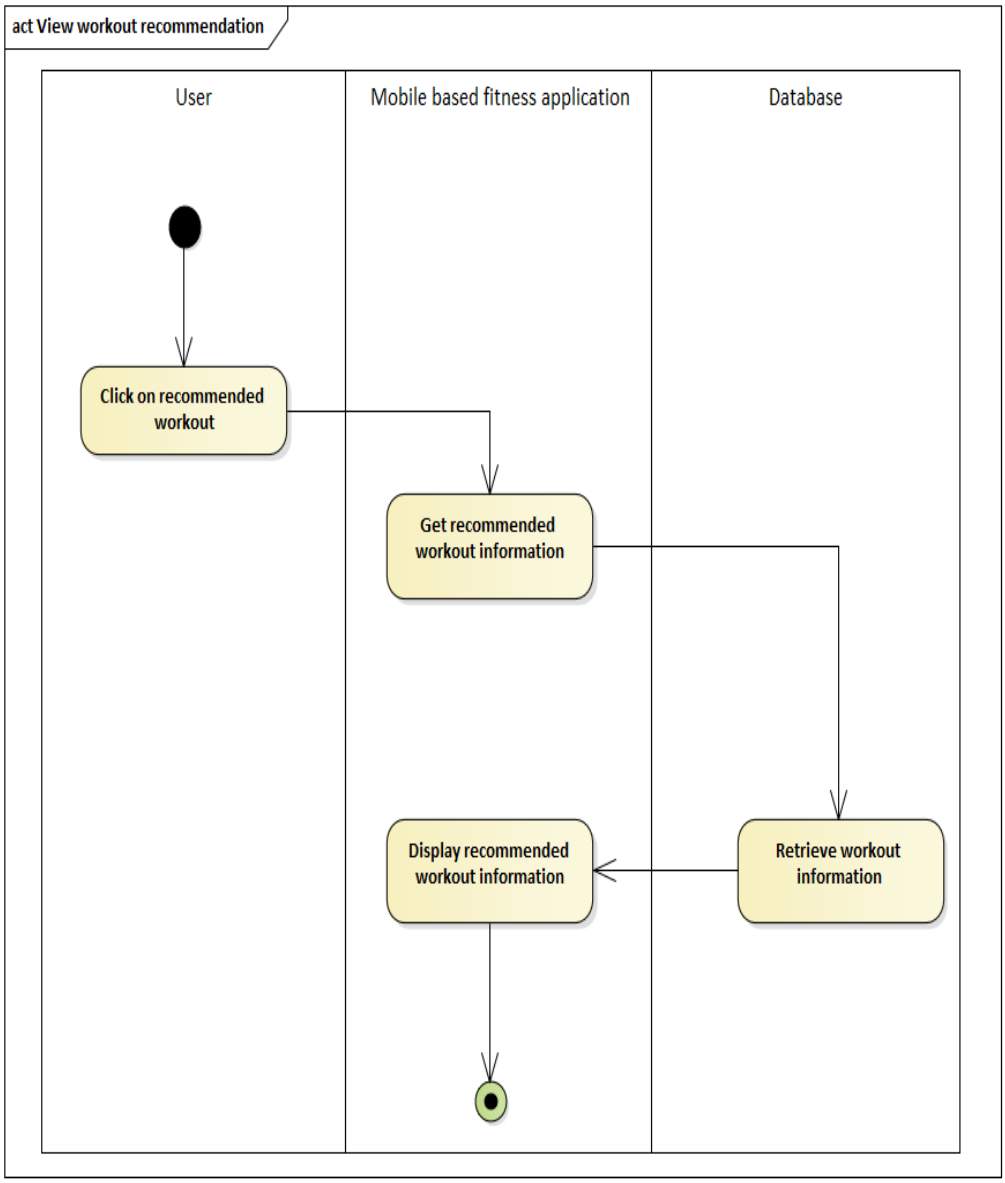


Figure 5.5: Activity Diagram – View Workout Recommendation

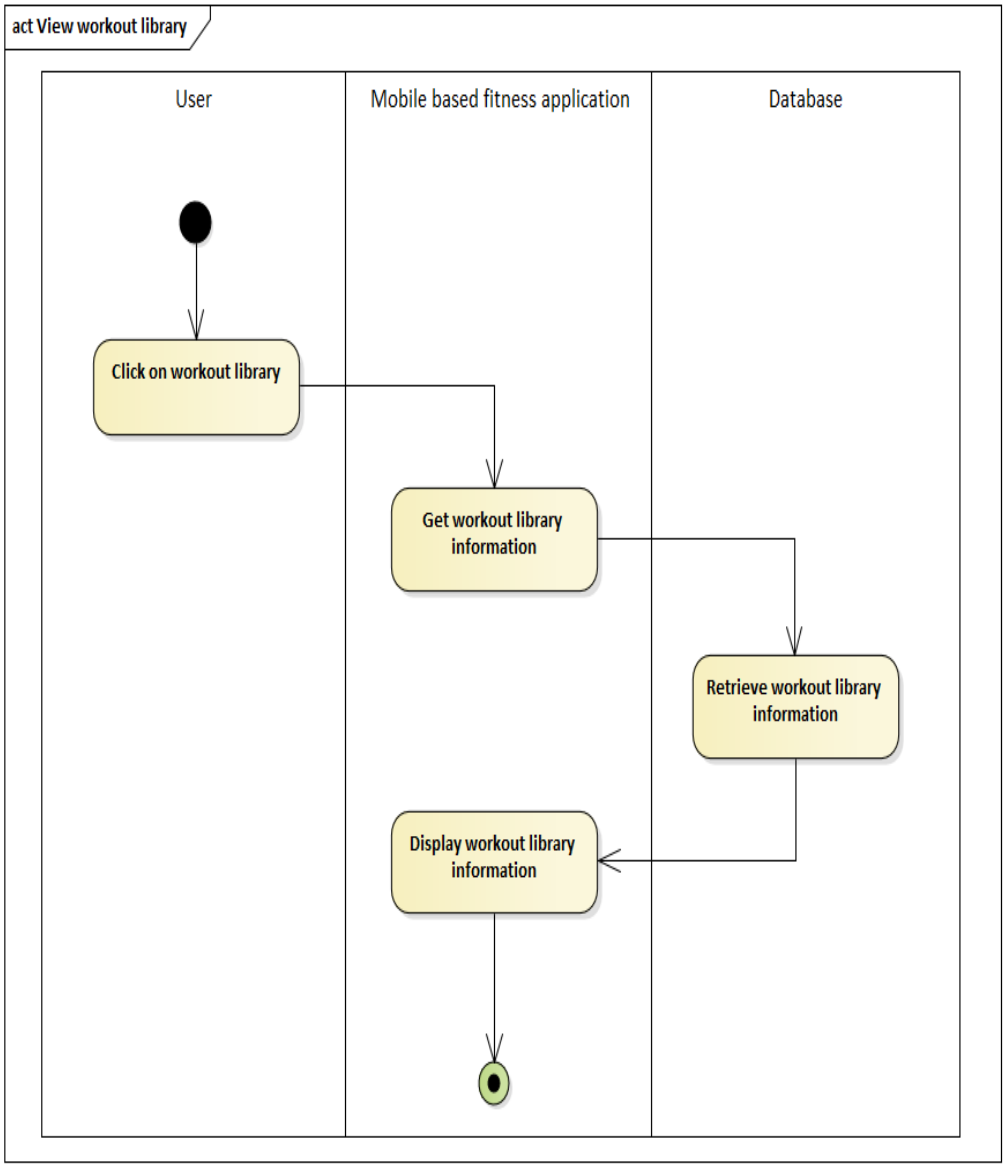


Figure 5.6: Activity Diagram – View Workout Library

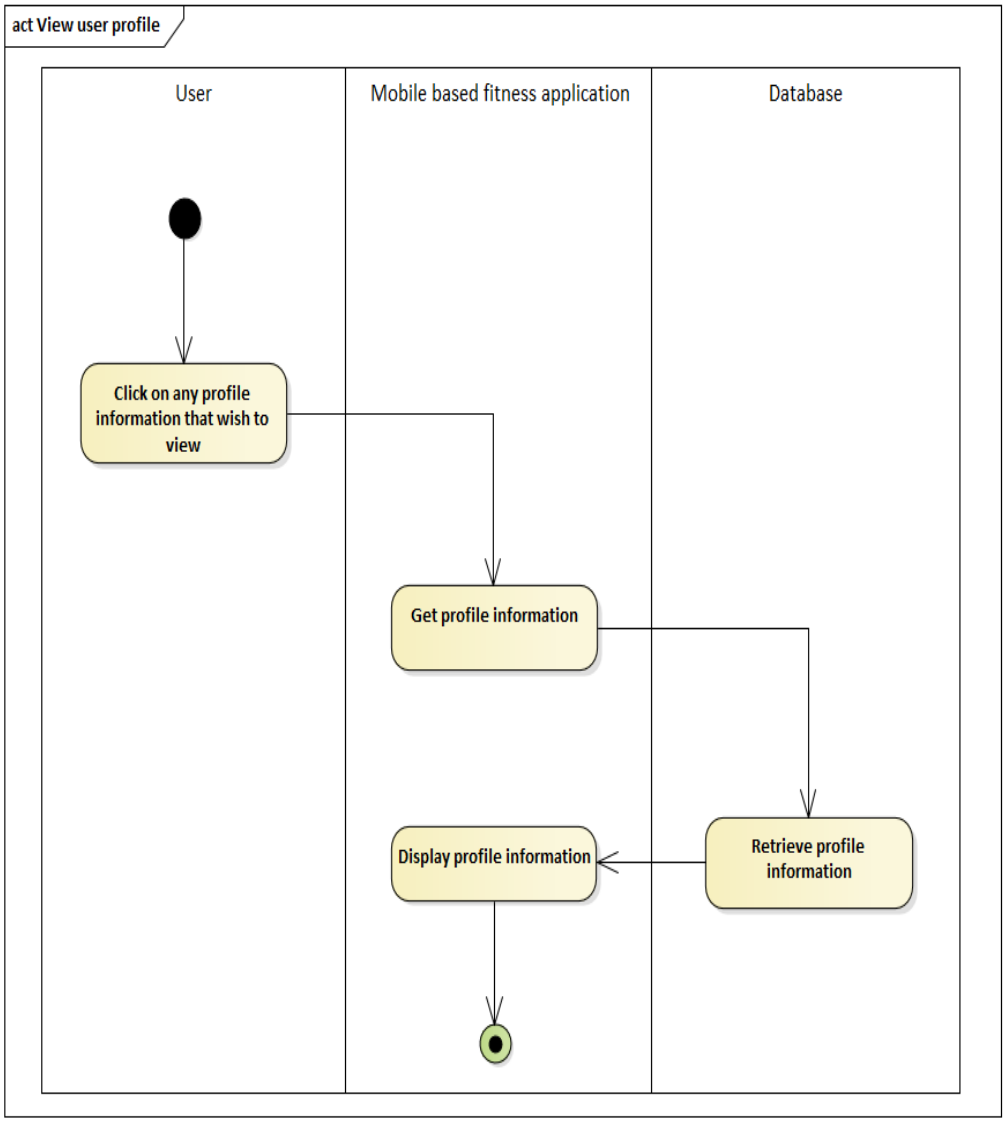


Figure 5.7: Activity Diagram – View User Profile

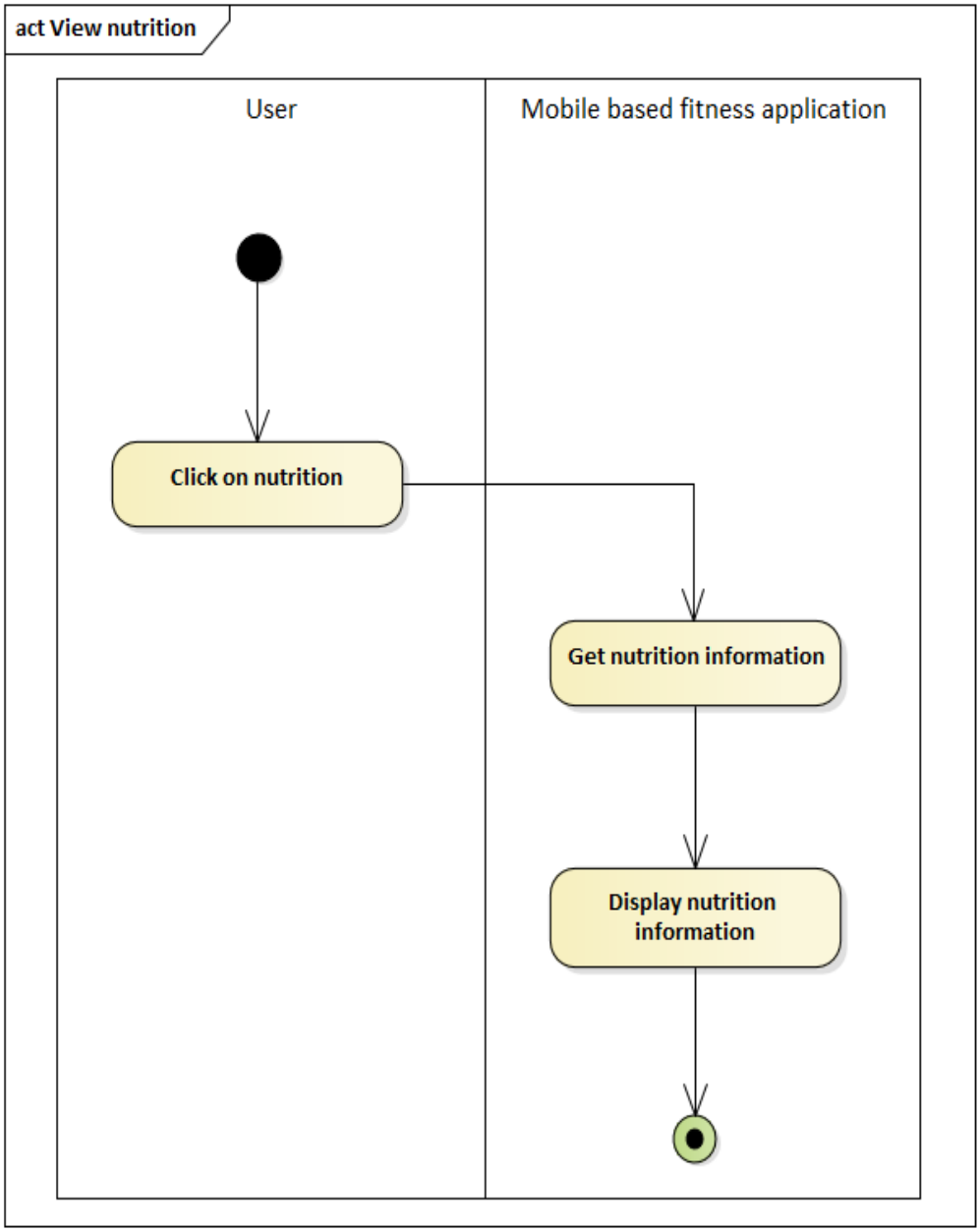


Figure 5.8: Activity Diagram – View Nutrition

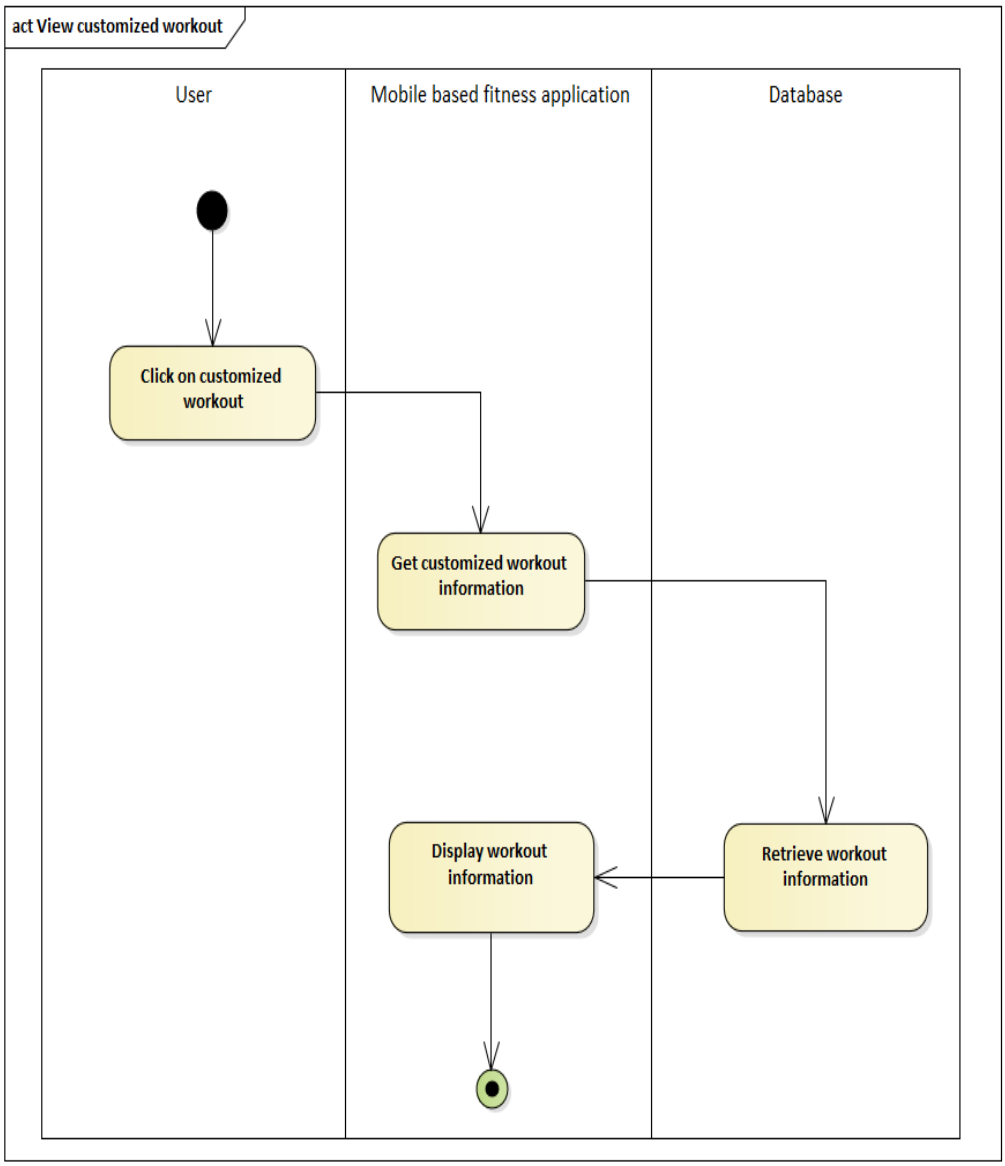


Figure 5.9: Activity Diagram – View Customized Workout

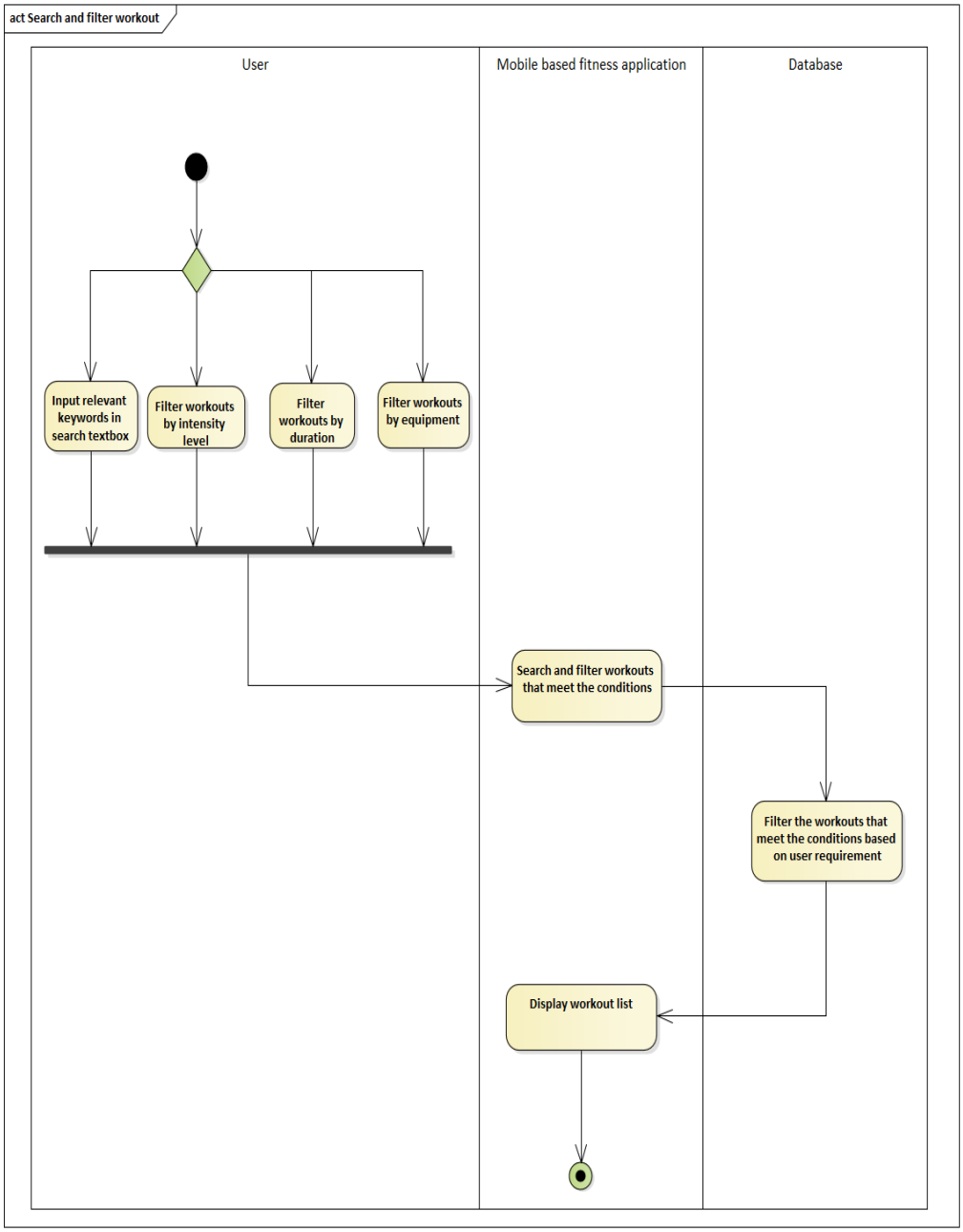


Figure 5.10: Activity Diagram – Search and Filter Workout

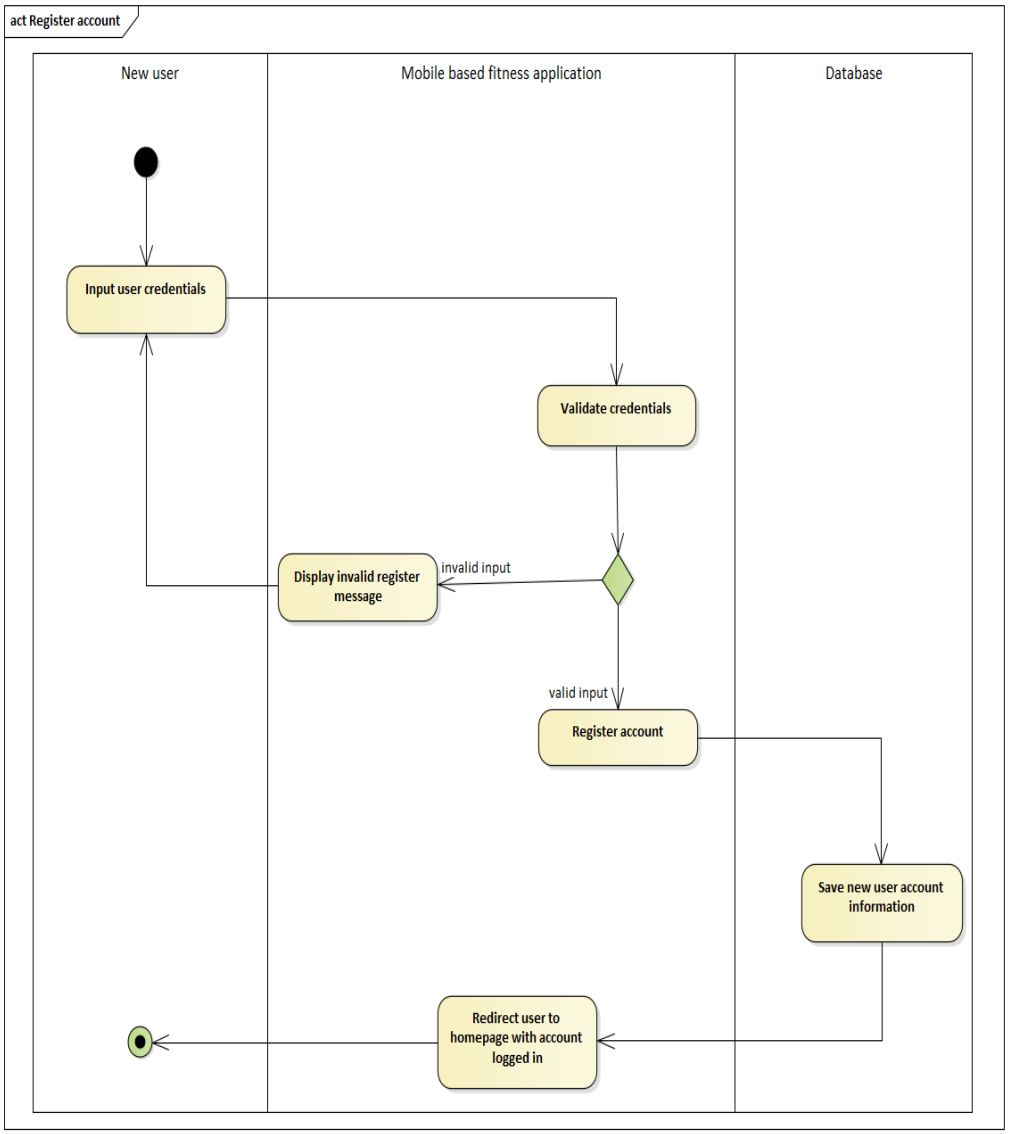


Figure 5.11: Activity Diagram – Register Account

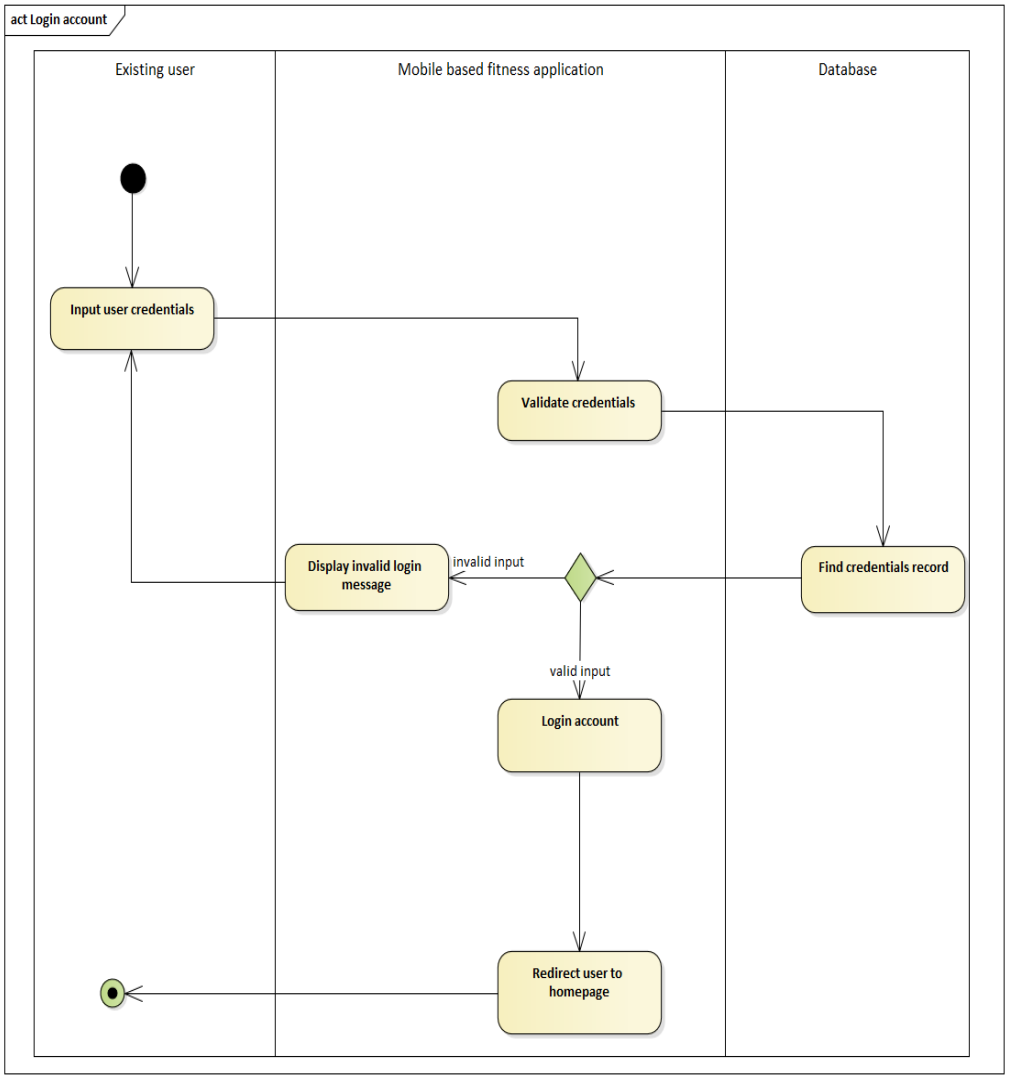


Figure 5.12: Activity Diagram – Login Account

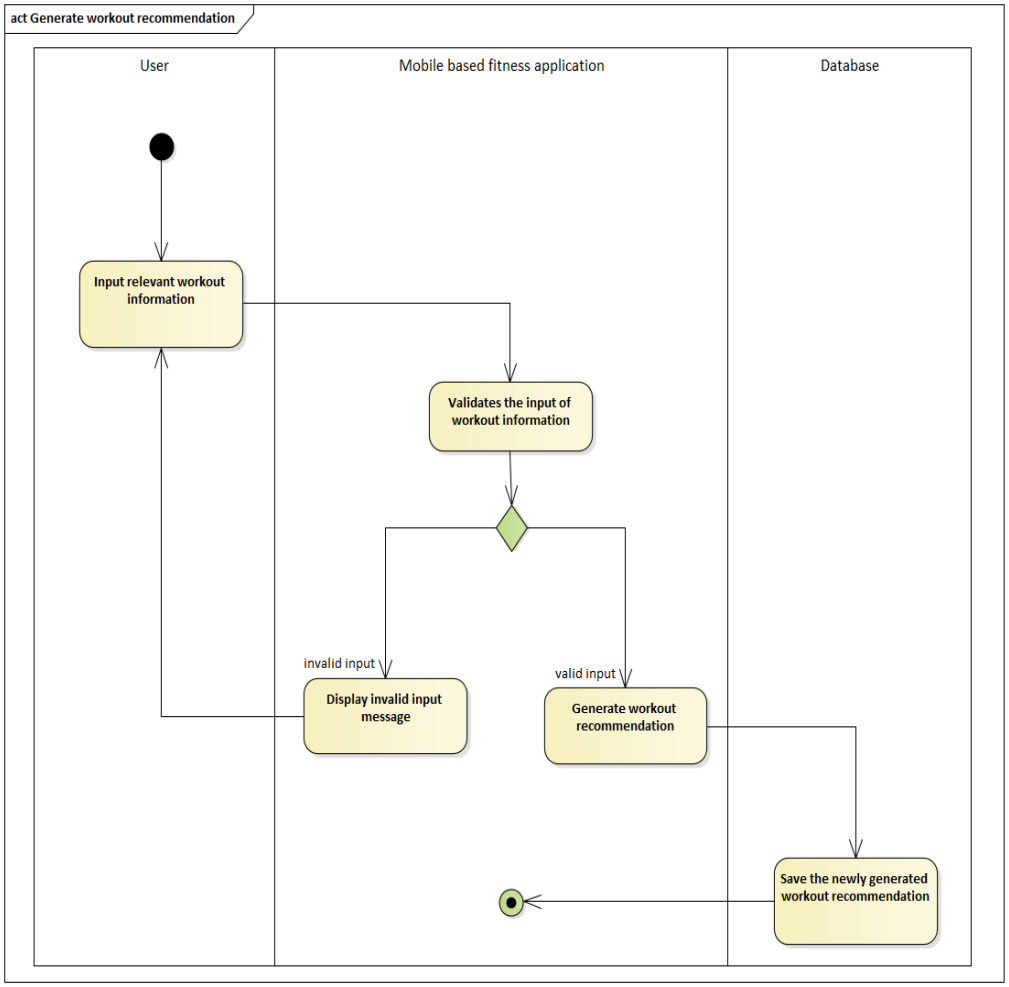


Figure 5.13: Activity Diagram – Generate Workout Recommendation

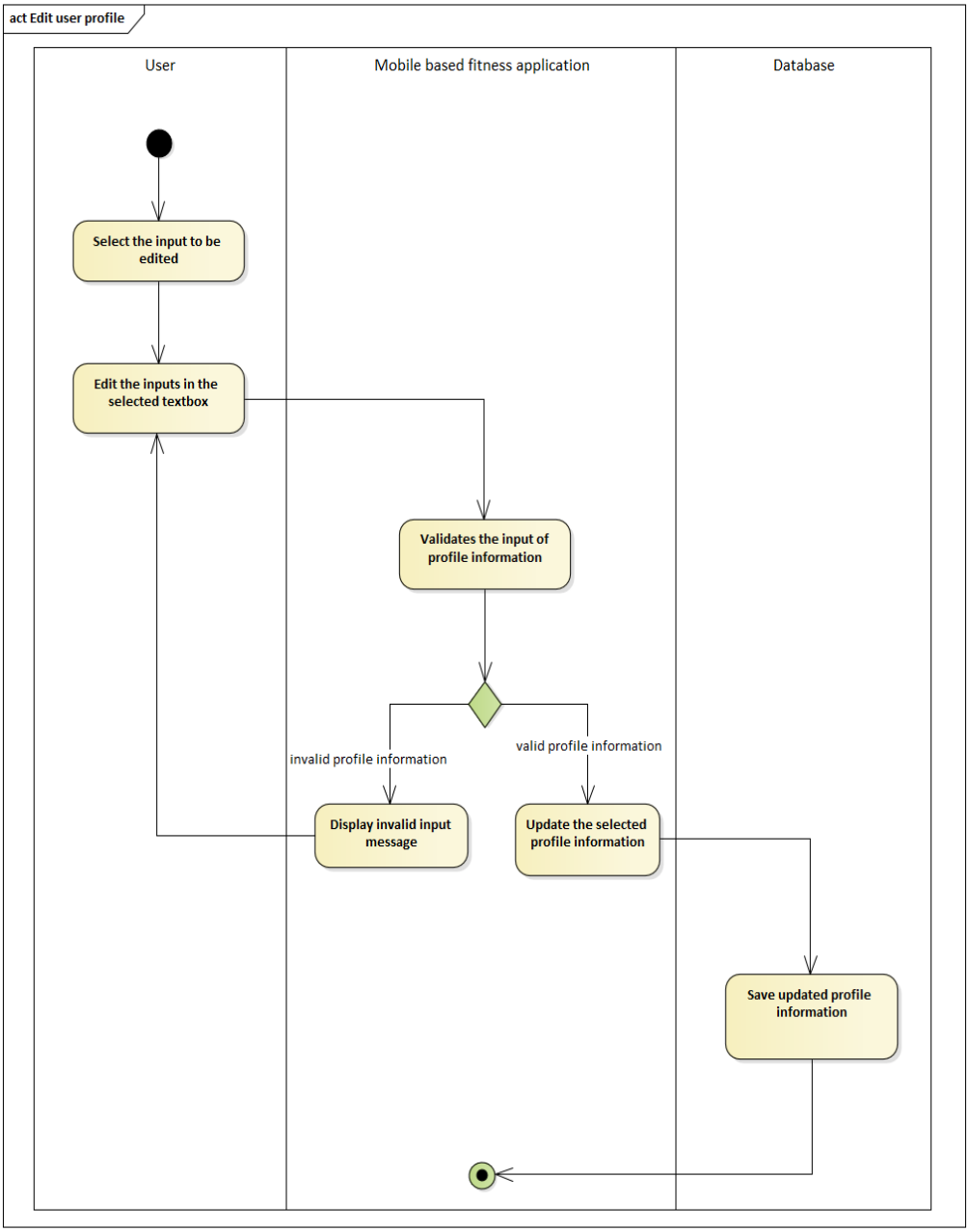


Figure 5.14: Activity Diagram – Edit User Profile

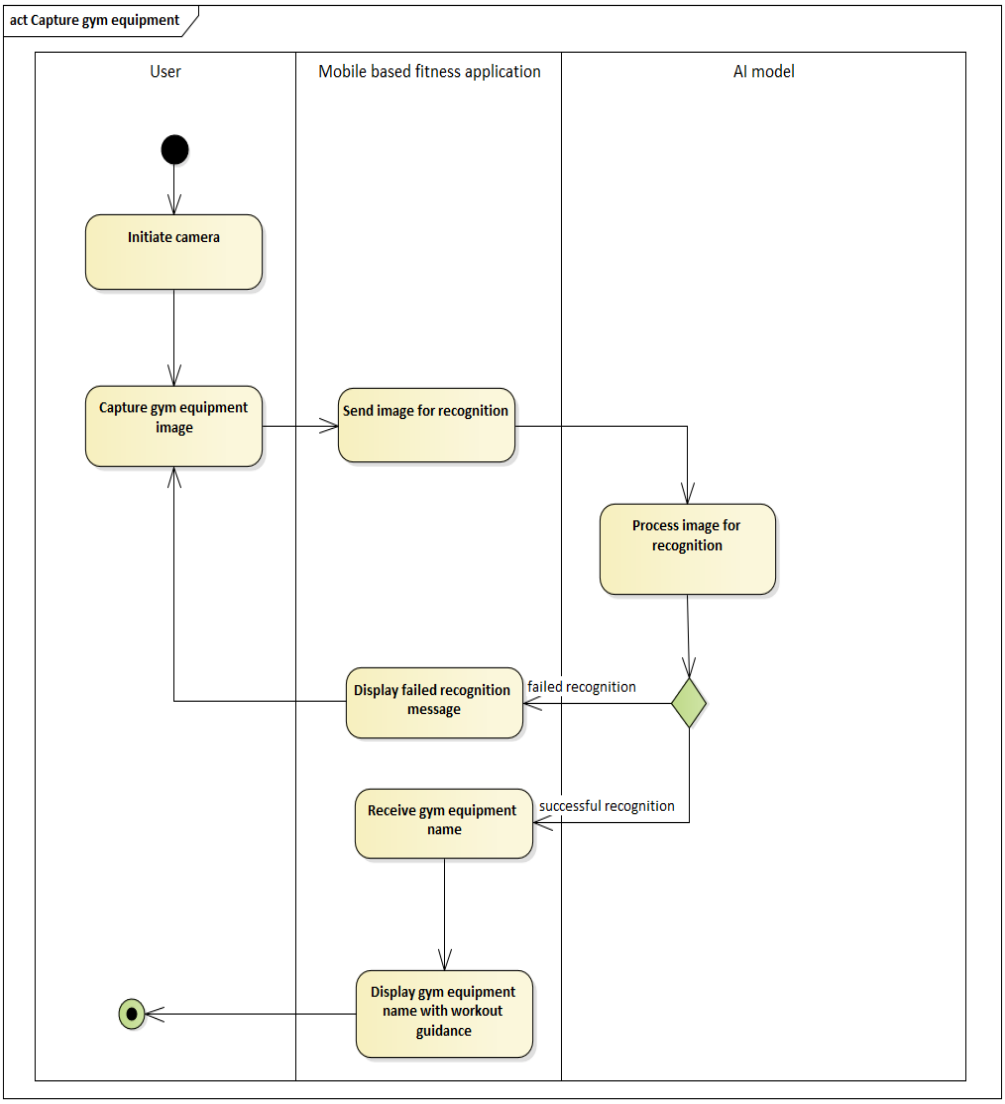


Figure 5.15: Activity Diagram – Capture Gym Equipment

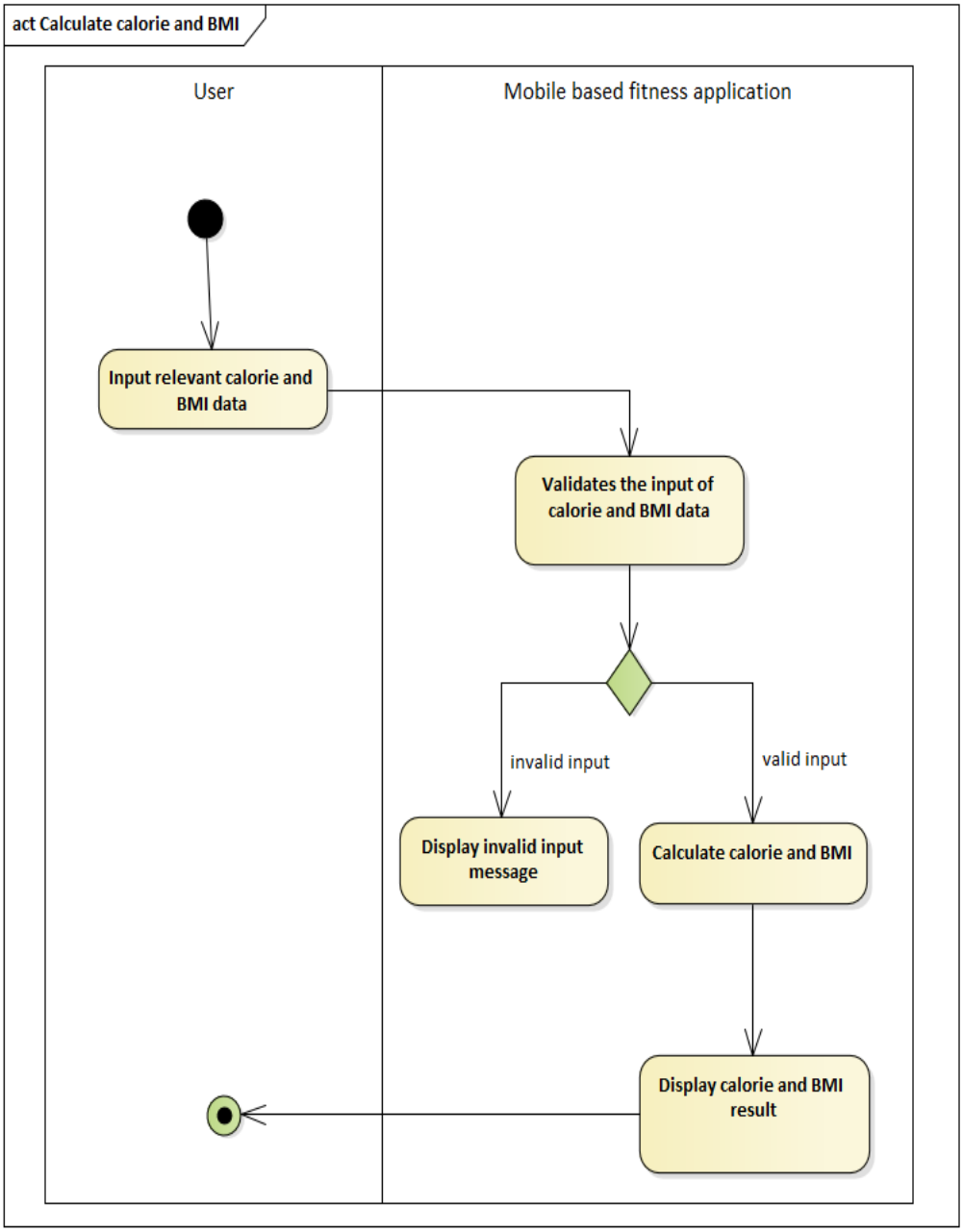


Figure 5.16: Activity Diagram – Calculate Calorie and BMI

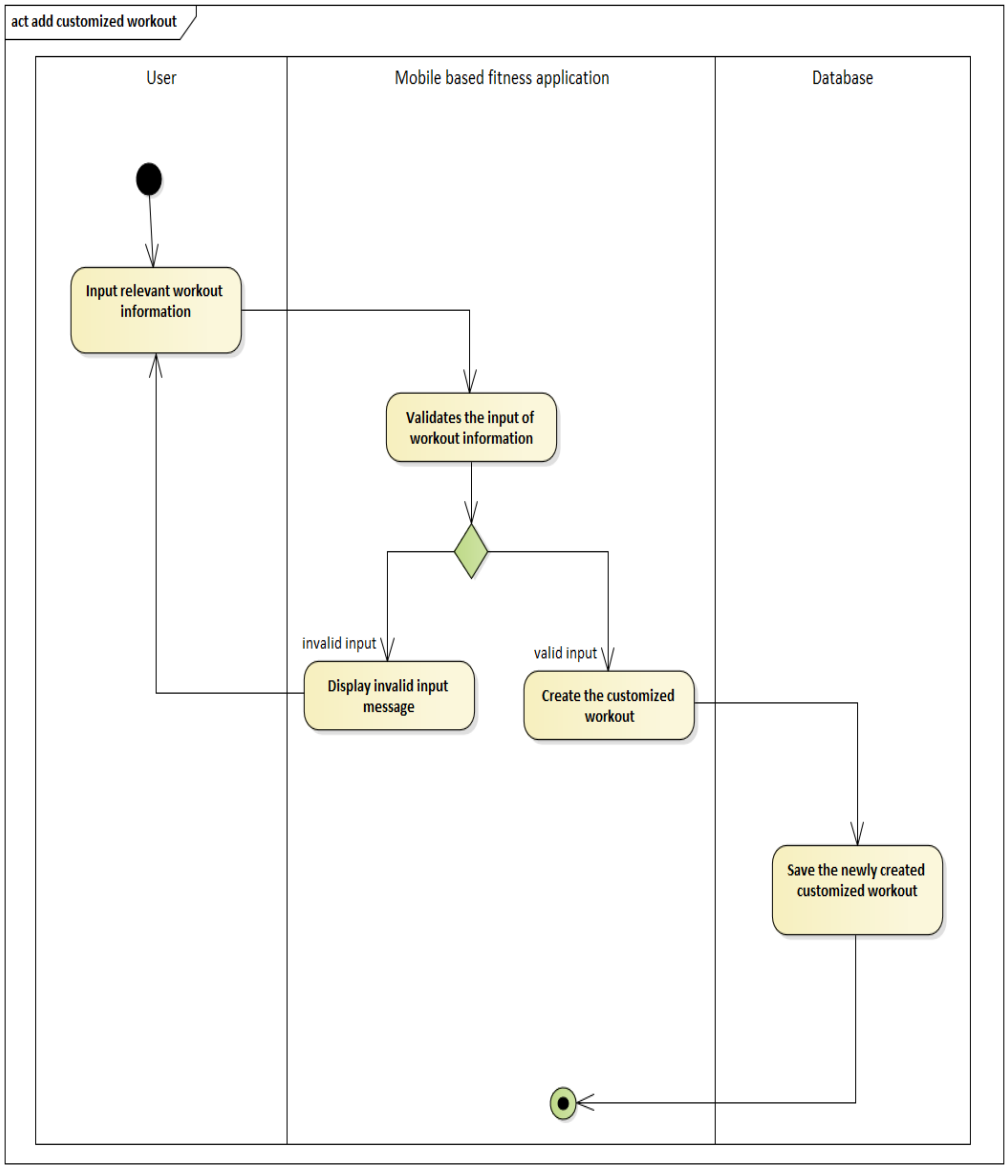


Figure 5.17: Activity Diagram – Add Customized Workout

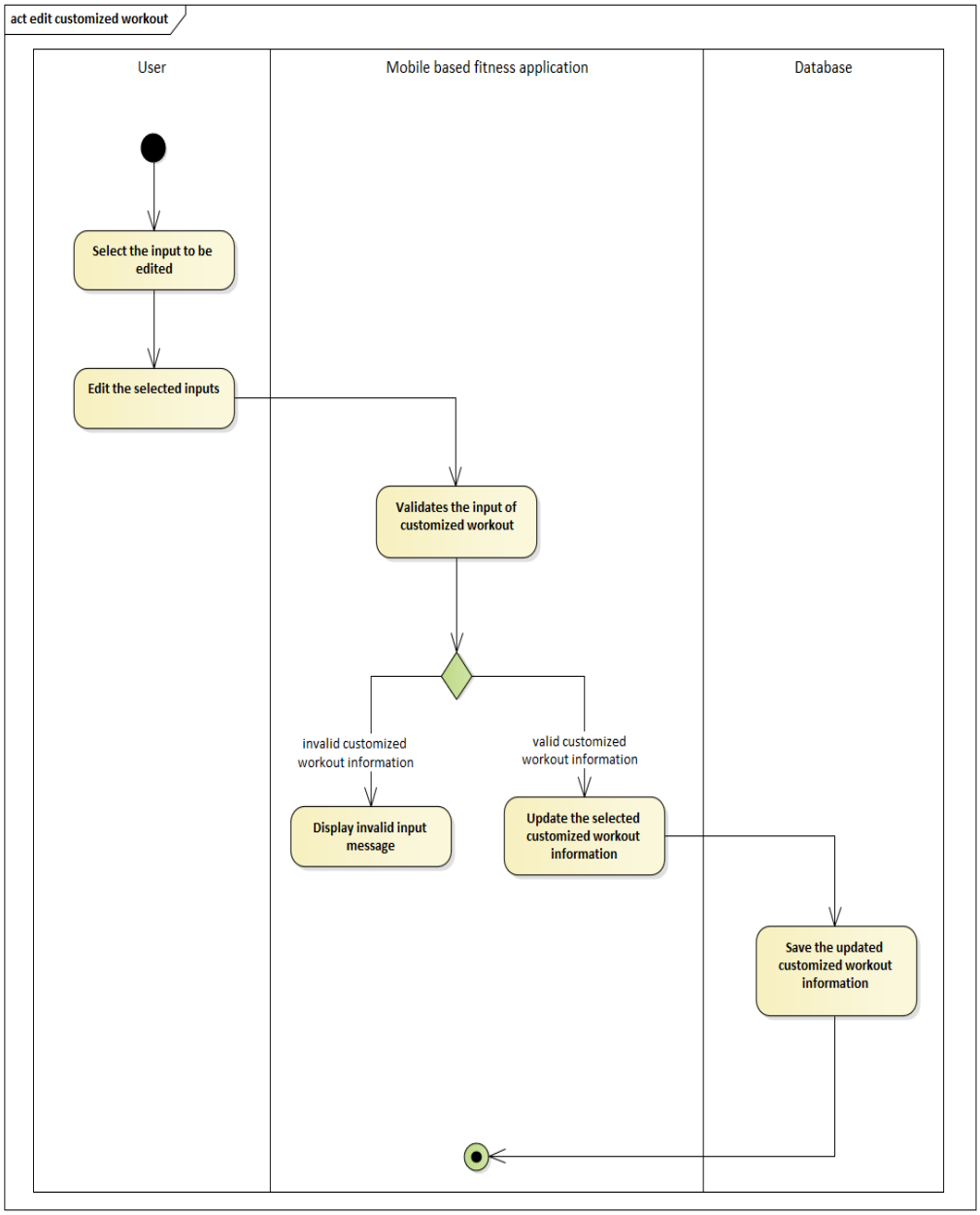


Figure 5.18: Activity Diagram – Edit Customized Workout

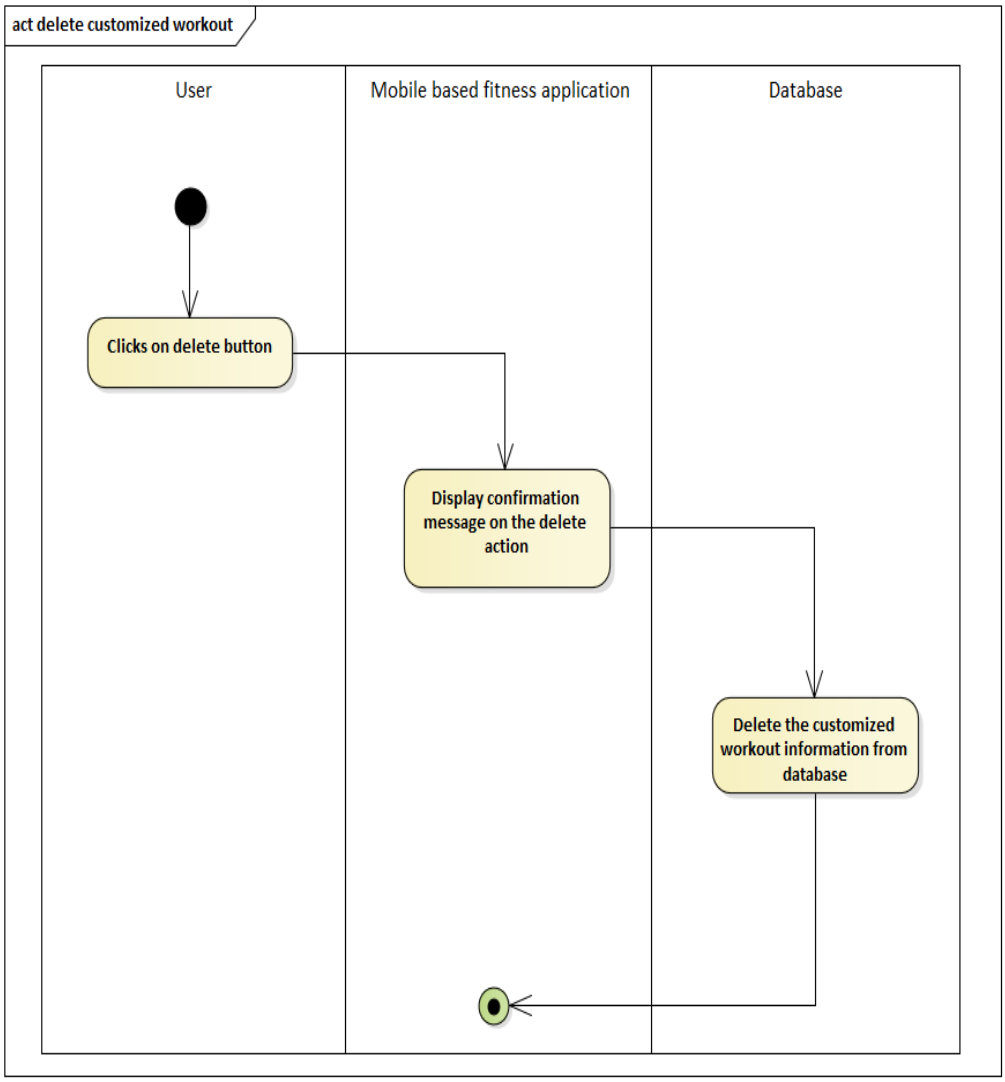


Figure 5.19: Activity Diagram – Delete Customized Workout

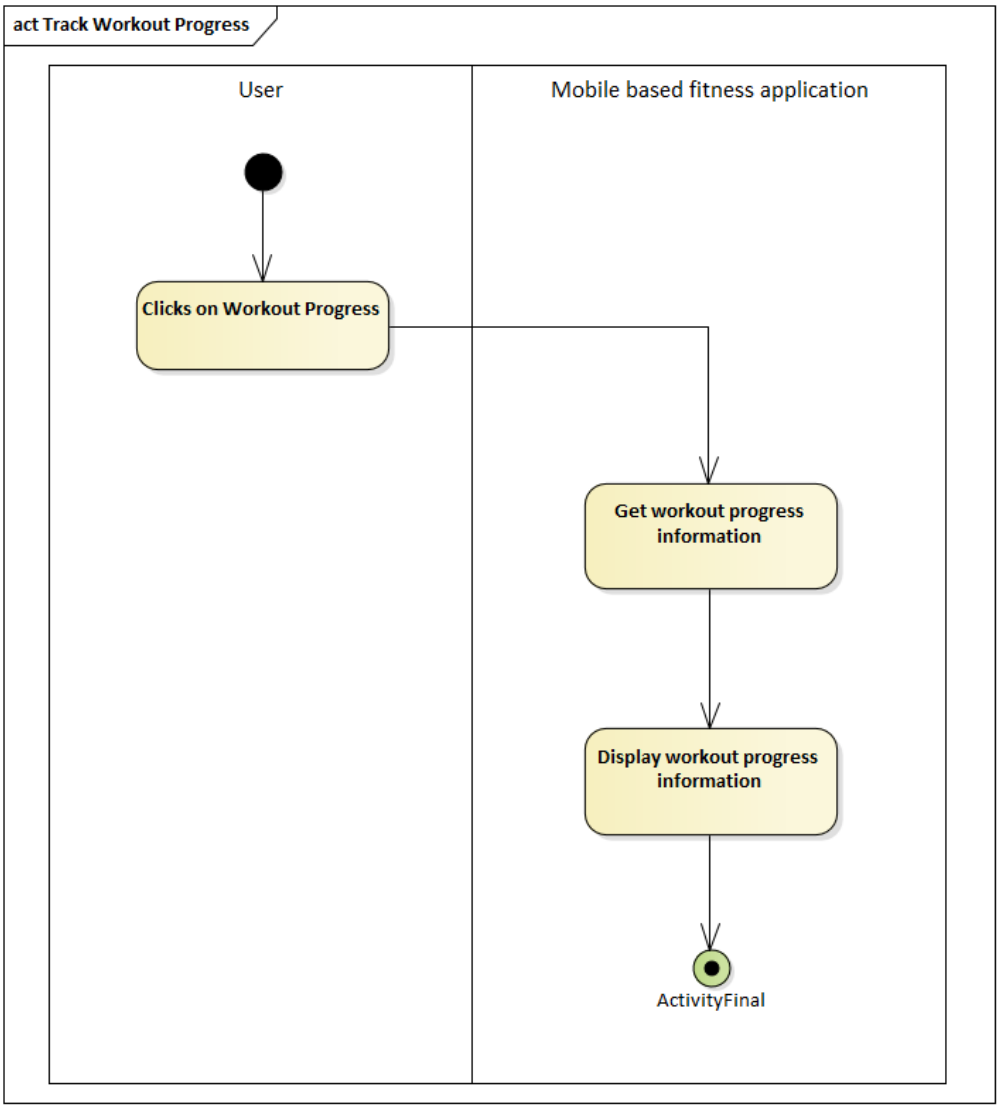


Figure 5.20: Activity Diagram – Track Workout Progress

5.5 User Interface Design

In this section, the user interfaces for each feature are displayed and explained as below. These user interfaces demonstrate an appealing design and screen flows, ensuring a seamless user experience while exploring the mobile application.

5.5.1 Main Screen

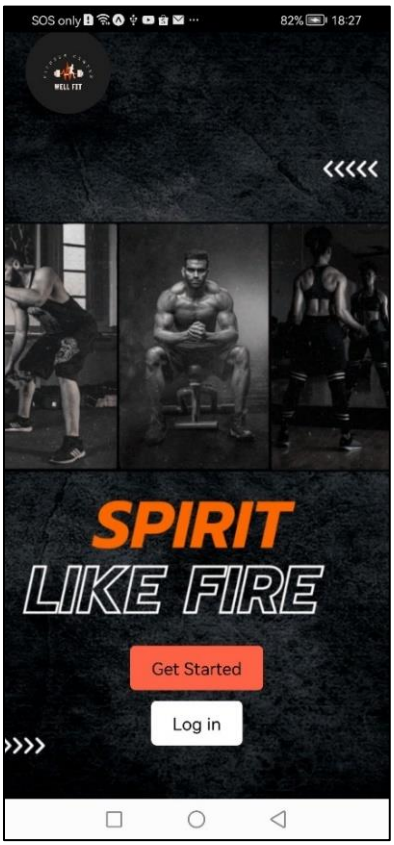


Figure 5.21: Main Screen

5.5.2 Register Screen

Figure 5.22: Register Screen

5.5.3 Login Screen

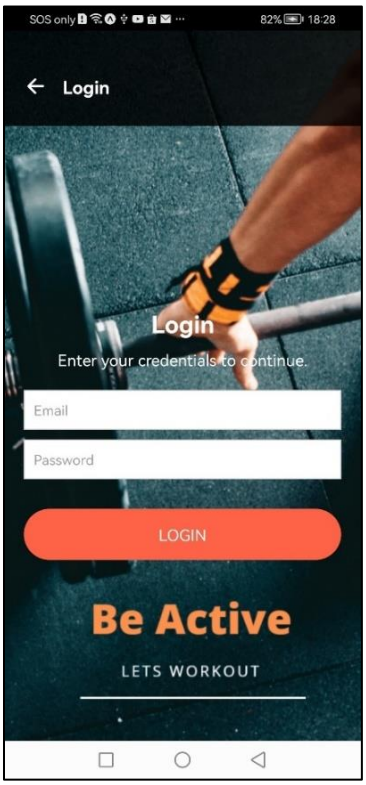
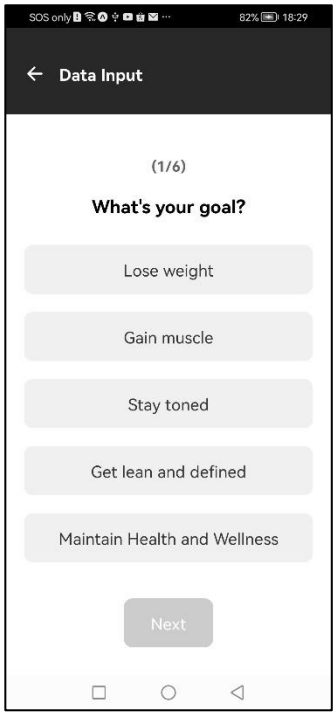
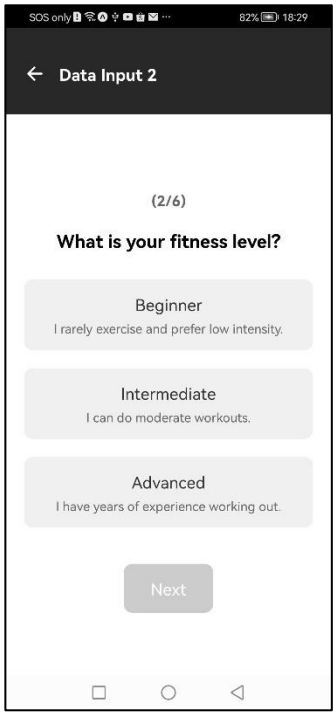


Figure 5.23: Login Screen

5.5.4 Data Input Screens



(a)

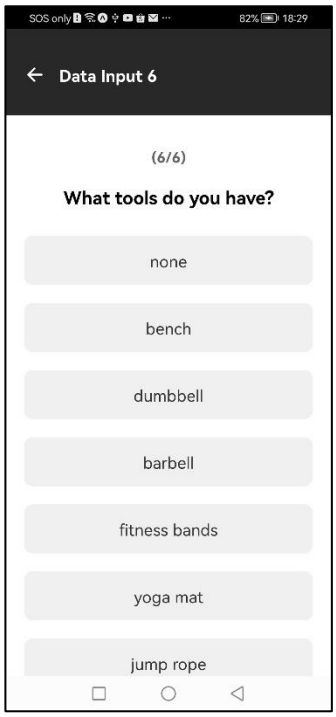


(b)

(c)

(d)

(e)



(f)

Figure 5.24(a)(b)(c)(d)(e)(f): Data Input Screen

5.5.5 Generate Plan Screen

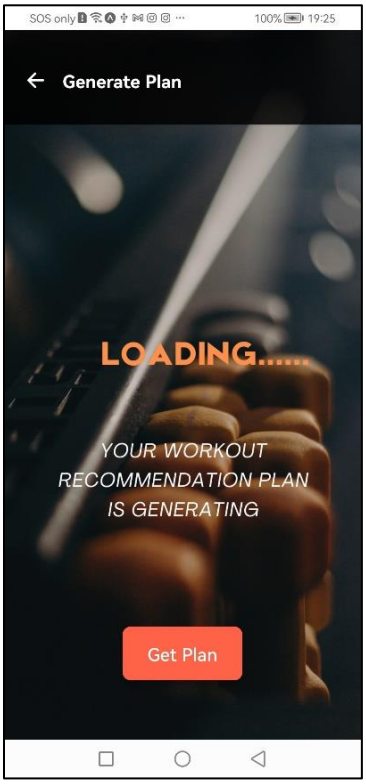


Figure 5.25: Generate Plan Screen

5.5.6 Home Screen

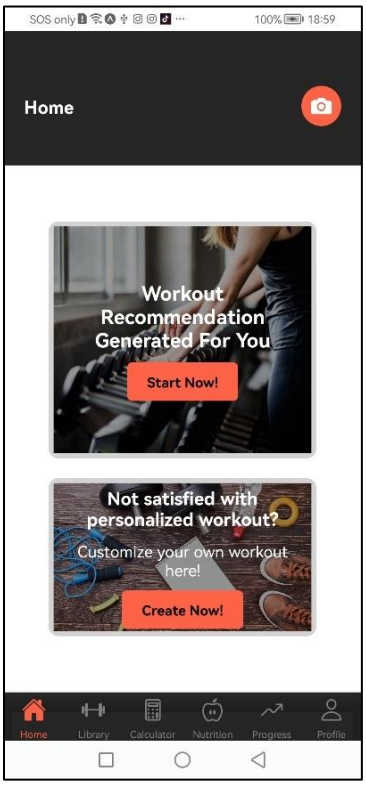


Figure 5.26: Home Screen

5.5.7 Workout Recommendation Screen

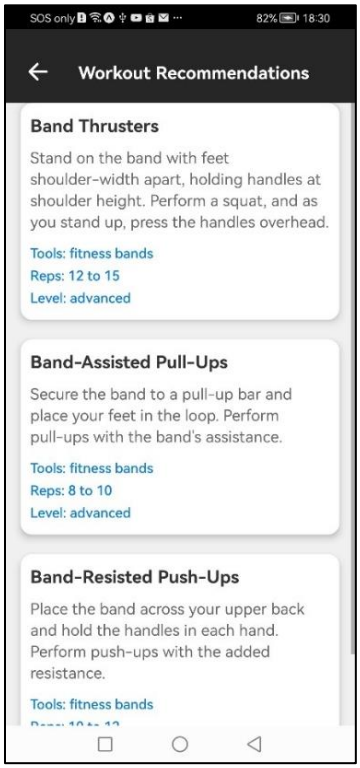
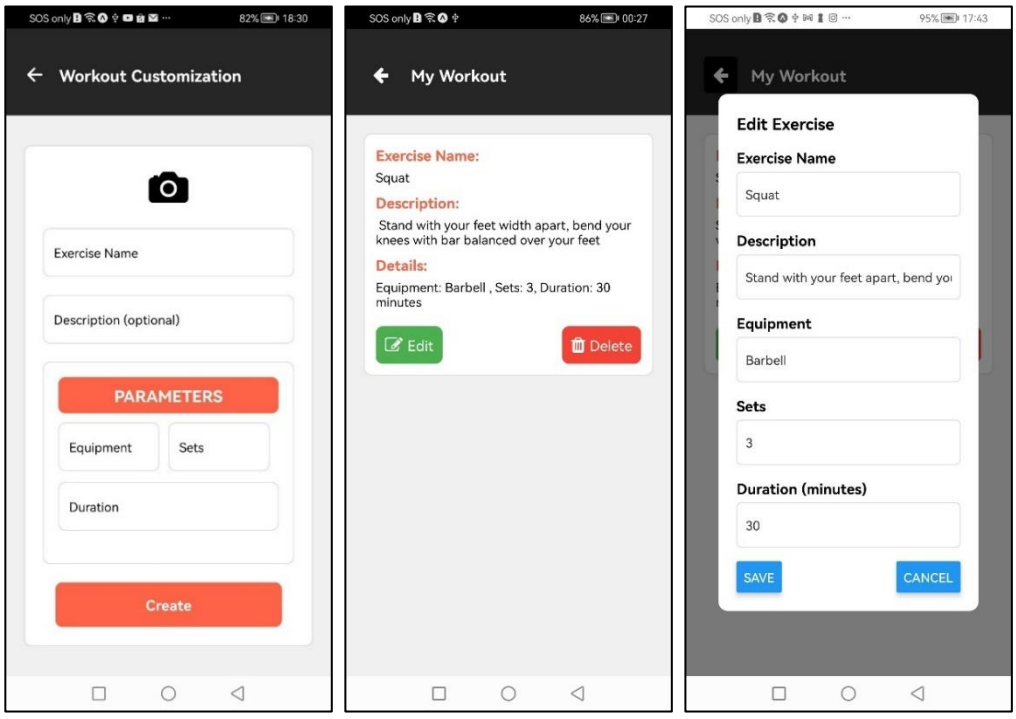


Figure 5.27: Workout Recommendation Screen

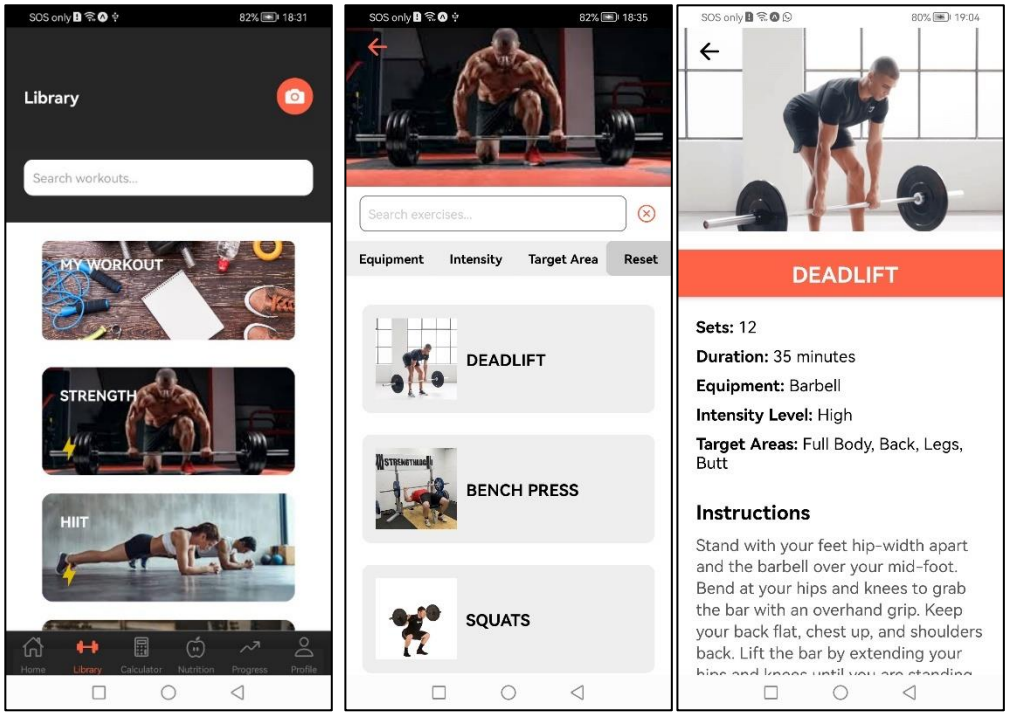
5.5.8 Workout Customization Screen



(a) (b) (c)

Figure 5.28(a)(b)(c): Workout Customization Screen

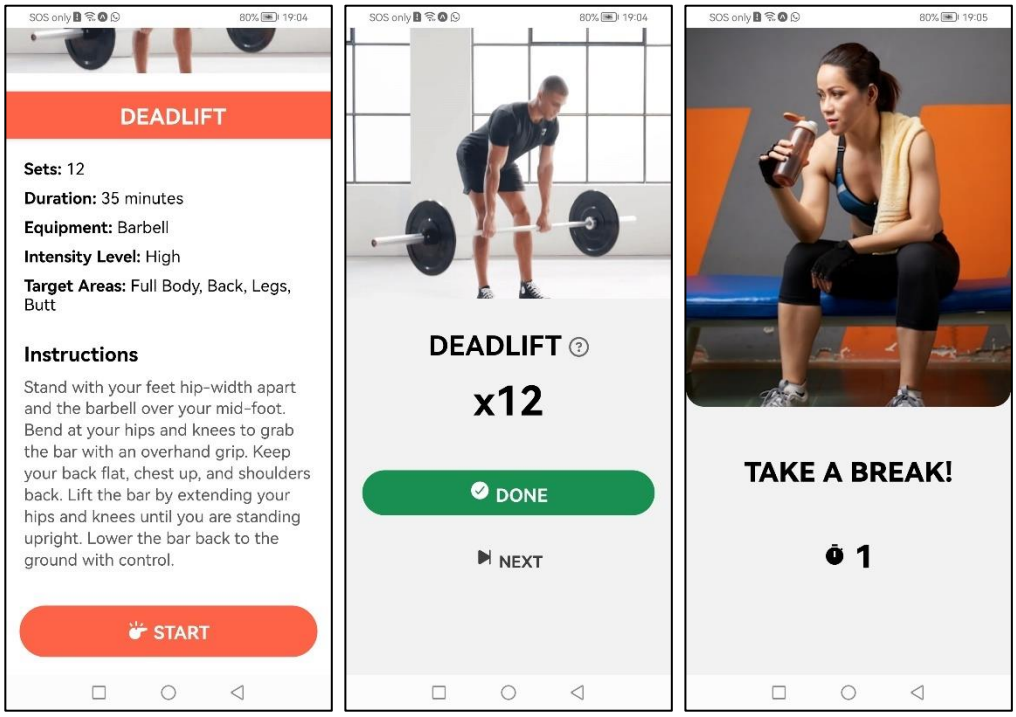
5.5.9 Workout Library Screen



(a) (b) (c)

Figure 5.29(a)(b)(c): Workout Library Screen

5.5.10 Workout Log Screen



(a) (b) (c)

Figure 5.30(a)(b)(c): Workout Log Screen

5.5.11 Calculator Screen

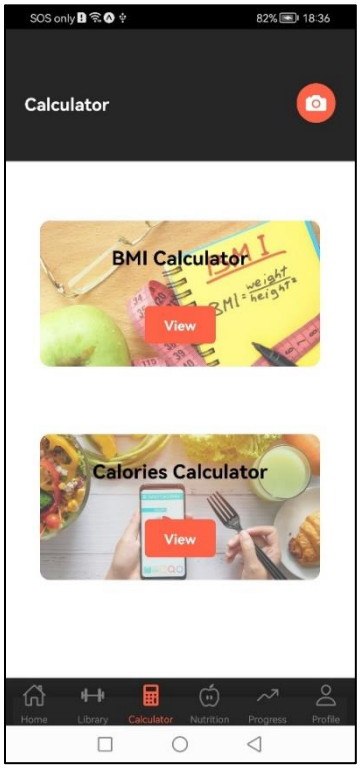
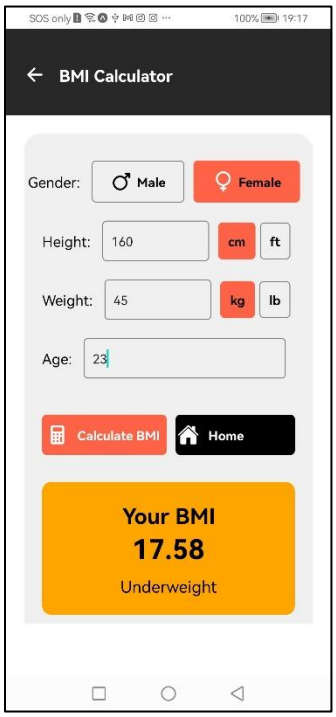


Figure 5.31: Calculator Screen

5.5.12 BMI Calculator Screen

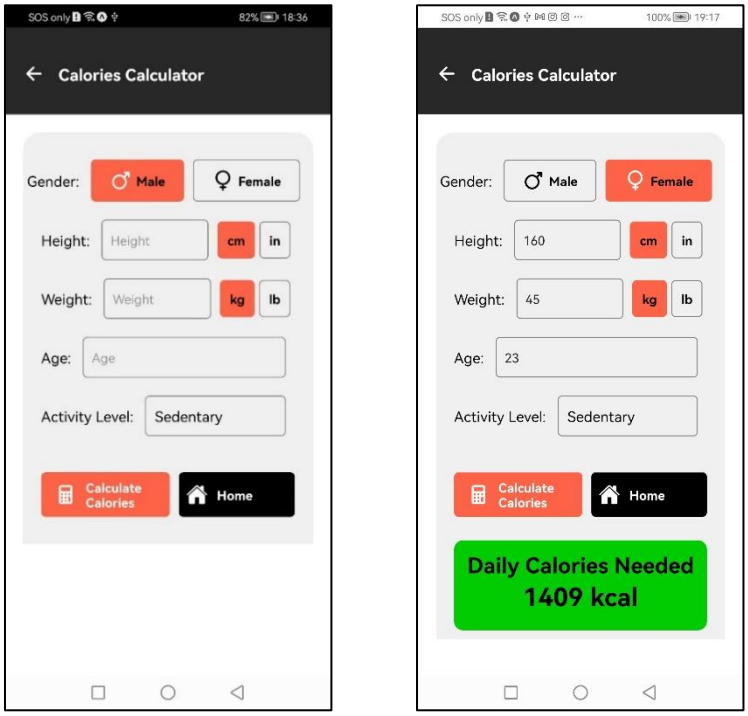
(a)



(b)

Figure 5.32(a)(b): BMI Calculator Screen

5.5.13 Calorie Calculator Screen



(a) (b)

Figure 5.33(a)(b): Calorie Calculator Screen

5.5.14 Nutrition Screen

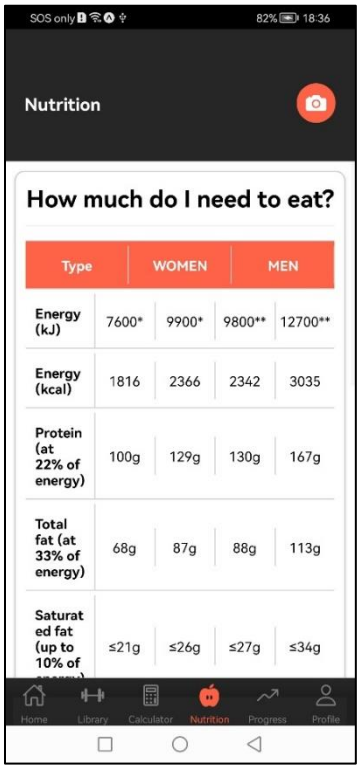


Figure 5.34: Nutrition Screen

5.5.15 Progress Screen

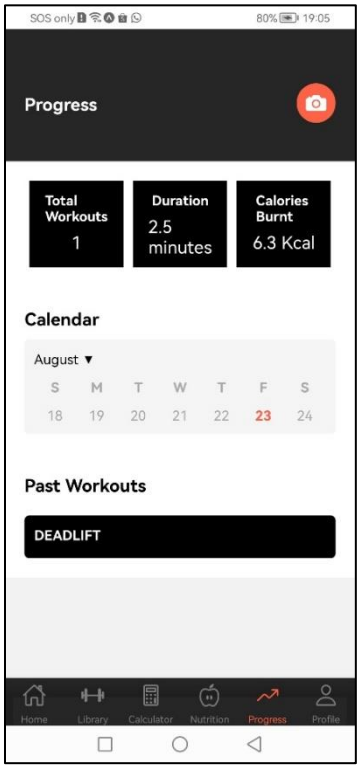
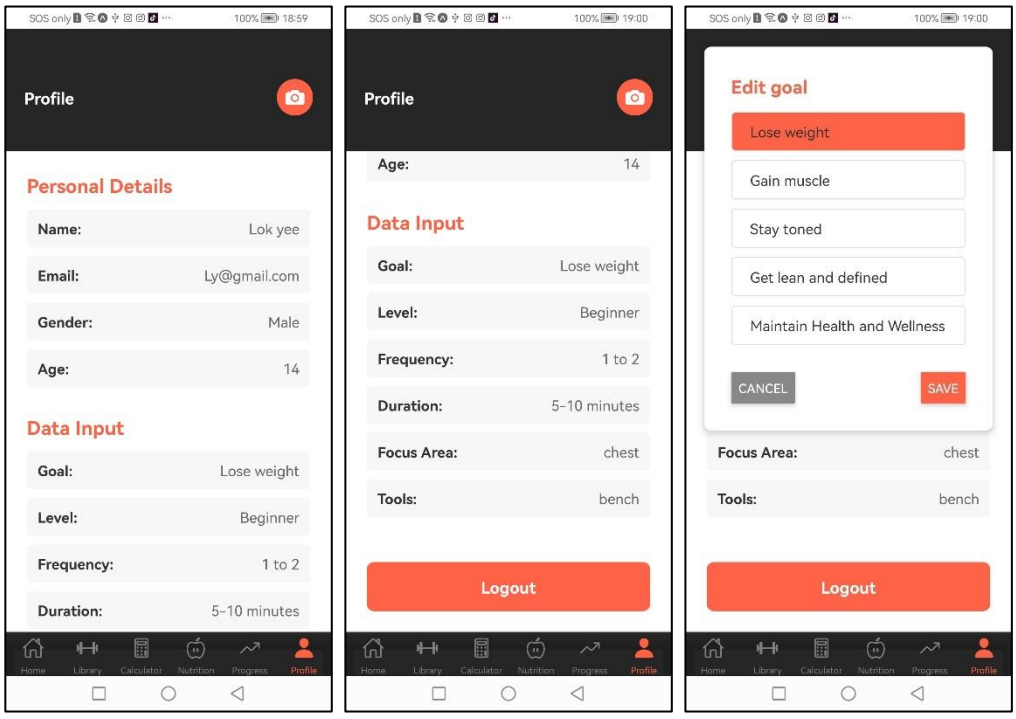


Figure 5.35: Progress Screen

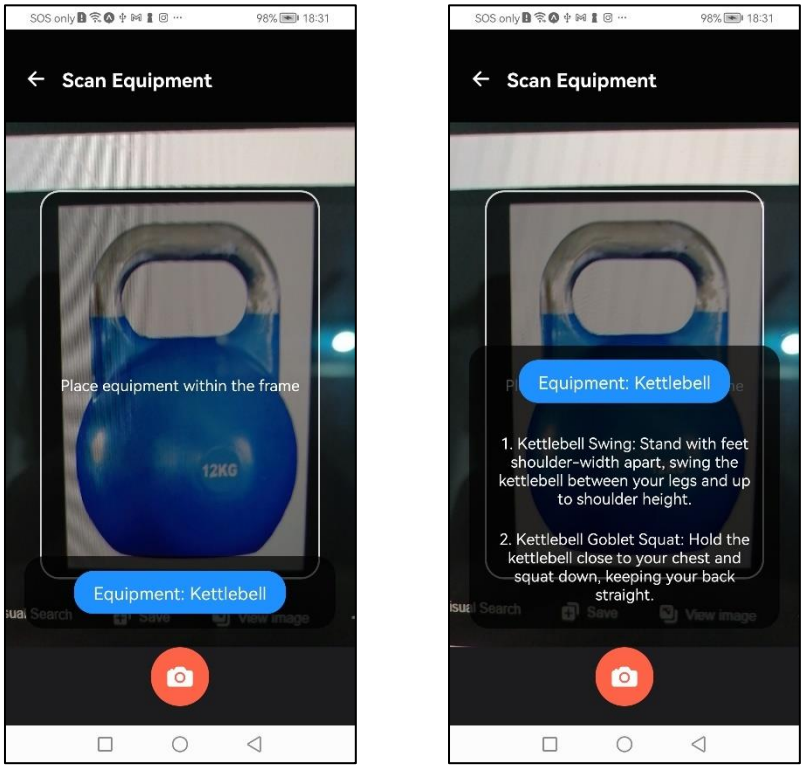
5.5.16 User Profile Screen



(a) (b) (c)

Figure 5.36(a)(b)(c): User Profile Screen

5.5.17 AI Gym Equipment Recognition Screen



(a) (b)

Figure 5.37(a)(b): AI Gym Equipment Recognition Screen

CHAPTER 6

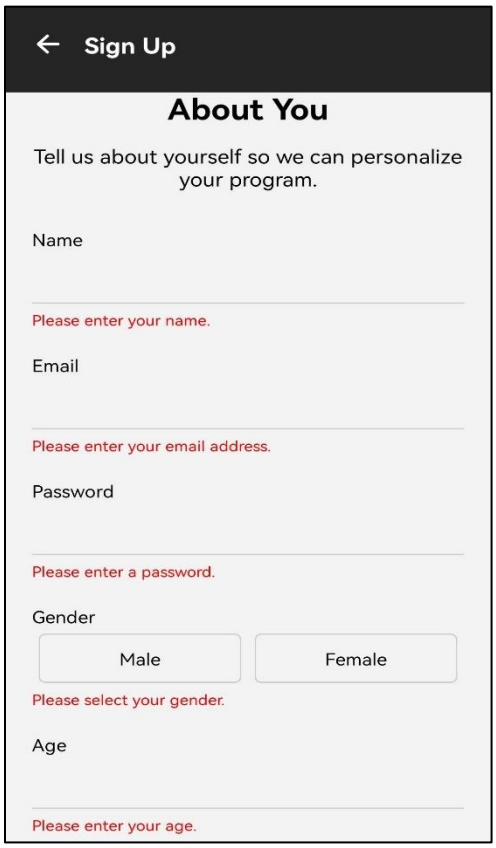
SYSTEM IMPLEMENTATION

6.1 Introduction

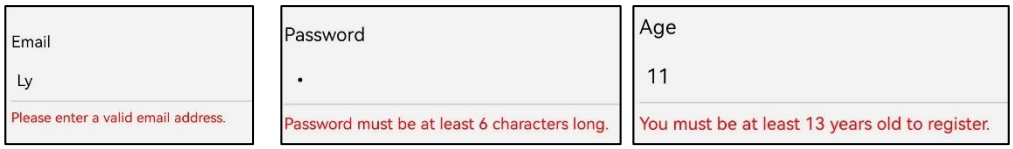
In this chapter, the modules, data validation and error handling for users are displayed as below. Additionally, the coding utilized in this application is explained in terms of its functions, endpoints and arguments. Lastly, the results of gym equipment recognition utilizing the VGG16, Resnet50 and InceptionV3 models are reviewed to be tabulated and contrasted.

6.2 SignUp module

For the first-time user, the system requires input details for the username, email address, password, age and gender selection. A few validations are added in the system, where all the input fields must not be left blank, users must enter a valid email address format, a 6-character long password and age greater and equals to 13. Otherwise, an error message will be shown directly beneath the input fields.



(a)



(b) (c) (d)

Figure 6.1(a)(b)(c)(d): Sign Up Form Validation Error Handling

After validation is passed, the sign-up details are sent to the backend (server.js) via the /register route.

```
axios.post('http://192.168.0.173:3000/register', { name, email, password, gender, age })
  .then(response => {
    if (response.data.success) {
      const userId = response.data.userId.toString();
      AsyncStorage.setItem('userId', userId);

      // Navigate to DataInput screen with userId
      navigation.navigate('DataInput', { userId });
    } else {
      Alert.alert('Registration Failed', response.data.message); // Show error message if registration fails
    }
  })
  .catch(error => {
    console.error('Error registering user:', error);
    Alert.alert('Registration Error', 'Failed to register user. Check if email already exists.'); // Alert if there's an error
  });
};
```

Figure 6.2: Code Snippet of Sign Up Function (frontend)

From the backend side, it determines whether the email is already stored in the database. It also hashes the password and keeps user data in the database if the email is new.

```
app.post('/register', async (req, res) => {
  const { name, email, password, gender, age } = req.body;

  try {
    const emailCheckQuery = 'SELECT * FROM users WHERE email = ?';
    db.query(emailCheckQuery, [email], async (err, results) => {
      if (err) {
        console.error('Error executing email check query:', err);
        return res.status(500).json({ success: false, message: 'Error checking email' });
      }

      if (results.length > 0) {
        // Email already exists
        console.log('Email already in use:', email);
        return res.status(400).json({ success: false, message: 'Email already in use' });
      }

      try {
        const hashedPassword = await bcrypt.hash(password, 10);
        const query = 'INSERT INTO users (name, email, password, gender, age) VALUES (?, ?, ?, ?, ?)';

        db.query(query, [name, email, hashedPassword, gender, age], (err, results) => {
          if (err) {
            console.error('Error executing user registration query:', err);
            return res.status(500).json({ success: false, message: 'Error registering user' });
          }

          const userId = results.insertId; // Retrieve the userId from the results

          console.log('User registered successfully with userId:', userId);
          res.status(200).json({ success: true, message: 'Registration successful', userId });
        });
      } catch (hashError) {
        console.error('Error hashing password:', hashError);
        res.status(500).json({ success: false, message: 'Error hashing password' });
      }
    });
```

Figure 6.3: Code Snippet of Sign Up Function (backend)

6.3 Login module

To login into the application, existing users must enter their email and password. Once the user enters their email and password, the system verifies the existence of the inputs by comparing the entered details with the stored details in the database. If the password and email are mismatched, the system shows an error message of the invalid email or password. Moreover, the login form checks whether any input fields are left empty. Once the input is validated, a POST request is sent to the backend. If the login is successful, the user is navigated to the Home screen with the account logged in. Otherwise, the users are blocked from signing in with the displayed error message.

(a) (b) (c)

Figure 6.4: Login Form Validation Error Handling

```
const handleLogin = async () => {
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

  setEmailError('');
  setPasswordError('');

  if (!emailRegex.test(email)) {
    setEmailError('Please enter a valid email address.');
    return;
  }

  if (password.length < 6) {
    setPasswordError('Password must be at least 6 characters long.');
    return;
  }

  try {
    const response = await axios.post('http://192.168.0.173:3000/login', {
      email,
      password,
    });

    if (response.data.success) {
      const userId = response.data.user.id.toString();
      await AsyncStorage.setItem('userId', userId);
      navigation.navigate('Home', { user: response.data.user });
    } else {
      setEmailError(response.data.message);
    }
  } catch (error) {
    console.error('Error logging in:', error);
    setEmailError('Account does not exist. Please register your account.');
  }
};
```

Figure 6.5: Code Snippet of Login Function (frontend)

```
app.post('/login', async (req, res) => {
  const { email, password } = req.body;
  const query = 'SELECT * FROM users WHERE email = ?';

  db.query(query, [email], async (err, results) => {
    if (err) {
      console.error('Error executing query:', err);
      return res.status(500).json({ success: false, message: 'Error logging in' });
    }

    if (results.length > 0) {
      const user = results[0];
      try {
        if (await bcrypt.compare(password, user.password)) {
          console.log('User logged in successfully:', user);
          res.status(200).json({ success: true, message: 'Login successful', user });
        } else {
          res.status(401).json({ success: false, message: 'Login failed. Invalid credentials' });
        }
      } catch (error) {
        console.error('Error comparing passwords:', error);
        res.status(500).json({ success: false, message: 'Error comparing passwords' });
      }
    } else {
      res.status(401).json({ success: false, message: 'Login failed. Invalid credentials' });
    }
  });
});
```

Figure 6.6: Code Snippet of Login Function (backend)

6.4 Workout Library with Search and Filter features

This project library offers 5 distinct workout categories: Strength, HIIT, Pilates, Yoga and Cardio. To locate exercises, users may apply a wide search filter across these categories or narrow down the search by utilizing the search function inside each of the categories. Within each category, it is supported by the filter features, where the user can filter the workouts with equipment, intensity and target area. These features work seamlessly together, capable of finding the best workouts according to user personal preferences. The figure below shows some instances of adding filters to the workout library and performing search functions from the workout library.

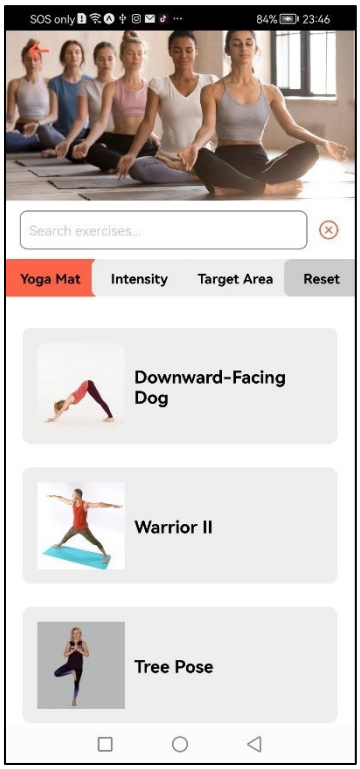


Figure 6.7: Workouts with Equipment Filter Applied

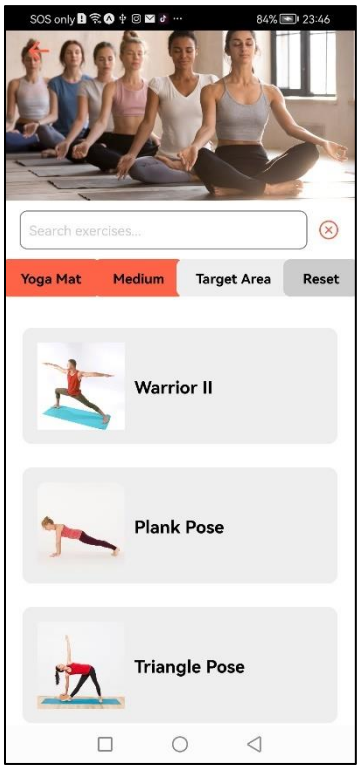


Figure 6.8: Workouts with Intensity Filter Add On

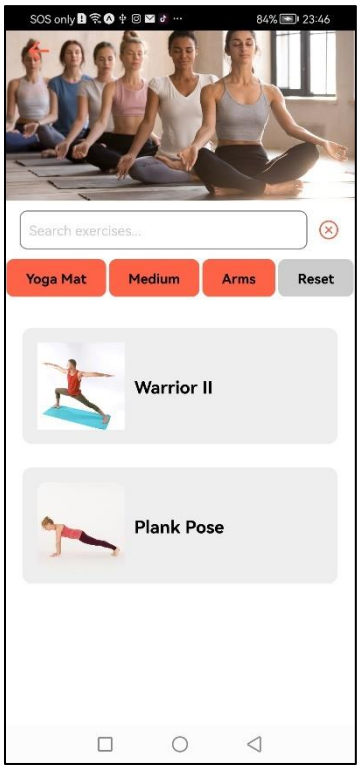


Figure 6.9: Workouts with Target Area Filter Add On

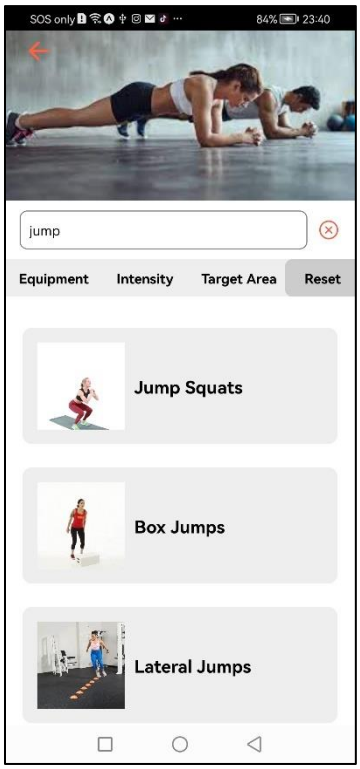


Figure 6.10: Workouts with Search function using keywords of “jump”

Furthermore, users will be able to remove any filters that have been imposed. Users can click the "Reset" button to quickly return the features to their initial condition after applying the filter.

```
    // Filter by equipment
    if (filters.equipment && filters.equipment !== 'All') {
      matchEquipment = item.equipment.includes(filters.equipment);
    }

    // Filter by intensity
    if (filters.intensity && filters.intensity !== 'All') {
      matchIntensity = item.intensity === filters.intensity;
    }

    // Filter by target area
    if (filters.targetArea && filters.targetArea !== 'All') {
      matchTargetArea = item.targetAreas.includes(filters.targetArea);
    }

    // Return true only if all filter conditions are matched
    return matchEquipment && matchIntensity && matchTargetArea;
  });
};

// Function to handle clicking on an exercise
const handleExercisePress = (item) => {
  navigation.navigate('ExerciseDetail', { exercise: item });
};

// Function to reset all filters
const resetAllFilters = () => {
  setFilters({
    equipment: null,
    intensity: null,
    targetArea: null,
  });
  setSelectedEquipment(null); // Reset selected equipment
  setSearchText(''); // Clear search text
```

Figure 6.11: Code snippet of the Filter Function

6.5 Workout Detail

Users will exposed to detailed information of each workout, including the motion guidance, sets, duration, equipment, intensity level, target areas and written instructions. According to the comprehensive workout details, users can opt to perform the workout, offering valuable insights on the calories burnt and duration spent on the workout and details of the completed workout.

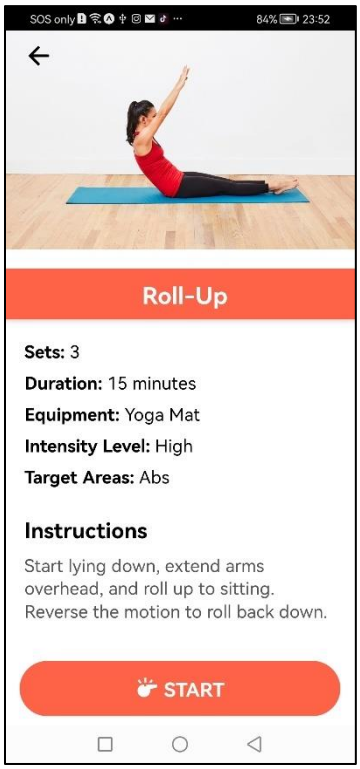


Figure 6.12: Workout with detailed information

6.6 Workout Customization

Additionally, this application involves workout customization features, allowing users to customize their workouts according to their preferences. The customization input includes exercise name, description, equipment, number of sets and duration. When entering all the necessary details, the application verifies all the inputs must not left blank and only numeric values are accepted for the parameters of sets and duration.

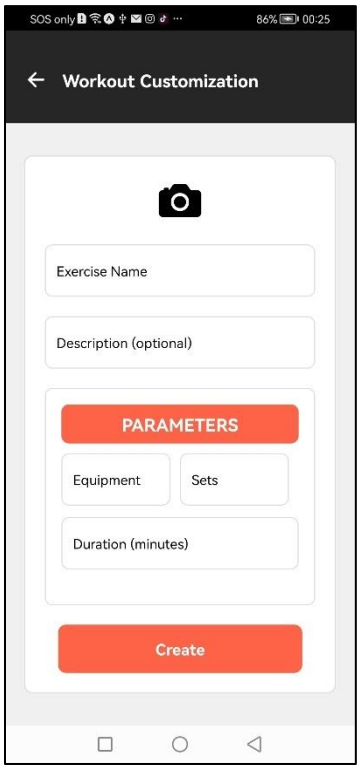
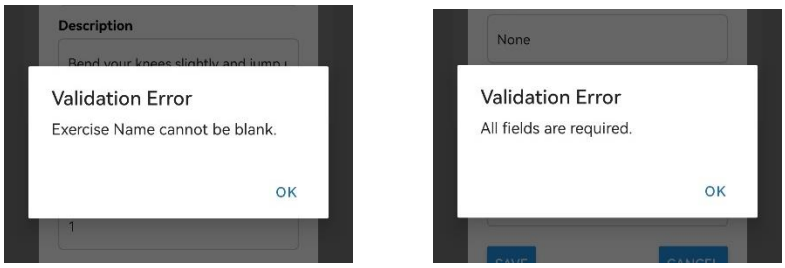


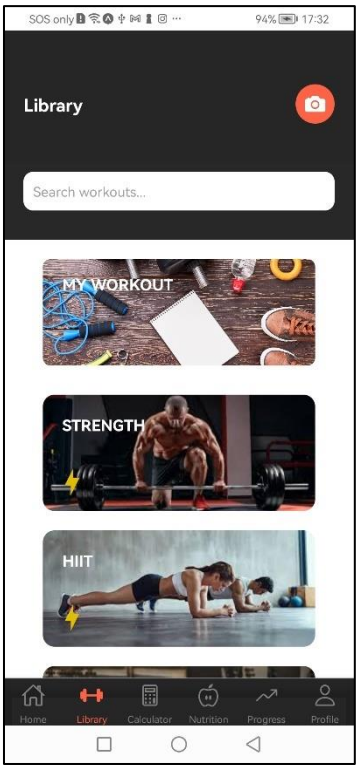
Figure 6.13: List of Workout Customization Input

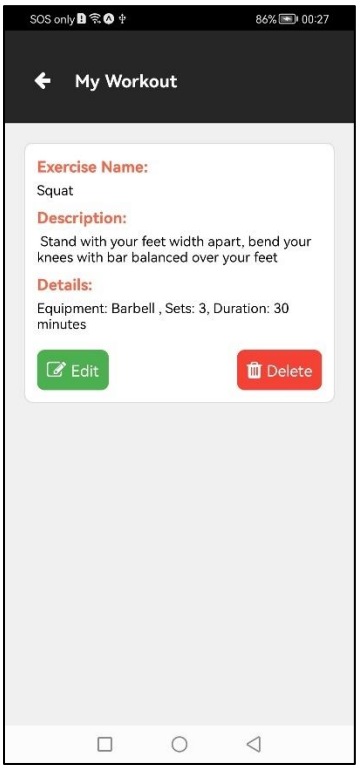


(a) (b)

Figure 6.14(a)(b): Validation Error for Workout Customization Input

After that, users can click on the “Create” button to save their personalized workouts by sending the POST request to the backend using axios. If the exercise is created successfully, a successful message will be prompted, and the exercises will be added to the workout library under the “MyWorkout” area.





(a) (b)

Figure 6.15(a)(b): Sample Of Workout Customization Generated under “MyWorkout” Section

```
  try {
    const userId = await AsyncStorage.getItem('userId'); // Retrieve userId from AsyncStorage
    const response = await axios.post('http://192.168.0.173:3000/exercises', {
      userId, // Include userId in the request body
      exerciseName,
      description,
      equipment,
      sets,
      duration,
    });

    Alert.alert('Success', 'Exercise added successfully!');
    // Optionally clear inputs here
    setExerciseName('');
    setDescription('');
    setEquipment('');
    setSets('');
    setDuration('');
  } catch (error) {
    Alert.alert('Error', 'Failed to add exercise.');
    console.error(error);
  }
};
```

Figure 6.16: Code Snippet of Add Workout Customization (frontend)

```
app.post('/exercises', (req, res) => {
  const { userId, exerciseName, description, equipment, sets, duration } = req.body;
  const query = 'INSERT INTO exercises (userId, exerciseName, description, equipment, sets, duration) VALUES (?, ?, ?, ?, ?, ?)';

  db.query(query, [userId, exerciseName, description, equipment, sets, duration], (err, result) => {
    if (err) {
      console.error('Error executing query:', err);
      return res.status(500).json({ error: err.message });
    }
    res.status(201).json({ message: 'Exercise added successfully!', data: result });
  });
});
```

Figure 6.17: Code Snippet of Add Workout Customization (backend)

In the “MyWorkout” area, users have the option to modify their workout inputs. Upon clicking on the edit button, the system displays the worklist details to be edited. When the user edits the workout details, the system verifies if no fields are left blank and the parameters for set and duration are numeric. After that, the modified information will be dynamically updated from the database, providing users with a corrected and updated workout instantaneously.

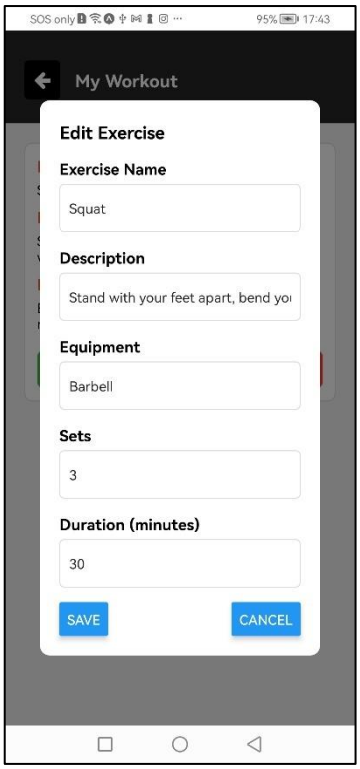


Figure 6.18: Inputs of Edit Workout Customization

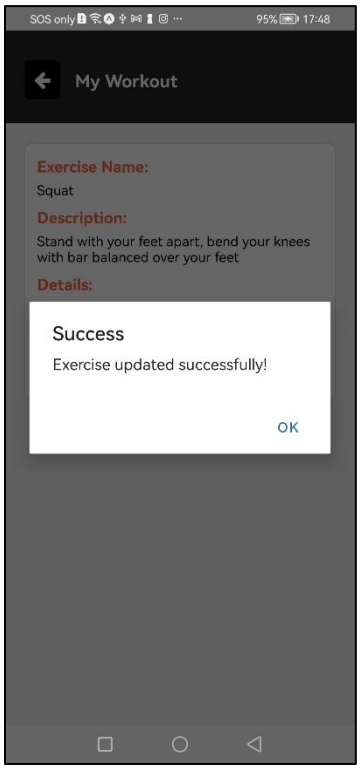


Figure 6.19: Successful Message of Edit Workout Customization

```
const handleSave = async () => {
  // Validation checks
  if (!selectedExercise.exerciseName || !selectedExercise.description || !selectedExercise.equipment ||
      !selectedExercise.sets || !selectedExercise.duration) {
    Alert.alert('Validation Error', 'All fields are required.');
    return;
  }

  if (isNaN(parseInt(selectedExercise.sets, 10)) || isNaN(parseInt(selectedExercise.duration, 10))) {
    Alert.alert('Validation Error', 'Sets and Duration must be numeric.');
    return;
  }

  try {
    if (selectedExercise) {
      const { id, exerciseName, description, equipment, sets, duration } = selectedExercise;
      const response = await axios.put(`http://192.168.0.173:3000/exercises/${id}`, {
        exerciseName,
        description,
        equipment,
        sets: parseInt(sets, 10),
        duration: parseInt(duration, 10),
      });
      Alert.alert('Success', 'Exercise updated successfully!');
      fetchExercises(); // Refresh exercises
      setModalVisible(false); // Close modal
    }
  } catch (error) {
    console.error('Error updating exercise:', error.response ? error.response.data : error.message);
    Alert.alert('Error', 'Failed to update exercise. Please try again.');
  }
};
```

Figure 6.20: Code Snippet of Edit Workout Customization (frontend)

```
// Route to update an exercise
app.put('/exercises/:id', (req, res) => {
  const exerciseId = req.params.id;
  const { exerciseName, description, equipment, sets, duration } = req.body;

  // Input validation
  if (!exerciseName || !description || !equipment || !sets || !duration) {
    return res.status(400).json({ error: 'All fields are required.' });
  }

  if (isNaN(parseInt(sets, 10)) || isNaN(parseInt(duration, 10))) {
    return res.status(400).json({ error: 'Sets and Duration must be numeric.' });
  }

  // Update query
  const query = `UPDATE exercises SET exerciseName = ?, description = ?, equipment = ?, sets = ?, duration = ? WHERE id = ?`;
  const values = [exerciseName, description, equipment, parseInt(sets, 10), parseInt(duration, 10), exerciseId];

  db.query(query, values, (err, result) => {
    if (err) {
      console.error('Error updating exercise:', err);
      return res.status(500).json({ error: 'Failed to update exercise.' });
    }

    if (result.affectedRows === 0) {
      return res.status(404).json({ error: 'Exercise not found.' });
    }

    res.json({ message: 'Exercise updated successfully!' });
  });
});
```

Figure 6.21: Code Snippet of Edit Workout Customization (backend)

Moreover, users can delete workouts that are no longer relevant by simply clicking on the delete button. The system will prompt a confirmation message in Figure 6.22 to the user if the user wishes to delete the workouts. This feature guarantees that user may adapt their workout plan to their changing demands, enhancing a seamless workout experience.

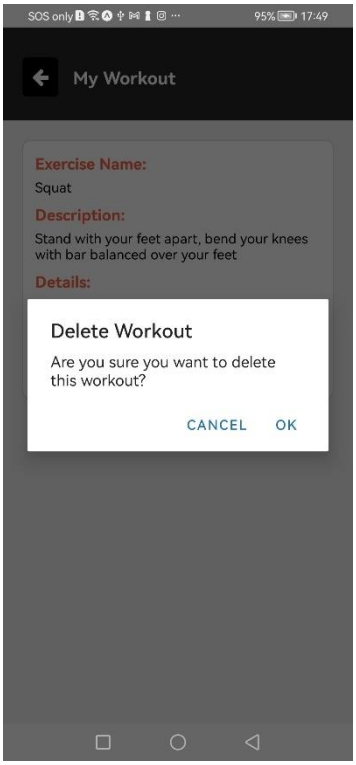


Figure 6.22: Confirmation Message on Delete Workout Customization

```
const handleDelete = async (exerciseId) => {
  try {
    Alert.alert(
      'Delete Workout',
      'Are you sure you want to delete this workout?',
      [
        {
          text: 'Cancel',
          style: 'cancel',
        },
        {
          text: 'OK',
          onPress: async () => {
            await axios.delete(`http://192.168.0.173:3000/exercises/${exerciseId}`);
            fetchExercises(); // Refresh exercises
          },
        },
      ],
      { cancelable: false }
    );
  } catch (error) {
    console.error('Error deleting exercise:', error);
  }
};
```

Figure 6.23: Code Snippet of Delete Workout Customization

6.7 Workout Recommendation

This mobile application features the workout recommendation which is available on the home screen. This feature analyzes data inputs from the user to generate personalized workout recommendations. The inputs collected from users are goals, level, focus area, duration, frequency and tools. According to the data inputs, the application generates 3 personalized workout recommendations for the user which is shown in Figure 6.24. Users can easily explore the recommended workouts with the exercise name, instructions, tools, repetitions and level. By leveraging these recommendations, user can easily obtain workouts that are aligned with their fitness input.

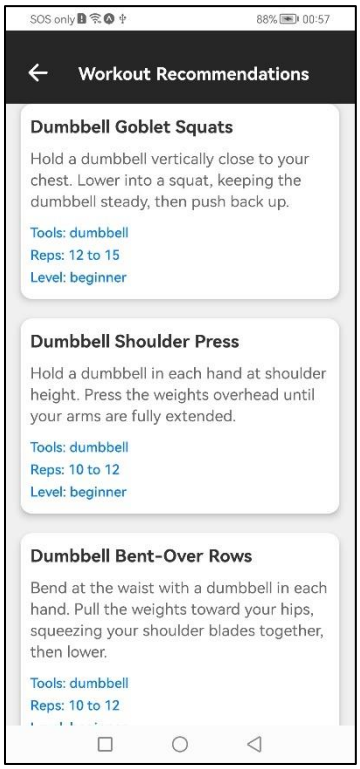
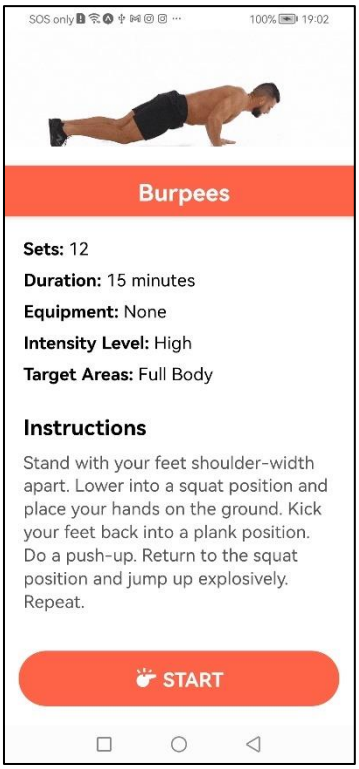


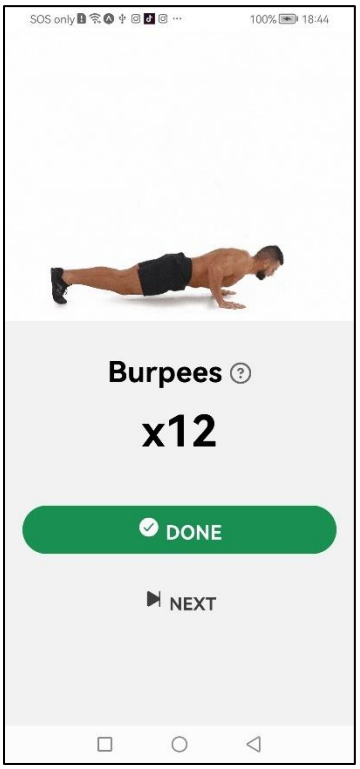
Figure 6.24: Output of Workout Recommendation

6.8 Workout Progress

This mobile application offers a valuable feature allowing the user to track and monitor their workout progress efficiently, providing a comprehensive list of past workouts alongside with the total number of workouts, total duration and calories burnt. Upon clicking on the Start button, the user will be logged into a specific workout, where the user are led through the workout step by step. The system keeps track of the current workout index and the number of sets for each workout. The figure 6.25 below depicts the logging session of a certain workout named Burpees.



(a)



(b)

Figure 6.25(a)(b): Log Workout

Once user click on the Next button after completing one of the sets of workouts, the user will be directed to the Rest Screen, illustrated in Figure 6.26. The system automatically counts down the remaining workouts, reducing them one by one until the entire workout session is completed. Users may designate their workouts as complete and the system will pass the workout statistics to the progress screen. This is where it provides a summary of the user's performance and progress, helping them to successfully reach their fitness objectives.

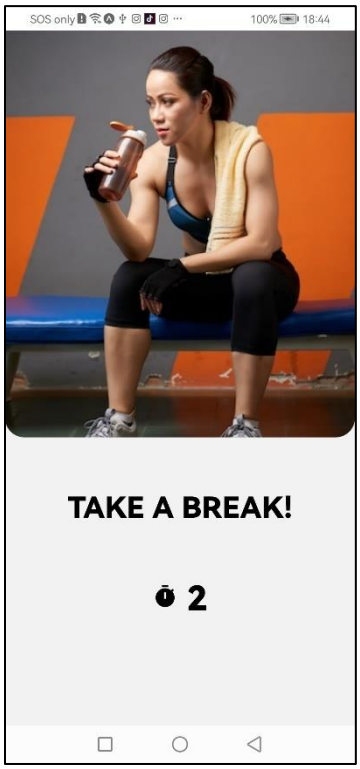


Figure 6.26: Rest Screen

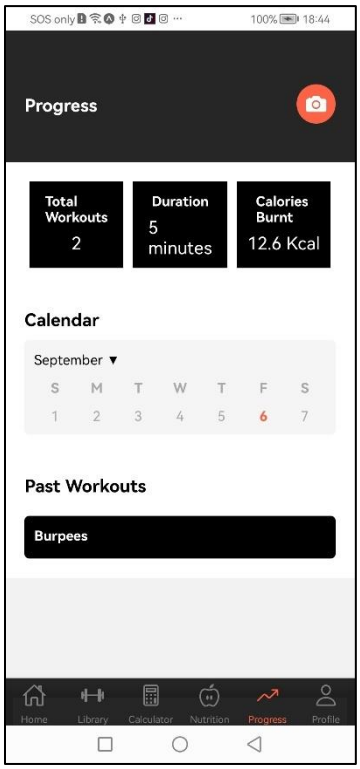


Figure 6.27: Output of Workout Progress

6.9 User Profile

On the user profile screen, user can view their personal details such as name, email address, age and gender. The screen also shows the user's data inputs for creating a personalized workout recommendation in addition to these personal details, such as goal, level, frequency, duration, focus area and tools. The user profile interface is divided into 2 sections, each intends to provide distinct and detailed information in an orderly manner. The personal details are added to the database and displayed via the profile screen once the user has registered their account. As for the data input section, once users have made selections on the data inputs for workout recommendations, it will be shown on the profile screen as well. The figure below displays the user profile modal and code snippet with personal details and data inputs.

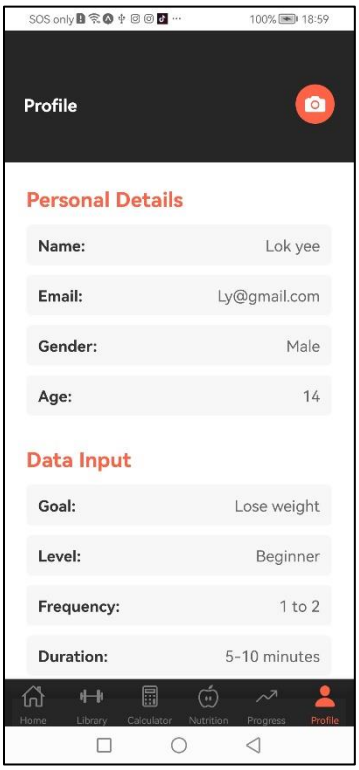


Figure 6.28: User Profile Modal for Personal Details

```
const fetchUserDetails = async () => {
  try {
    const userId = await AsyncStorage.getItem('userId');
    if (!userId) {
      console.error('User ID not found');
      return;
    }

    const response = await axios.get(`http://192.168.0.173:3000/userDetails?userId=${userId}`);
    const userData = response.data.user;

    setDetails({
      name: userData.name,
      email: userData.email,
      gender: userData.gender,
      age: userData.age.toString()
    });
  } catch (error) {
    console.error('Error fetching user details:', error);
    Alert.alert('Error', 'Failed to fetch user details. Please try again.');
  }
};
```

Figure 6.29: Code Snippet for Personal Details (frontend)

```
// GET user details by userId
app.get('/userDetails', async (req, res) => {
  const userId = req.query.userId; // Retrieve userId from query parameter
  const query = 'SELECT name, email, gender, age FROM users WHERE id = ?';

  try {
    db.query(query, [userId], (err, results) => {
      if (err) {
        console.error('Error fetching user details:', err);
        return res.status(500).json({ success: false, message: 'Error fetching user details' });
      }

      if (results.length > 0) {
        const user = results[0]; // Assuming userId is unique, so we expect only one result
        res.status(200).json({ success: true, message: 'User details retrieved successfully', user });
      } else {
        res.status(404).json({ success: false, message: 'User not found' });
      }
    });
  } catch (error) {
    console.error('Error fetching user details:', error);
    res.status(500).json({ success: false, message: 'Error fetching user details' });
  }
});
```

Figure 6.30: Code Snippet for Personal Details (backend)

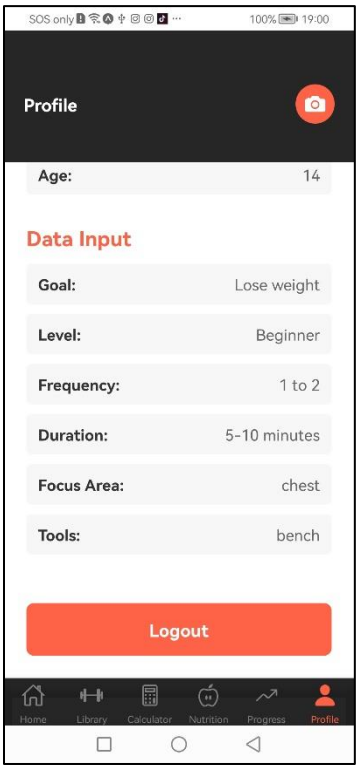


Figure 6.31: User Profile Modal for Data Inputs

```
app.get('/userDataInput', (req, res) => {
  const userId = req.query.userId;

  if (!userId) {
    console.error('Missing userId parameter');
    return res.status(400).send('Missing userId parameter');
  }

  console.log('Fetching data input for userId:', userId);

  const query = 'SELECT goal, level, frequency, duration, focus_area, tools FROM data_inputs WHERE userId = ?';
  db.query(query, [userId], (err, results) => {
    if (err) {
      console.error('Error fetching user data input:', err);
      return res.status(500).send('Error fetching user data input');
    }

    if (results.length === 0) {
      console.warn('User data input not found for userId:', userId);
      return res.status(404).send('User data input not found');
    }

    console.log('User data input fetched successfully:', results[0]);
    res.send({ dataInput: results[0] });
  });
});
```

Figure 6.32: Code Snippet for Data Inputs (frontend)

```
const fetchUserDataInput = async () => {
  try {
    const userId = await AsyncStorage.getItem('userId');
    if (!userId) {
      console.error('User ID not found');
      return;
    }

    const response = await axios.get(`http://192.168.0.173:3000/userDataInput?userId=${userId}`);
    const userDataInput = response.data.dataInput;

    setDataInput({
      goal: userDataInput.goal,
      level: userDataInput.level,
      frequency: userDataInput.frequency,
      duration: userDataInput.duration,
      focus_area: userDataInput.focus_area,
      tools: userDataInput.tools
    });
  } catch (error) {
    console.error('Error fetching user data input:', error);
    Alert.alert('Error', 'Failed to fetch user data input. Please try again.');
  }
};
```

Figure 6.33: Code Snippet for Data Inputs (backend)

Despite the profile screen is not allowed for the editing of personal details, the data inputs are intended to be dynamic and customizable. Users may easily adjust their data inputs if their fitness objectives change over time. The system makes sure that any changes made by the user are reflected in generating a new workout recommendation by permitting these updates. This feature is useful for user who wishes to change their goals and demand for the workout.

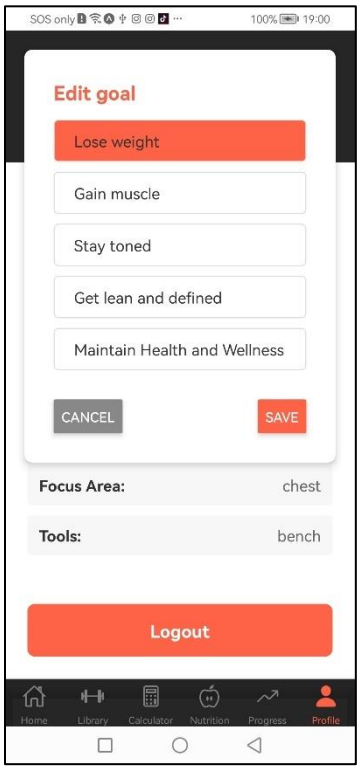


Figure 6.34: Sample of Updating User Data Inputs

```
const handleSave = async () => {
  try {
    const userId = await AsyncStorage.getItem('userId');
    if (!userId) {
      console.error('User ID not found');
      return;
    }

    // Update local state
    const updatedDataInput = { ...dataInput, [currentDetail]: currentValue };
    setDataInput(updatedDataInput);

    // Send PUT request to update database
    await axios.put(`http://192.168.0.173:3000/userDataInput/${userId}`, updatedDataInput);

    setModalVisible(false); // Close modal after successful update
  } catch (error) {
    console.error('Error updating user data input:', error);
    Alert.alert('Error', 'Failed to update user data input. Please try again.');
  }
};
```

Figure 6.35: Code Snippet for Updating User Data Inputs (frontend)

```
app.put('/userDataInput/:userId', (req, res) => {
  const userId = req.params.userId;
  const { goal, level, frequency, duration, focus_area, tools } = req.body;

  const query = `
    UPDATE data_inputs
    SET goal = ?, level = ?, frequency = ?, duration = ?, focus_area = ?, tools = ?
    WHERE userId = ?`;

  db.query(query, [goal, level, frequency, duration, focus_area, tools, userId], (err, results) => {
    if (err) {
      console.error('Error executing query:', err);
      return res.status(500).json({ success: false, message: 'Error updating user data input' });
    }

    console.log('User data input updated successfully');
    res.status(200).json({ success: true, message: 'User data input updated successfully' });
  });
});
```

Figure 6.36: Code Snippet for Updating User Data Inputs (backend)

6.10 Calorie and BMI Calculator

The calculator features in the mobile application are divided into two sections, which is designed to calculate the user's body mass index (BMI) and daily calorie intake.

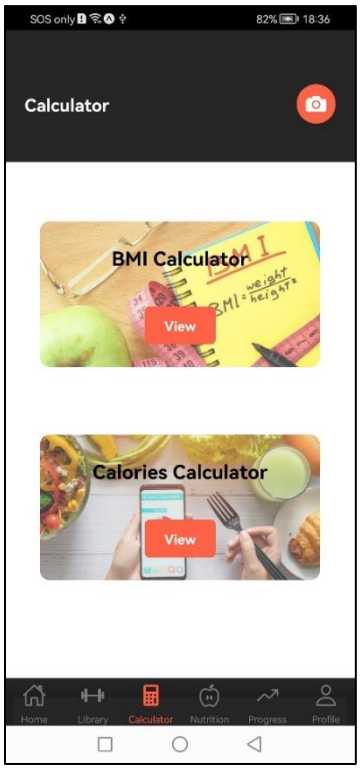


Figure 6.37: Calculator Screen

To calculate BMI, the system requires the user to input the height, weight, gender and age, which is used to compute the BMI value. When the BMI value is generated, the system categorizes it according to the range and displays the results to the user, whether the user is underweight, normal or obese. This is helpful for the users that are conscious of their weight and general health.

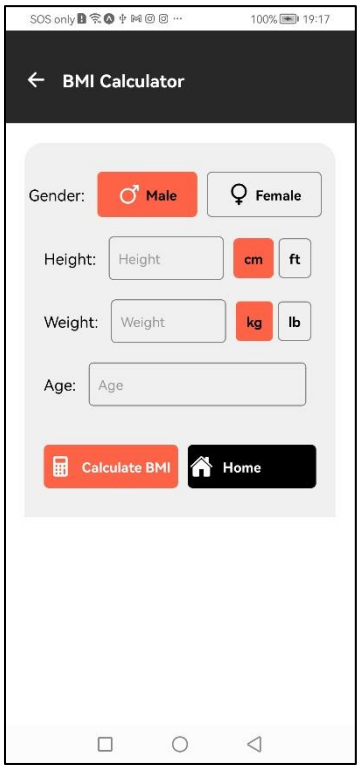


Figure 6.38: Input of BMI Calculator

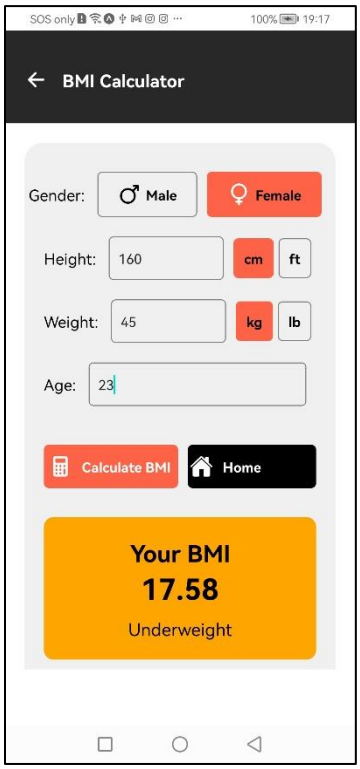


Figure 6.39: Sample of BMI Result

In order to determine the metabolic rate, the user must also enter the user's gender, height, weight, age, and activity level on the calorie calculator. It calculates the Total Daily Energy Expenditure based on the user's metabolic rate, which aids in figuring out how many calories the user needs to consume daily to efficiently accomplish their fitness and health objectives.

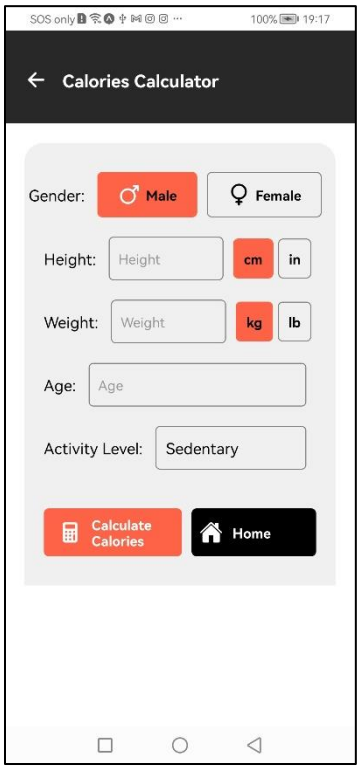


Figure 6.40: Input of Calorie Calculator

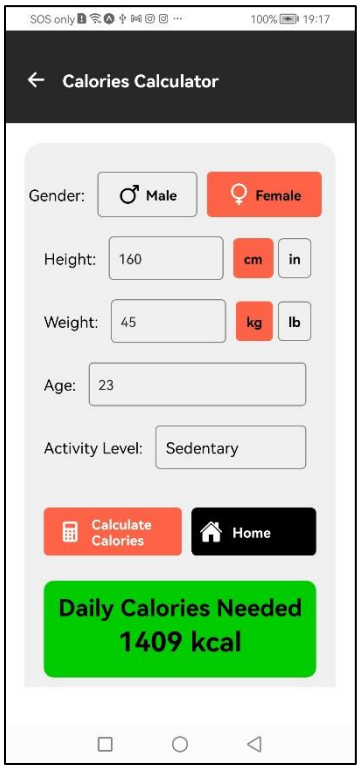
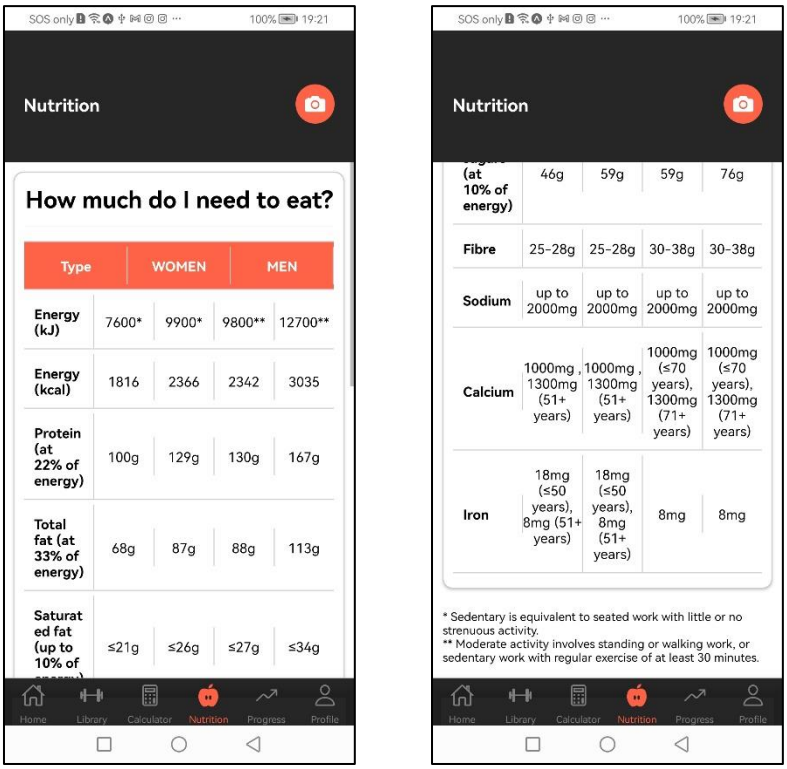


Figure 6.41: Sample of Calorie Result

6.11 Nutrition

The mobile application includes comprehensive nutrition information about dietary intake, dedicated to men and women only. It is tabulated to display the recommendations for daily intake of macronutrients and micronutrients, including carbohydrates, proteins, fats, vitamins and minerals. With the aid of this screen, users take into account on their own health by making well-informed nutritional decisions in support of their wellness and fitness goals.



(a) (b)

Figure 6.42(a)(b): Nutrition table

6.12 Gym equipment recognition with AI model

The gym equipment recognition is managed by an AI model, where users scan the gym equipment using the camera from the mobile application. The models are tested to provide the interpretation and analysis of the results in terms of evaluation metrics.

6.12.1 Data Collection and Preprocessing

The dataset contains 3000 gym equipment images consisting of 10 categories: aerobic steppers, barbell, bench press, dumbbell, elliptical machine, kettlebell, multi-machine, pull bar, rowing machine and treadmill, which provide a comprehensive dataset for model training. The sample data set images for each of the gym equipment categories are shown in the Figure 6.43. In addition, a shuffling step is involved by shuffling the image, preventing the model from learning and recognizing the pattern. A preprocessing step is then executed to resize all the images to a pixel dimension of 224 × 224 and a data augmentation technique such as image rotation, translation, and zoom range is implemented

on the training dataset to improve model generalization capabilities. After that, the dataset was divided for training with 2400 images (80 %) and testing with 600 images (20%).

Figure 6.43: Data Set Images

6.12.2 Model Description

VGG16, ResNet50 and InceptionV3 are chosen to investigate which model proves to recognize gym equipment most cost-effective and accurately. Hence, the architecture explains the selection of these three models as the best alternatives for recognizing gym equipment.

6.12.2.1 VGG16

According to Kumaran, Y. (2024), VGG16 was developed by the Visual Graphics Group at Oxford University. This pre-trained model is distinguished by its depth, consisting of 16 layers in total with weights, including 13 convolutional layers and 3 fully connected layers. The architecture is known for its uniform use of 3x3 convolutional layers throughout, which are stacked in increasing depth as the network progresses.

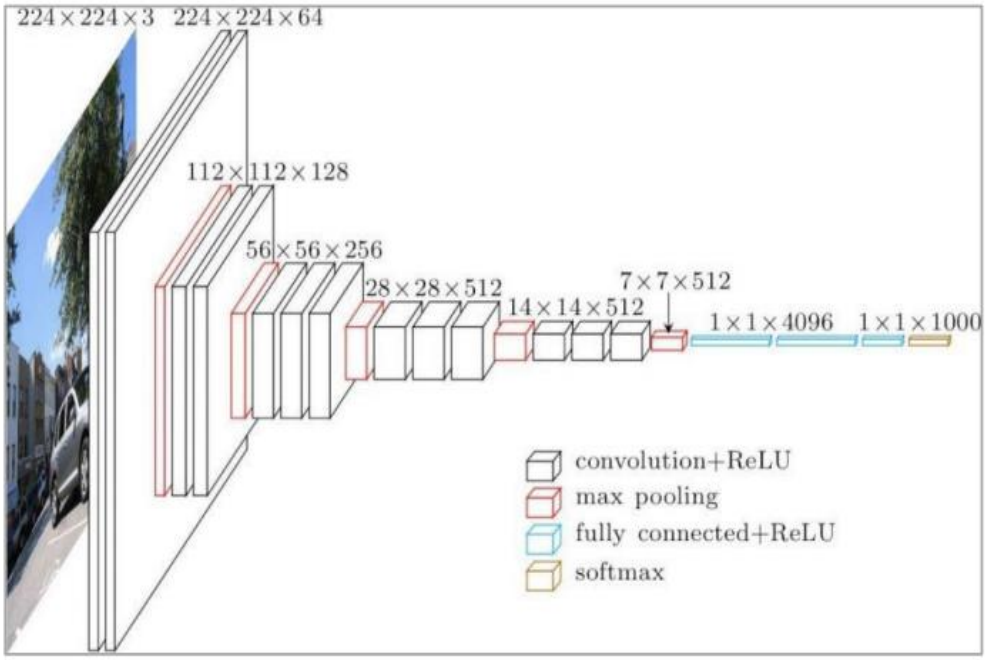


Figure 6.44: VGG-16 Architecture (Hassan, 2018)

The first layer of VGG16 accepts a 224x224 image as input. This input is passed through convolutional layers of 3x3 kernels with 1 stride, followed by a max pooling layer that utilizes 2x2 kernels with 2 strides. After that, it passes through 3 fully connected layers, with 4096 neurons in the first 2 layers and 1000 neurons in the softmax layer.

The model is designed using a stack of increasingly deeper max-pooling layers followed by a series of convolutional layers. Despite its computational intensity, the model enables to learn complex data, which leads to a widespread reputation for producing reliable and accurate predictions. Furthermore, the VGG16 model, which was first presented by Karen Simonyan and Andrew Zisserman in 2014, has obtained top ranks in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition in terms of classifying pictures into 1000 categories and recognizing objects from 200 classes. Hence, VGG-16 is one of my top picks for a model in my gym equipment recognition task because of its simplicity, versatility, popularity, and great performance.

6.12.2.2 ResNet50

Ikechukwu (2021) has described the architecture of the ResNet50 model as a 50-layer convolutional neural network that consists of 48 convolutional layers, one MaxPool layer, and one average pool layer. When deeping down the network, the problem of gradient vanishing arises by preventing the first few hidden layers from updating their weights, thus slowing down the training process of the model. ResNet50, on the other hand, can bypass one or more model layers to solve the vanishing gradient issue. As a result, the network can acquire more accurate representations of the input data.

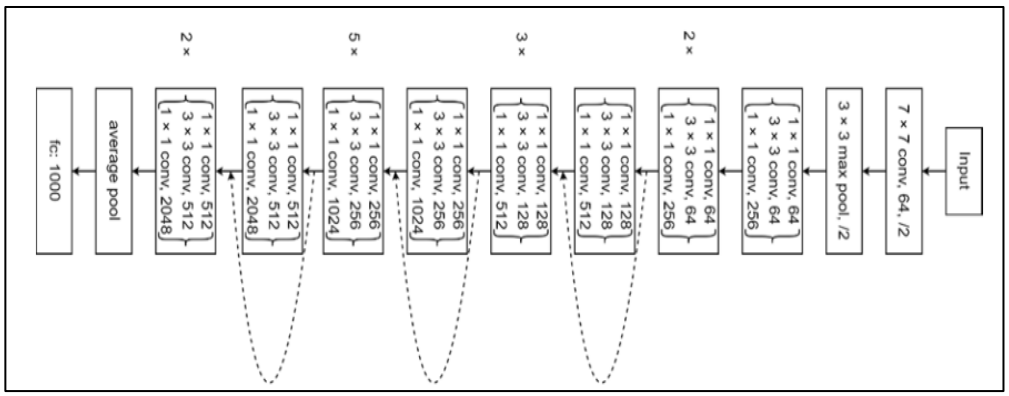


Figure 6.45: The architecture of ResNet50

Additionally, the ILSVRC imagenet competition was won in 2015 using the ResNet50 design. Çınar, A., 2021 also mentioned the model's widespread adoption results from its ability to facilitate training with smaller data sets, proven highly effective in lowering the computing expenses. Hence, implementing Inception V3 model is the optimal choice despite its advantages.

6.12.2.3 InceptionV3

InceptionV3 is a convolutional neural network that are developed by Google in 2015. It starts with a series of convolutional, max pooling, and inception modules to learn various features from images at multiple scales and resolutions simultaneously, thereby increase the ability to recognize complex patterns in images. The inception modules covered a few optimizers in the model which are discussed below:

i. Factorization into Smaller Convolutions

The model breaks down the large filter size from 5x5 convolution into smaller filters 3x3. This factorization not only reduces the parameter count, it also leads to cost savings while preserving the representational power of the network which promotes the extraction quality of images.

ii. Use of Auxiliary Classifiers

InceptionV3 also includes auxiliary classifiers to improve the convergence and learning stability of deep networks (Szegedy et al., 2015). However, it is used as a regularizer and does not contribute to the final predictions due to no improvements shown.

iii. Efficient Grid Size Reduction

Instead of using the traditional way of pooling layers, Inception V3 employs a stride convolutions approach to reduce computational cost as well as maintain high dimensional sparse representation.

iv. Model Regularization via Label Smoothing

Label smoothing is a regularization technique to prevent the model from becoming overfitting, thereby ensure a more robust model.

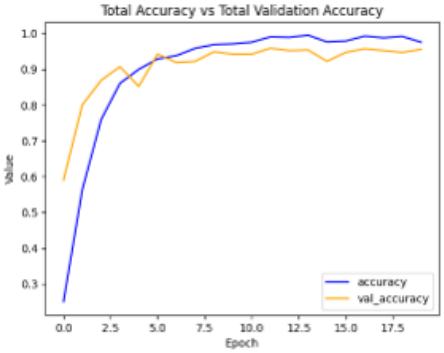
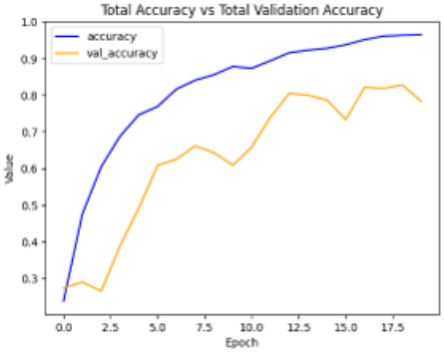
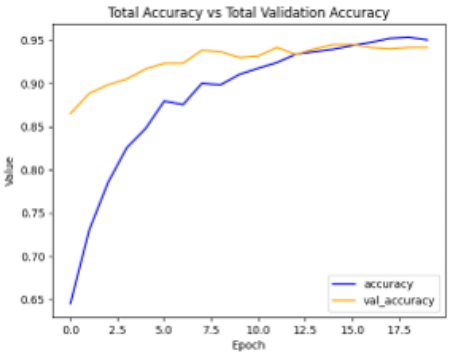
By rethinking the inception architecture results on the optimizer in reducing the parameters and increasing computational efficiency as well as becoming the 1st runner for image classification in the ILSVRC Competition 2015, Inception V3 is chosen to be involved in gym equipment recognition.

6.12.3 Model Evaluation

3 of the models are trained for 20 epochs and a batch size of 32. Additionally, they are employed with an early stopping callback to prevent overfitting. Once done training, the tables are shown below for the comparison between 3 models.

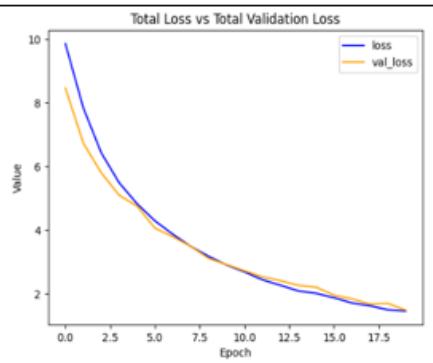
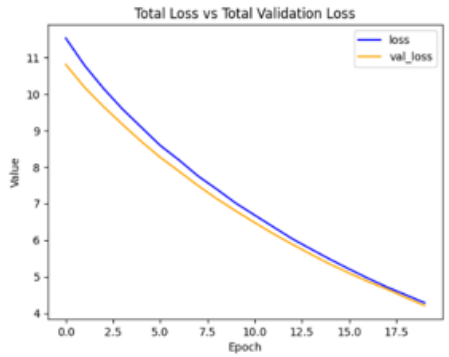
i. Accuracy Comparison

Table 6.1: Accuracy Comparison

Metrics/ Model	Accuracy	Validation Accuracy	Graph
VGG16	0.9864	0.9550	
Resnet50	0.9695	0.7833	
Inception V3	0.9448	0.9417	

ii. Loss Comparison

Table 6.2: Loss Comparison

Metrics/ Model	Loss	Validation Loss	Graph
VGG16	1.4381	1.4843	
Resnet50	2.2131	2.7460	
Inception V3	4.3445	4.2099	

iii. Training Time

Table 6.3: Training Time

Metric/ Model	VGG16	InceptionV3	Resnet50
Training time per step	8 seconds	2 seconds	2 seconds
Total training time	144 seconds	34 seconds	45 seconds

iv. Performance

Table 6.4: Performance

Metrics /Model	VGG16	InceptionV3	Resnet50
Accuracy	0.955	0.9417	0.7833
Recall	0.955	0.9417	0.7833
Precision	0.9561	0.9421	0.8302
F1-score	0.955	0.9411	0.7871

Out of the three models, VGG16 performs the best, with the greatest accuracy of 0.9864 and the highest validation accuracy of 0.9550. Its small loss values—a training loss of 1.4841 and a validation loss of 1.4843—further enhance this impressive performance.

This model shows a remarkable capacity to generalize successfully from the training set to the validation set. However, it may require more computational resources due to the depth and complexity of the model, with the greatest training time per step of 8 seconds and a total training duration of 144 seconds.

Among the above 3 models, VGG16 outperforms in terms of accuracy, recall, precision and F1-score. It has accuracy and recall of 0.955, precision of 0.9561 and F1-score of 0.955, ranking first across all metrics adopted. This model is highly effective in correctly identifying the classes (high Recall), and it has a low rate of false positives (high Precision). The balanced F1-score is further equally strong in both Precision and Recall, indicating that it performs well in identifying correct classes and minimizing both false positives and false negatives. In addition, the balanced F1-score has comparable strength in Precision and Recall, suggesting that it is effective at classifying correct classes and reducing false positives and false negatives.

Hence, the VGG16 model distinguish itself from the other two models due to its robustness of good performance under different conditions, where all the categories of gym equipment can be classified appropriately most by convolutional neural networks. These remarkable metrics highlight the module’s ability to correctly identify gym equipment based on publicly available datasets and solidify its position as the top performer across all evaluated measures. In conclusion, the selected AI model for this gym

equipment recognition task is the VGG16 model, greatly improving the usability of the application by providing the precise gym equipment name and workout instructions.

6.12.3 Capture Gym Equipment Images

Upon scanning the gym equipment, the camera component of the mobile application captures the image of the gym equipment. The image is then resized to 224x224 so that the model can achieve better recognition. Subsequently, the picture is transmitted to the Flask server from the mobile application, which is shown in the figure below.

```
const handleCapture = async () => {
  if (cameraRef.current) {
    try {
      setLoading(true);
      const photo = await cameraRef.current.takePictureAsync({ base64: true });
      const resizedPhoto = await ImageManipulator.manipulateAsync(
        photo.uri,
        [{ resize: { width: 224, height: 224 } }],
        { compress: 0.7, format: ImageManipulator.SaveFormat.JPEG }
      );

      const fileUri = resizedPhoto.uri;
      const formData = new FormData();
      formData.append('file', {
        uri: fileUri,
        type: 'image/jpeg',
        name: 'photo.jpg',
      });

      const response = await axios.post('http://192.168.0.173:5000/recognize', formData, {
        headers: {
          'Content-Type': 'multipart/form-data',
        },
      });
```

Figure 6.46: Gym Equipment Image Processing

The server loads the h5 file-formatted, pre-trained deep learning model that includes a variety of exercise equipment. Once the server processes the image and identifies the equipment, it sends back the equipment name and workout instructions to the mobile application. This allows users to rapidly grasp the proper usage of different exercise equipment. Moreover, instruction like “Please place your gym equipment inside the frame” is displayed at the top of the screen to increase the accuracy of gym equipment recognition. Figure 6.47 illustrates the screen of the gym equipment recognition scanning with the output results displayed in Figure 6.48.

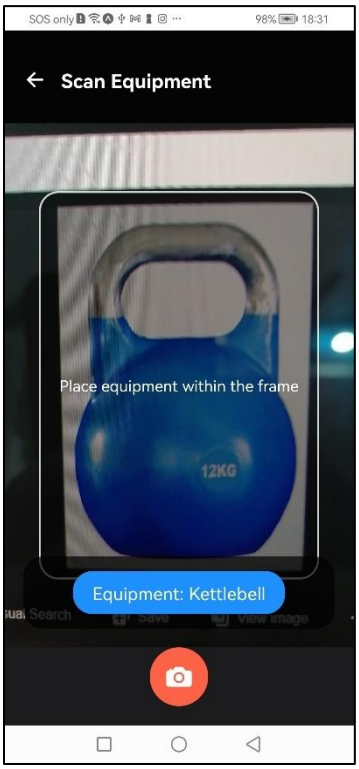


Figure 6.47: Gym Equipment Recognition Scanning

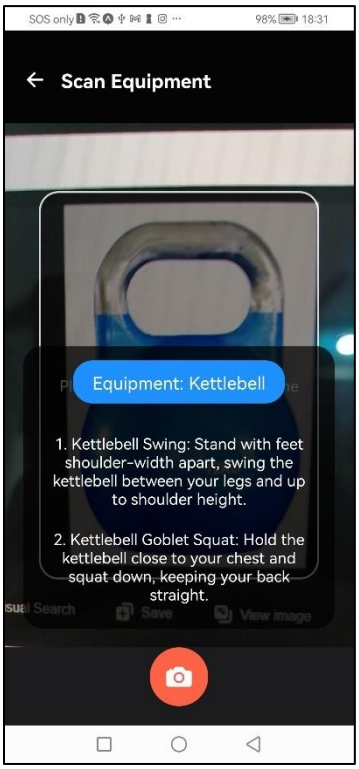
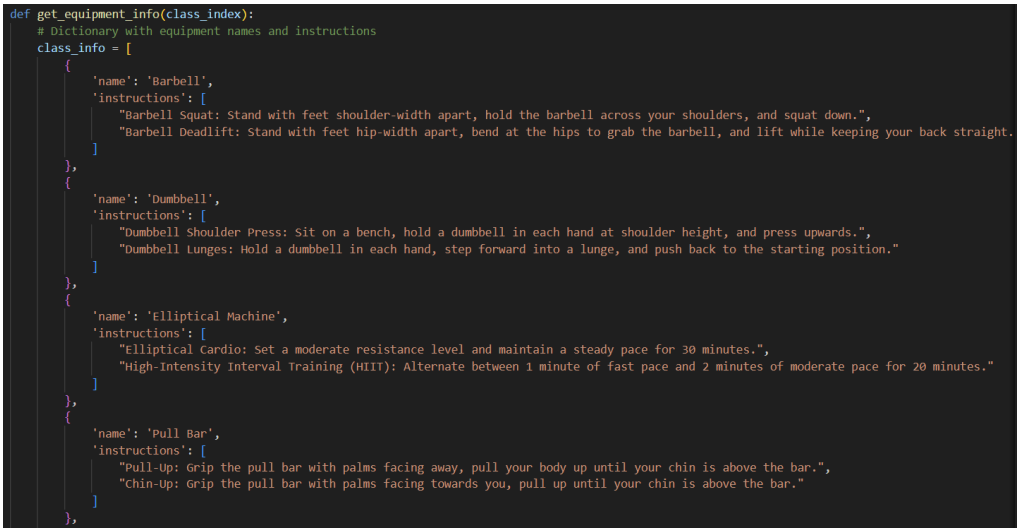


Figure 6.48: Output of Scanned Result

In the case of the scanned result, the system will return the result with comprehensive details such as equipment names with two guidances. Figure 6.49 shows an example of this data that is kept in a Python file.



```
def get_equipment_info(class_index):
    # Dictionary with equipment names and instructions
    class_info = [
        {
            'name': 'Barbell',
            'instructions': [
                "Barbell Squat: Stand with feet shoulder-width apart, hold the barbell across your shoulders, and squat down.",
                "Barbell Deadlift: Stand with feet hip-width apart, bend at the hips to grab the barbell, and lift while keeping your back straight."
            ]
        },
        {
            'name': 'Dumbbell',
            'instructions': [
                "Dumbbell Shoulder Press: Sit on a bench, hold a dumbbell in each hand at shoulder height, and press upwards.",
                "Dumbbell Lunges: Hold a dumbbell in each hand, step forward into a lunge, and push back to the starting position."
            ]
        },
        {
            'name': 'Elliptical Machine',
            'instructions': [
                "Elliptical Cardio: Set a moderate resistance level and maintain a steady pace for 30 minutes.",
                "High-Intensity Interval Training (HIIT): Alternate between 1 minute of fast pace and 2 minutes of moderate pace for 20 minutes."
            ]
        },
        {
            'name': 'Pull Bar',
            'instructions': [
                "Pull-Up: Grip the pull bar with palms facing away, pull your body up until your chin is above the bar.",
                "Chin-Up: Grip the pull bar with palms facing towards you, pull up until your chin is above the bar."
            ]
        },
```

Figure 6.49: Sample of Gym Equipment Name with Instructions in Python File

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

Testing plays as an important role in software development to discover and fix bugs, ensuring that the product meets specifications for functionality and performance. To verify and assess the system, a number of tests are run, including unit, usability and user acceptance tests.

7.2 Unit Testing

Unit testing is an essential software development technique that entails independently evaluating each program component to ensure that it is correct. By testing each of these components separately, developers can make sure it functions well before evaluating the program as a whole. The tables below show the test cases for unit test.

Table 7.1: Unit Test Case – User Register

Test Case ID	TC-001	**Module Name**	Register Module	
Test Title	Unit Test Case for Register Account			
Pre-condition	-			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
Input new email and valid password criteria	1. Input a valid name. 2. Input a valid email. 3. Input a valid password. 4. Choose a gender. 5. Input a valid age. 6. Click the Register button.	The system directs to the data input page with the account registered.	The system directs to the data input page with the account registered.	Pass
Input existing email	1. Input a valid name 2. Input an existing email. 3. Input a valid password. 4. Choose a gender. 5. Input a valid age. 6. Click the Register button.	The system displays an error message of invalid email.	The system displays an error message of invalid email.	Pass

Password does not meet criteria	1. Input a valid name. 2. Input a valid email. 3. Input an invalid password. 4. Choose a gender. 5. Input a valid age. 6. Click the Register button.	The system displays an error message of an invalid password.	The system displays an error message of an invalid password.	Pass
Empty inputs	1. Click the Register button.	The system displays the alert message of invalid inputs.	The system displays the alert message of invalid inputs.	Pass
Input invalid format of email	1. Input a valid name. 2. Input an invalid email format. 3. Input a valid password. 4. Choose a gender. 5. Input a valid age. 6. Click the Register button.	The system displays an error message of invalid email format.	The system displays an error message of invalid email format.	Pass

Input invalid format of password	1. Input a valid name. 2. Input a valid email. 3. Input an invalid password format. 4. Choose a gender. 5. Input a valid age. 6. Click the Register button.	The system displays an error message of invalid password format.	The system displays an error message of invalid password format.	Pass
Input invalid format of age	1. Input a valid name. 2. Input a valid email. 3. Input a valid password. 4. Choose a gender. 5. Input an invalid age format. 6. Click the Register button.	The system displays an error message of invalid age format.	The system displays an error message of invalid age format.	Pass

Table 7.2: Unit Test Case – User Login

Test Case ID	TC-002	**Module Name**	Login Module
Test Title	Unit Test Case for Login Account		
Pre-condition	User has a registered account		

Test Case Description	Execution Steps	Expected Result	Actual Result	Status
Input registered email and correct password	1. Input a valid email. 2. Input a valid password. 3. Click the Login button.	The system redirects to the logged in home page.	The system redirects to the logged in home page.	Pass
Input an unregistered email and correct password	1. Input an invalid email. 2. Input valid password. 3. Click the Login button.	The system displays an error message of unregistered email.	The system displays an error message of unregistered email.	Pass
Input incorrect password	1. Input a valid email. 2. Input an incorrect password. 3. Click the Login button.	The system displays error message of incorrect password.	The system displays error message of incorrect password.	Pass
Empty inputs	1. Click the Login button.	The system displays alert message of invalid inputs.	The system displays alert message of invalid inputs.	Pass

Input invalid format of email	1. Input invalid email format. 2. Input valid password. 3. Clicks the Login button	The system displays error message of invalid email format.	The system displays error message of invalid email format.	Pass

Table 7.3: Unit Test Case – View Workout Library

Test Case ID	TC-003	**Module Name**	Workout Module	
Test Title	Unit Test Case for User View Workout Library			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
View workout library categories	1. Click on the workout library icon at the bottom tab.	The system displays the workout library categories.	The system displays the workout library categories.	Pass
View workout library list	1. Click the workout categories tab in the Workout Library page.	The system displays the workout list page for selected categories.	The system displays the workout list page for selected categories.	Pass

View workout information	1. Click on the selected workout tab on the Workout List page.	The system displays the workout details on selected workouts.	The system displays the workout details on selected workouts.	Pass

Table 7.4: Unit Test Case – Search and Filter Workout

Test Case ID	TC-004	**Module Name**	Workout Module	
Test Title	Unit Test Case for User Search and Filter Workout			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
Input keywords for workout searching	1. Input a keyword in the search textbox. 2. Click the Apply button.	The system displays the keyword search filtered workout list.	The system redirects to the workout list page.	Pass
Empty input for workout searching	1. Click the Apply button.	The system displays the workout list with no search filter.	The system displays the workout list with no search filter.	Pass

Filter by equipment	1. Select the equipment option. 2. Click the Apply button.	The system displays the workout list with filtered equipment.	The system displays the workout list with filtered equipment.	Pass
Remove filter of equipment	1 . Click the Reset button	The system displays workout list with no equipment filter.	The system displays workout list with no equipment filter.	Pass
Filter by intensity	1. Select the intensity option. 2. Click the Apply button.	The system displays the workout list with filtered intensity.	The system displays the workout list with filtered intensity.	Pass
Remove filter of intensity	1. Click the Reset button	The system displays workout list with no intensity filter.	The system displays workout list with no intensity filter.	Pass
Filter by target area	1. Select the target area option. 2. Click the Apply button.	The system displays the workout list with filtered target area.	The system displays the workout list with filtered target area.	Pass
Remove filter of target area	1. Click the Reset button.	The system displays workout list with no target area filter.	The system displays workout list with no target area filter.	Pass

Table 7.5: Unit Test Case – View Nutrition

Test Case ID	TC-005	**Module Name**	Nutrition Module	
Test Title	Unit Test Case for User View Nutrition			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
View nutritional information	1. Click on the nutrition icon at the bottom tab.	The system displays the nutritional information.	The system displays the nutritional information.	Pass

Table 7.6: Unit Test Case – View Workout Recommendation

Test Case ID	TC-006	**Module Name**	Workout Module	
Test Title	Unit Test Case for User View Workout Recommendation			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
View workout recommendation list	1. Click on the workout recommendation button.	The system displays the workout recommendation list	The system displays the workout recommendation list	Pass

Table 7.7: Unit Test Case – Edit User Profile

Test Case ID	TC-007	**Module Name**	User Profile Module	
Test Title	Unit Test Case for User edit user profile			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
Edit user profile with valid inputs	1. Click on user data input tab. 2. Enter valid user data input details. 3. Click on save button.	The system displays the edited user profile details	The system displays the edited user profile details	Pass
Edit user profile with invalid or empty inputs	1. Input with invalid data input details or empty inputs. 2. Click on save button.	The system displays error message of the inputs	The system displays error message of the inputs	Pass

Table 7.8: Unit Test Case – View User Profile

Test Case ID	TC-008	**Module Name**	User Profile Module	
Test Title	Unit Test Case for User View User Profile			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**

View user profile	1. Click on the user profile icon at bottom tab. 2. Click on the user data input tab.	The system displays the user profile details.	The system displays the user profile details.	Pass

Table 7.9: Unit Test Case – Manage Customized Workout

Test Case ID	TC-009	**Module Name**	Workout Module	
Test Title	Unit Test Case for Manage Customized Workout			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
View customized workout list	1. Click on the workout library icon at the bottom tab. 2. Click on the MyWorkout tab.	The system displays the customized workout list.	The system displays the customized workout list.	Pass
Add customized workout with valid inputs	1. Input valid customized workout details. 2. Click on the Save button.	The system displays notification of created customized workout.	The system displays notification of created customized workout.	Pass
Add customized workout with invalid or empty inputs	1. Input invalid or empty customized workout details.	Ths system displays error message of the inputs.	Ths system displays error message of the inputs.	Pass

Edit customized workout with valid inputs	1. Click on customized workout input tab. 2. Enter valid customized workout details. 3. Click on the Save button.	The system displays edited customized workout list.	The system displays edited customized workout list.	Pass
Edit customized workout with invalid or empty inputs	1. Click on customized workout input tab. 2. Enter invalid or empty customized workout details. 3. Click on the Save button.	The system displays error message of the inputs.	The system displays error message of the inputs.	Pass
Delete customized workout	1. Click on the Delete button.	The system displays notification of deleted customized workout.	The system displays notification of deleted customized workout.	Pass

Table 7.10: Unit Test Case – Generate Workout Recommendation

Test Case ID	TC-010	**Module Name**	Workout Module	
Test Title	Unit Test Case for User Generate Workout Recommendation			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**

Generate workout recommendation	1. Click on the data input tab in data input page.	The system generates the workout recommendation	The system generates the workout recommendation	Pass

Table 7.11: Unit Test Case – Calculate BMI and Calorie

Test Case ID	TC-011	**Module Name**	Calculator Module	
Test Title	Unit Test Case for User Calculate BMI and Calorie			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
Calculate BMI and Calorie with valid inputs	1. Click on the calculator icon at the bottom tab. 2. Click on BMI or Calorie calculator tab. 3. Input valid BMI and Calorie details. 4. Click on the Save button.	The system displays the BMI and calorie result.	The system displays the BMI and calorie result.	Pass
Calculate BMI and Calorie with invalid or empty inputs	1. Click on the calculator icon at the bottom tab. 2. Click on BMI or Calorie calculator tab.	The system displays error message of the inputs.	The system displays error message of the inputs.	Pass

	3. Input invalid or empty BMI and Calorie details. 4. Click on the Save button.			

Table 7.12: Unit Test Case – Capture Gym Equipment

Test Case ID	TC-012	**Module Name**	AI Gym Equipment Recognition Module	
Test Title	Unit Test Case for User Capture Gym Equipment			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
Capture valid image	1. Click on the camera icon. 2. Capture valid image by clicking capture button.	The system displays the gym equipment name and guidance from AI model.	The system displays the gym equipment name and guidance from AI model.	Pass
Capture invalid image	1. Click on the camera icon. 2. Capture invalid image by clicking capture button.	The system displays error message of the image.	The system displays error message of the image.	Pass

Table 7.13: Unit Test Case – Track Workout Progress

Test Case ID	TC-013	**Module Name**	Workout Module	
Test Title	Unit Test Case for User Track Workout Progress			
Pre-condition	User has logged into his/her account.			
Test Case Description	**Execution Steps**	**Expected Result**	**Actual Result**	**Status**
View workout progress	1. Click on the workout progress icon at bottom tab.	The system displays the workout progress details.	The system displays the workout progress details.	Pass

7.3 System Usability Testing

System usability testing is carried out to determine if the system is user-friendly from the end user's standpoint by evaluating the functionality of the application as the user navigates through it. The evaluation's findings will have a direct impact on the final product's approval and user acceptability. The System Usability Scale (SUS), which is well-known for its efficacy in measuring user satisfaction and usability, has been chosen as the assessment metric to gauge system usability. The survey questions consist of only 10 statements which were developed by John Brooke are involved in the usability testing to gather feedback from users, as shown in the tables below.

System Usability Scale Questionnaire	**Strongly Disagree**				**Strongly Agree**
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

Figure 7.1: SUS Standard Template (Brooke.J, 1986)

Delving deeper into the specifics of the template mentioned above, we see that every question is rated on a 5-point Likert scale, with the options being "Strongly Disagree" to "Strongly Agree." The scale's objective is to enable users to assess how much they agree with each statement regard to their viewpoint and experience with the system.

To calculate the SUS score efficiently, the calculation of the odd-numbered statements is subtracting 1 from user ratings, while the scores for even-numbered statements are subtracted from 5 . After that, the scores are sum up as the total SUS score after they have been calculated. The entire score is then multiplied by 2.5 to get the final SUS score, which falls within the range of 0 to 100. Based on this scoring method, it assist in assessing the usability of the system with metrics as seen in the figure below.

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
< 51	F	Awful

Figure 7.2: SUS Grading Table

As compared to the table above, the high final SUS score indicates better usability. A score of more than 68 is usually regarded as good, and more than 80.5 as exceptional. Based on the usability test response result displayed in the table below, the system received an average usability score of 92.5%, equivalent to a Grade A rating. This suggests that the mobile application is very easy to use and intuitive. The SUS survey form can be referred to the appendices B.

Table 7.14: SUS Survey

Participants	**Scores for each Question**										**Total**	**Percentage**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**		
1	5	1	5	1	5	1	5	1	5	1	40	100%
2	5	1	5	1	4	1	5	1	5	1	39	97.5%
3	3	1	4	2	4	4	5	2	4	1	30	75%
4	5	1	5	1	5	1	5	1	5	1	40	100%
5	4	2	4	1	5	1	4	1	5	1	36	90%
Average SUS Score												92.5%
Grade	A				**Adjective Rating**						Excellent	

Furthermore, thorough test scenarios will be deliberately developed for users to provide feedback during the usability testing process. The purpose of these test scenarios is to provide systematic recommendations for assessing the usability, performance, and functionality of the website. Every scenario will have all the necessary details to replicate user interactions and experiences in the real world. The table below depicts the test scenarios for user.

Table 7.15: Test Scenario for User

ID	Test Scenario	Scenario Description
1	Register	Imagine that you are a guest and you wish to create an account as a member of the fitness mobile application. How would you create an account?
2	Login	Imagine that you are a member who wishes to login to the fitness mobile application. How would you login to your account?
3	View User Profile	Imagine that you wish to view your user profile. What steps would you take to view your profile details?
4	Edit User Profile	Imagine that you want to edit your data inputs, such as goal, target area. How would you edit your profile information?

5	Search and Filter Workout	Imagine upon browsing the workout list, you find that there are too many workouts, you wish to find specific workouts in the workout library. How would you perform search and filter operations?
6	Remove Workout Filter	Imagine that after applying filters to the workout library, you wish to undo the filter option. How would you remove the filterings applied?
7	Capture Gym Equipment	Imagine that you want to identify a piece of equipment using the mobile application. How would you capture and identify gym equipment using the app?
8	Generate Workout Recommendation	Imagine that you wish the system to generate workout recommendations personally for you. How would you provide data input?
9	View Workout Recommendation	Imagine that after generating workout recommendations, you want to see the details of the personalized workouts. How would you view your workout recommendations?
10	View Workout Library	Imagine that you are interested in browsing all available workouts in the app's library. What steps would you take to view the workout library details?
11	View Nutrition	Imagine that you want to view nutritional recommendations. How would you access the nutrition information screen?
12	Calculate BMI and Calorie	Imagine that you wish to calculate your BMI and Calorie using the app's calculator. How would you perform these calculations?
13	Manage Customized Workout	Imagine that you want to create, edit, or delete a customized workout routine in the app. How would you manage your customized workouts?

14	Track Workout Progress	Imagine that you wish to track workout progress in the app. How would you access the workout progress information?

7.4 User Acceptance Testing

In this project, there are a total of 5 users who are involved in the User Acceptance Testing (UAT). Users are provided with scenarios in order to complete the tests. The templates for users on the UAT findings are provided below. The UAT results are available in the appendices C.

Table 7.16: User Acceptance Testing Template for User

Mobile-Based Fitness Application				
Test Module	**Test Case ID**	**Test Scenario**	**Status**	**Comment**
Register	UAT-001	1. Able to register a newaccount	Pass	
Login	UAT-002	1. Able to login to the newly registered account	Pass	
Workout	UAT-003	1. Able to view workout library categories	Pass	
	UAT-004	2. Able to view workout library list	Pass	
	UAT-005	3. Able to view workout information	Pass	
	UAT-006	4. Able to apply search filter on workout list	Pass	
	UAT-007	5. Able to remove searchfilter on workout list	Pass	
	UAT-008	6. Able to apply equipment filter on workout list	Pass	

	UAT-009	7. Able to remove equipment filter on workout list	Pass	
	UAT-010	8. Able to apply intensity filter on the workout list	Pass	
	UAT-011	9. Able to remove intensity filter on workout list	Pass	
	UAT-012	10. Able to apply target area filter on workout list	Pass	
	UAT-013	11. Able to remove target area filter on workout list	Pass	
	UAT-014	12. Able to view workout recommendation list	Pass	
	UAT-015	13. Able to view customized workout list	Pass	
	UAT-016	14. Able to add customized workout	Pass	
	UAT-017	15. Able to edit customized workout	Pass	
	UAT-018	16. Able to delete customized workout	Pass	
	UAT-019	17. Able to generate workout recommendation	Pass	
	UAT-020	18. Able to track workout progress	Pass	
User Profile	UAT-021	1. Able to edit user profile	Pass	
	UAT-022	2. Able to view user profile	Pass	

Calculator	UAT-023	1. Able to calculate BMI and Calorie	Pass	
AI Gym Equipment Recognition	UAT-024	1. Able to capture image	Pass	

CHAPTER 8

CONCLUSION AND FUTURE WORKS

8.1 Conclusion

This project has successfully completed its four primary objectives in spite of a short timeline.

The first objective was to determine the features and functionalities to be incorporated into the mobile fitness application. This was accomplished by involving an in-depth review of the fitness app market, literature reviews, user surveys, and methodologies.

The second objective aimed to integrate AI gym equipment recognition into the mobile application to enhance user safety by providing information regarding each piece of equipment, thereby increasing user's awareness and understanding of their usage. This objective has been met as the AI model detects the equipment correctly to provide safety instructions and guidelines for the user workout's journey.

The third objective focused on enhancing user workout skills by incorporating features into the mobile application that promote awareness of workout varieties. This objective has been achieved by offering a wide variety of workouts that were catered to various fitness levels and objectives to improve their overall health.

The last objective involved conducting a thorough testing and evaluation of the integrated AI gym equipment recognition system within the mobile application to assess its user satisfaction. This objective has been fulfilled as the app was improved to enhance the overall user experience.

8.2 Limitations and Recommendations

Even though the objectives were effectively met throughout the project, several limitations were found within the application during the development and testing phases. These limitations are shown in the table below along with the recommendations for future revisions to optimize the performance of the application.

Table 8.1: Limitations and Recommendations for Future Work

No.	Limitations	Recommendations
1	Lack of community and social features	Features like leaderboards, post feed, workout interactions and forums can be added to the application to make the application more interactive. These additions will enable users to connect, discuss and interact with each other, promoting a sense of community in the application.
2	No integration with fitness devices	Another shortcoming of this project is the absence of fitness device integration. The application depends solely on user data input cannot fully represent their fitness journey. Hence, integrating fitness devices like Garmin, and Apple Watch can resolve this issue by synchronizing user heart rate and workout data through application connection. This can provide users with a more accurate and complete fitness journey.
3	Basic nutrition information	The basic nutrition information requires user to manually manage their nutrition intake. This manual process might lead to human error. By implementing a personalized nutrition feature, users can receive personalized dietary recommendations based on their requirements and objectives, which will decrease manual monitoring and increase accuracy.

4	No real-time workout feedback	The application offers progress tracking but it lacks of real-time workout feedback. Therefore, audio instructions may be included with real-time workout feedback to effectively solve this problem. For example, the user may monitor proper posture by using the phone's camera. Voice instructions can be used to get any real-time guidance or advice. This guarantees that exercises are performed safely and correctly, further improving the user experience.

REFERENCES

Adenowo, A.A. and Adenowo, B.A., 2013. Software engineering methodologies: a review of the waterfall model and object-oriented approach. *International Journal of Scientific & Engineering Research*, *4*(7), pp.427-434.

Admin (2023). *Common Types Of Gym Accidents And How To Avoid Them*. [online] HealthNord. Available at: https://www.healthnord.com/en-us/education/common-types-of-gym-accidents-and-how-to-avoid-them/.

Agarwal, R., Prasad, J., Tanniru, M. and Lynch, J., 2000. Risks of rapid application development. *Communications of the ACM*, *43*(11es), pp.1-es.

Alshamrani, A. and Bahattab, A., 2015. A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, *12*(1), p.106.

Anon, (2020). *Food Nutrition and the Importance of Dietary Guidelines for Healthy Living - Food Safety Helpline*. [online] Available at: https://foodsafetyhelpline.com/food-nutrition-and-the-importance-of-dietary-guidelines-for-healthy-living/.

Anon, (2022). *Software Prototype - The First Step in Custom Software Development - iTech*. [online] Available at: https://itechindia.co/blog/software-prototype-for-software-development/.

Aroral, H.K., 2021. Waterfall process operations in the fast-paced world: project management exploratory analysis. *International Journal of Applied Business and Management Studies*, *6*(1), pp.91-99.

Balaji, S. and Murugaiyan, M.S., 2012. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), pp.26-30.

Berger, H., Beynon-Davies, P. and Cleary, P., 2004. The utility of a rapid application development (RAD) approach for a large complex information Systems development. *ECIS 2004 Proceedings*, p.7.

Beynon-Davies, P. and Holmes, S., 2002. Design breakdowns, scenarios and rapid application development. Information and Software Technology, [online] 44(10), pp.579–592. https://doi.org/https://doi.org/10.1016/S0950-5849(02)00078-2.

Brooke, J., 1996. SUS: A quick and dirty usability scale. *Usability Evaluation in Industry*.

Chandra, V., 2015. Comparison between various software development methodologies. International Journal of Computer Applications, 131(9), pp.7–10.

Cloudinary. (n.d.). *Why Is AI Image Recognition Important and How Does it Work?* [online] Available at: https://cloudinary.com/guides/ai/why-is-ai-image-recognition-important-and-how-does-it-work [Accessed 23 Mar. 2024].
Dennis, A. and Internet Archive (2014). *Systems analysis and design*. [online] *Internet Archive*. Hoboken, NJ : Wiley. Available at: https://archive.org/details/systemsanalysisd0000denn_j0w2 [Accessed 30 Mar. 2024].

Despa, M.L., 2014. Comparative study on software development methodologies. *Database Systems Journal*, *5*(3).

De Smedt, T. and Daelemans, W., 2012. Pattern for python. *The Journal of Machine Learning Research*, *13*(1), pp.2063-2067.

Doshi, D., Jain, L. and Gala, K., 2021. Review of the spiral model and its applications. *Int. J. Eng. Appl. Sci. Technol*, *5*, pp.311-316.

DuBois, P., 2013. *MySQL*. Addison-Wesley.

Eisenman, B., 2015. *Learning react native: Building native mobile apps with JavaScript*. " O'Reilly Media, Inc.".

Fazal (2020). *A Comparative Analysis of RAD and Agile Technique for Management of Computing Graduation Projects*. [online] Available at: https://www.researchgate.net/publication/342068506_A_Comparative_Analysis_of_RAD_and_Agile_Technique_for_Management_of_Computing_Graduation_Projects [Accessed 31 Mar. 2024].

Geambaşu, C.V., Jianu, I., Jianu, I. and Gavrilă, A., 2011. Influence factors for the choice of a software development methodology. *Accounting and Management Information Systems*, *10*(4), pp.479-494.

Ikechukwu, A.V., Murali, S. and Honnaraju, B., 2023, August. COPDNet: An Explainable ResNet50 Model for the Diagnosis of COPD from CXR Images. In *2023 IEEE 4th Annual Flagship India Council International Subsections Conference (INDISCON)* (pp. 1-7). IEEE.

Kumaran S, Y., Jeya, J.J., Khan, S.B., Alzahrani, S. and Alojail, M., 2024. Explainable lung cancer classification with ensemble transfer learning of VGG16, Resnet50 and InceptionV3 using grad-cam. *BMC medical imaging*, *24*(1), p.176.

Mishra, N., Jahan, I., Nadeem, M.R. and Sharma, V., 2023, May. A comparative study of resnet50, efficientnetb7, inceptionv3, vgg16 models in crop and weed classification. In *2023 4th International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 1-5). IEEE.

Mukesh, C., Likhita, A. and Yamini, A., 2023, July. Performance Analysis of InceptionV3, VGG16, and Resnet50 Models for Crevices Recognition on Surfaces. In *International Conference on Data Science and Applications* (pp. 161-172). Singapore: Springer Nature Singapore.

Nikolaieva, A., n.d. 8 Best Software Development Methodologies. [online] Available at: [Accessed 1 April 2024]

Oladele, A., n.d. Top 15 Software Development Methodologies: Benefits and Drawbacks. [online] Available at: [Accessed 1 April 2024]

Oliveira, E.R., Branco, A.C., Carvalho, D., Sacramento, E.R., Tymoshchuk, O., Pedro, L., Antunes, M.J., Almeida, A.M. and Ramos, F., 2022. An Iterative Process for the Evaluation of a Mobile Application Prototype. *SN Computer Science*, *3*(4), p.262.

Pargaonkar, S., 2023. A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. *International Journal of Scientific and Research Publications (IJSRP)*, *13*(08).

Pillai, R., Sharma, A., Sharma, N. and Gupta, R., 2023, March. Brain tumor classification using VGG 16, ResNet50, and inception V3 transfer learning models. In *2023 2nd International Conference for Innovation in Technology (INOCON)* (pp. 1-5). IEEE.

Pressman, R.S. (2009.). *Software Engineering- A Practitioner's Approach by Roger S. Pressman.pdf*. [online] Available at: https://drive.google.com/file/d/0B4FvADGfA7T8S3lCNE1IZlpQc1E/view?usp=sharing.

Pressman, R.S. and Internet Archive (1997). *Software engineering : a practitioner's approach*. [online] *Internet Archive*. New York : McGraw-Hill. Available at: https://archive.org/details/softwareengineer0000pres [Accessed 31 Mar. 2024].
Rault, J., 2022. Top 7 Software Development Methodologies Every Developer Needs to Know. [online] Available at: [Accessed 24 March 2024].

Senarath, U.S., 2021. Waterfall methodology, prototyping and agile development. *Tech. Rep.*, pp.1-16.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sommerville, I. (2015). *GlobAl EdiTioN Software Engineering TENTH EdiTioN*. [online] Available at: https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf [Accessed 31 Mar. 2024].

Soobia.et.al., S. (2019). Analysis of Software Development Methodologies. *International Journal of Computing and Digital Systems*, [online] 8(5), pp.445–460. doi:https://doi.org/10.12785/ijcds/080502.

Sophia (2023). *The Ultimate Guide to Calculators for Health and Fitness*. [online] Medium. Available at: https://medium.com/@coolcalculator43/the-ultimate-guide-to-calculators-for-health-and-fitness-4d479e0bf2b7 [Accessed 23 Mar. 2024].

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Tram, N., 2022. All Advantages and Disadvantages of Prototype Model. [online] Available at: [Accessed 24 March 2024].

Vaswani, V., 2009. MySQL database usage & administration. McGraw Hill Professional.

Yıldırım, M. and Çınar, A., 2021. Classification of Skin Cancer Images with Convolutional Neural Network Architectures. *Turkish Journal of Science and Technology*, *16*(2), pp.187-195.

Young, D. (2013). *(PDF) Software Development Methodologies*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/255710396_Software_Development_Methodologies.

APPENDICES

Appendix A: Questionnaire

Survey Question For WellFit: A Smart Fitness Companion for Personalized Wellness App

Drae respondents, This survey aims to gather requirements, valuable insights and feedback from potential users and individuals that are interested in fitness and wellness. Hence, We would like to invite you to complete this survey as it aims to investigate on user preferences for the WellFit mobile app.. All the information and responses given by you will be kept confidential and will only be used for study purposes. Thank you.

* Indicates required question

Personal Information

1. Name: *

2. Email: *

3. Age categories: *

Mark only one oval.

- ◯ Teenagers (below 20)
- ◯ Adults (21 and above)
- ◯ Seniors (50 and above)

4. Gender *

Mark only one oval.

- ◯ Male
- ◯ Female

Preferences and Requirements

5. How would you rate your current fitness level? *

Mark only one oval.

- ◯ Beginner
- ◯ Intermediate
- ◯ Advanced

6. Have you used any platforms or applications for workout? *

Mark only one oval.

- ◯ Yes
- ◯ No

7. How important is it for you to have a mobile fitness app? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very important

8. How often do you exercise in a week? *

Mark only one oval.

- ◯ Less than once a week
- ◯ 1-2 times a week
- ◯ 3-4 times a week
- ◯ 5 or more times a week

9. What time of day do you prefer to work out? (Select all that apply) *

Check all that apply.

- ☐ Morning
- ☐ Afternoon
- ☐ Evening

10. What are your primary fitness goals? (Select all that apply) *

Check all that apply.

- [ ] Weight loss
- [ ] Muscle gain
- [ ] Overall health improvement
- [ ] Stress reduction

11. How do you currently track your fitness progress? *

Mark only one oval.

- ◯ Paper and pen
- ◯ Fitness app
- ◯ No tracking
- ◯ Other: _____

12. Do you utilize any equipment or tools during workout? *

Mark only one oval.

- ◯ Yes
- ◯ No

13. How often would you like to receive progress tracking updates and insights on your fitness journey? *

Mark only one oval.

- ◯ Daily
- ◯ Weekly
- ◯ Monthly

14. How likely are you with inputting personal data (e.g., age, gender, weight, goals) into a fitness app? *

Mark only one oval.

- ◯ Very likely
- ◯ Likely
- ◯ Neutral
- ◯ Unlikely
- ◯ Very unlikely

Usability and Functionality

15. What features do you find essential in a fitness and wellness app? (Select all that apply) *

Check all that apply.

- ☐ Data input (eg: gender, age)
- ☐ Workout recommendation
- ☐ Workout filtering
- ☐ Workout information
- ☐ Workout searching
- ☐ Workout customization
- ☐ Workout library
- ☐ Workout progress tracking report
- ☐ Nutrition intake information
- ☐ Calories/ BMI calculator
- ☐ Other: _____

16. Would you use a feature that recognizes gym equipment and provides guidance on proper usage through AI technology? *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

17. How likely are you to switch the app if an app offers superior AI-powered equipment recognition features? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very likely

18. How important is accurate equipment recognition to your gym experience? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Very important

19. Would you prefer guided workouts like video demonstrations or written instructions in a fitness app? *

Mark only one oval.

- ◯ Yes
- ◯ No

Feedback

20. Any additional comments or suggestions you'd like to provide?

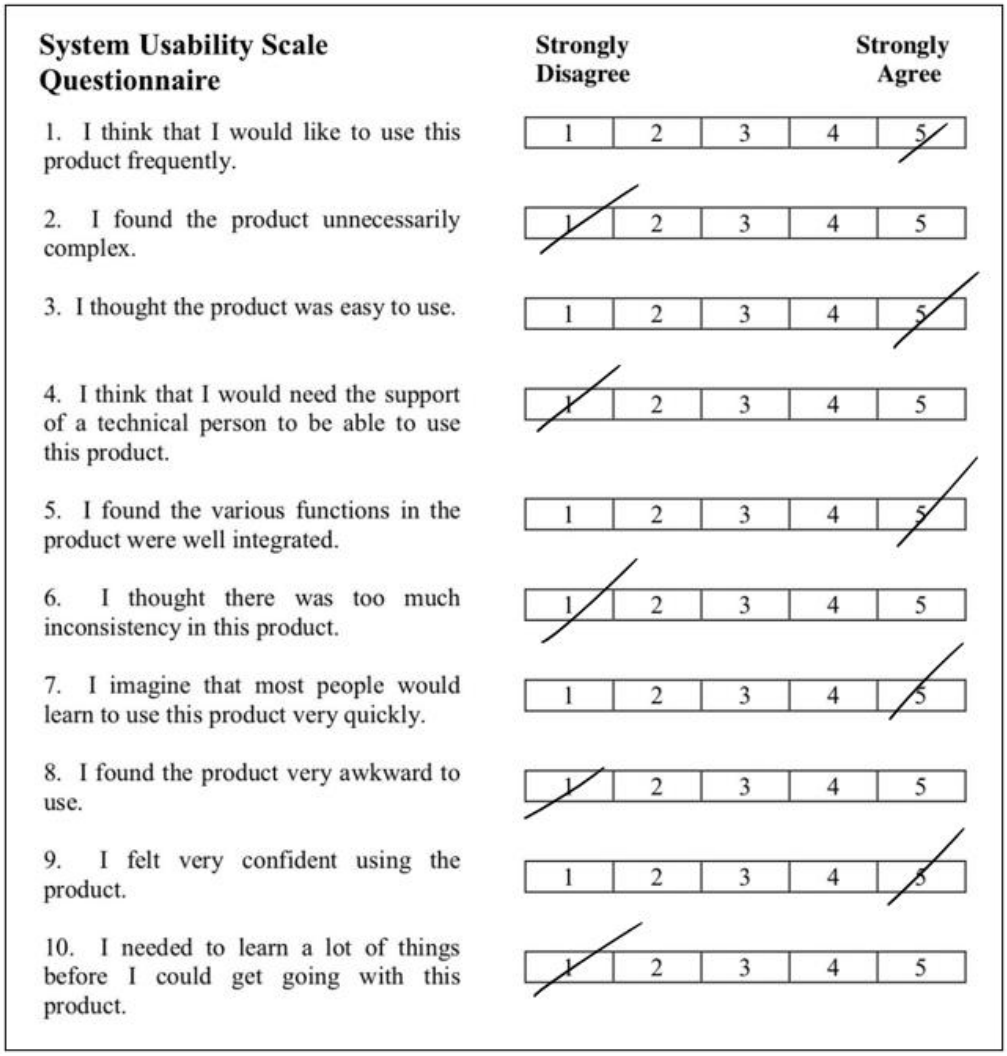
Appendix B: SUS Survey

Mobile SUS – Participant 1:

Name: Leow Hui Wen

Age: 28

Occupation: Office Worker



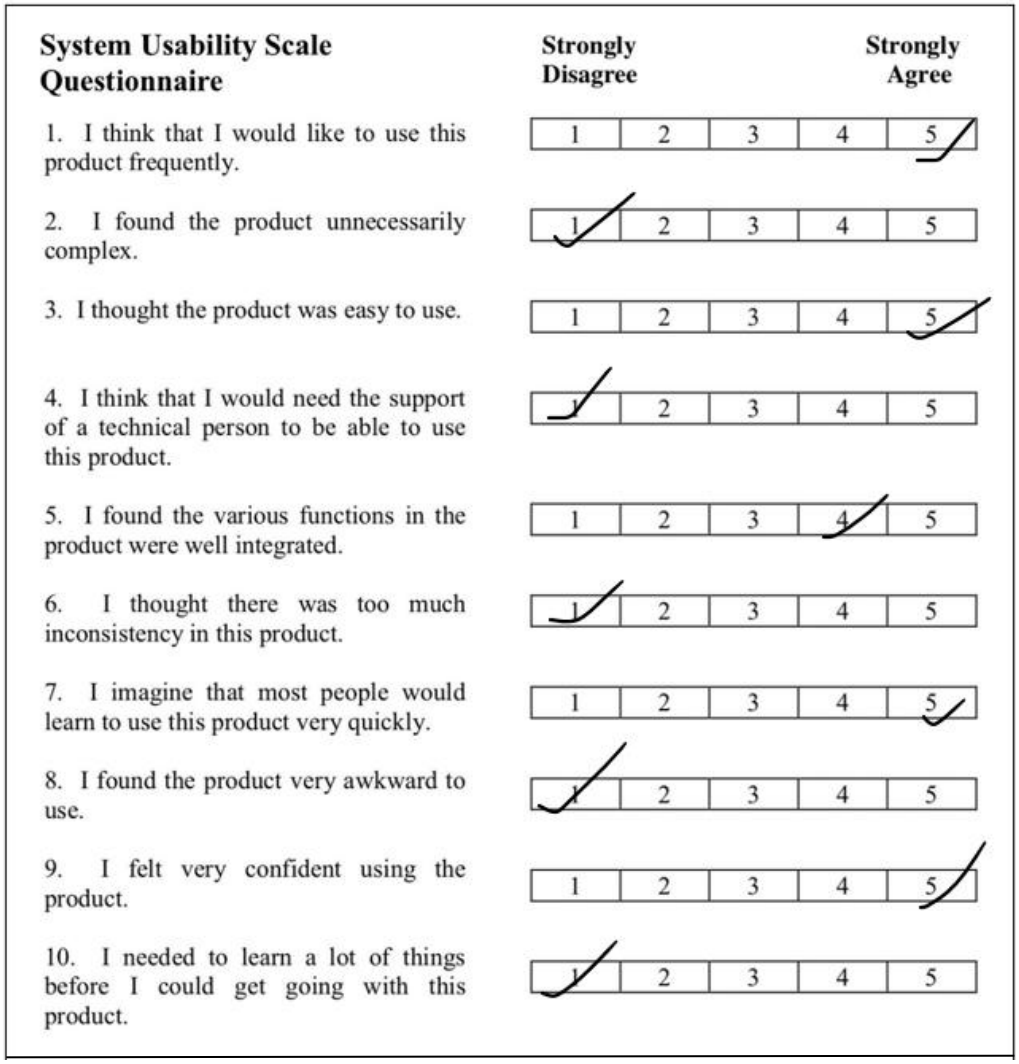
System Usability Scale Questionnaire	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

Mobile SUS – Participant 2:

Name: Chris Lee

Age: 32

Occupation: Fitness Coach



System Usability Scale Questionnaire	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5 ✓
2. I found the product unnecessarily complex.	1 ✓	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5 ✓
4. I think that I would need the support of a technical person to be able to use this product.	1 ✓	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4 ✓	5
6. I thought there was too much inconsistency in this product.	1 ✓	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5 ✓
8. I found the product very awkward to use.	1 ✓	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5 ✓
10. I needed to learn a lot of things before I could get going with this product.	1 ✓	2	3	4	5

Mobile SUS – Participant 3:

Name: Goh Yik Sheng

Age: 26

Occupation: Software Developer

System Usability Scale Questionnaire	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

Mobile SUS – Participant 4:

Name: Lee Yong Khang

Age:36

Occupation: Manager

System Usability Scale Questionnaire	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5 ✓
2. I found the product unnecessarily complex.	1 ✓	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5 ✓
4. I think that I would need the support of a technical person to be able to use this product.	1 ✓	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5 ✓
6. I thought there was too much inconsistency in this product.	1 ✓	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5 ✓
8. I found the product very awkward to use.	1 ✓	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5 ✓
10. I needed to learn a lot of things before I could get going with this product.	1 ✓	2	3	4	5

Mobile SUS – Participant 5:

Name: Tan Ee Han

Age:24

Occupation: Student

System Usability Scale Questionnaire	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4 ✓	5
2. I found the product unnecessarily complex.	1	2 ✓	3	4	5
3. I thought the product was easy to use.	1	2	3	4 ✓	5
4. I think that I would need the support of a technical person to be able to use this product.	1 ✓	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5 ✓
6. I thought there was too much inconsistency in this product.	1 ✓	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4 ✓	5
8. I found the product very awkward to use.	1 ✓	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5 ✓
10. I needed to learn a lot of things before I could get going with this product.	1 ✓	2	3	4	5

Appendix C: UAT Results

Participant 1:

Name: Leow Hui Wen

Test Starting Time: 8.15pm

Test Ending Time: 8.45pm

Mobile-Based Fitness Application				
Test Module	**Test Case ID**	**Test Scenario**	**Status**	**Comment**
Register	UAT-001	1. Able to register a new account	Pass	-
Login	UAT-002	1. Able to login to the newly registered account	Pass	-
Workout	UAT-003	1. Able to view workout library categories	Pass	-
	UAT-004	2. Able to view workout library list	Pass	-
	UAT-005	3. Able to view workout information	Pass	-
	UAT-006	4. Able to apply search filter on workout list	Pass	-
	UAT-007	5. Able to remove search filter on workout list	Pass	-
	UAT-008	6. Able to apply equipment filter on workout list	Pass	-
	UAT-009	7. Able to remove equipment filter on workout list	Pass	-
	UAT-010	8. Able to apply intensity filter on the workout list	Pass	-

	UAT-011	9. Able to remove intensity filter on workout list	Pass	-
	UAT-012	10. Able to apply target area filter on workout list	Pass	-
	UAT-013	11. Able to remove target area filter on workout list	Pass	-
	UAT-014	12. Able to view workout recommendation list	Pass	-
	UAT-015	13. Able to view customized workout list	Pass	-
	UAT-016	14. Able to add customized workout	Pass	-
	UAT-017	15. Able to edit customized workout	Pass	-
	UAT-018	16. Able to delete customized workout	Pass	-
	UAT-019	17. Able to generate workout recommendation	Pass	-
	UAT-020	18. Able to track workout progress	Pass	Consider adding visualization like charts or graph
User Profile	UAT-021	1. Able to edit user profile	Pass	-
	UAT-022	2. Able to view user profile	Pass	-
Calculator	UAT-023	1. Able to calculate BMI and Calorie	Pass	-
AI Gym Equipment Recognition	UAT-024	1. Able to capture image	Pass	-

Participant 2:

Name: Chris Lee

Test Starting Time: 7.02pm

Test Ending Time: 7.36pm

Mobile-Based Fitness Application				
Test Module	**Test Case ID**	**Test Scenario**	**Status**	**Comment**
Register	UAT-001	1. Able to register a new account	Pass	-
Login	UAT-002	1. Able to login to the newly registered account	Pass	Consider including "Remember Me" button
Workout	UAT-003	1. Able to view workout library categories	Pass	-
	UAT-004	2. Able to view workout library list	Pass	-
	UAT-005	3. Able to view workout information	Pass	-
	UAT-006	4. Able to apply search filter on workout list	Pass	-
	UAT-007	5. Able to remove search filter on workout list	Pass	-
	UAT-008	6. Able to apply equipment filter on workout list	Pass	-
	UAT-009	7. Able to remove equipment filter on workout list	Pass	-
	UAT-010	8. Able to apply intensity filter on the workout list	Pass	-

	UAT-011	9. Able to remove intensity filter on workout list	Pass	-
	UAT-012	10. Able to apply target area filter on workout list	Pass	-
	UAT-013	11. Able to remove target area filter on workout list	Pass	-
	UAT-014	12. Able to view workout recommendation list	Pass	-
	UAT-015	13. Able to view customized workout list	Pass	-
	UAT-016	14. Able to add customized workout	Pass	-
	UAT-017	15. Able to edit customized workout	Pass	-
	UAT-018	16. Able to delete customized workout	Pass	-
	UAT-019	17. Able to generate workout recommendation	Pass	-
	UAT-020	18. Able to track workout progress	Pass	-
User Profile	UAT-021	1. Able to edit user profile	Pass	-
	UAT-022	2. Able to view user profile	Pass	-
Calculator	UAT-023	1. Able to calculate BMI and Calorie	Pass	-
AI Gym Equipment Recognition	UAT-024	1. Able to capture image	Pass	-

Participant 3:

Name: Goh Yik Sheng

Test Starting Time: 9.09pm

Test Ending Time: 9.43pm

Mobile-Based Fitness Application				
Test Module	**Test Case ID**	**Test Scenario**	**Status**	**Comment**
Register	UAT-001	1. Able to register a new account	Pass	-
Login	UAT-002	1. Able to login to the newly registered account	Pass	-
Workout	UAT-003	1. Able to view workout library categories	Pass	Should have favorite/ bookmark workout option
	UAT-004	2. Able to view workout library list	Pass	-
	UAT-005	3. Able to view workout information	Pass	-
	UAT-006	4. Able to apply search filter on workout list	Pass	-
	UAT-007	5. Able to remove search filter on workout list	Pass	-
	UAT-008	6. Able to apply equipment filter on workout list	Pass	-
	UAT-009	7. Able to remove equipment filter on workout list	Pass	-
	UAT-010	8. Able to apply intensity filter on the workout list	Pass	-

	UAT-011	9. Able to remove intensity filter on workout list	Pass	-
	UAT-012	10. Able to apply target area filter on workout list	Pass	-
	UAT-013	11. Able to remove target area filter on workout list	Pass	-
	UAT-014	12. Able to view workout recommendation list	Pass	-
	UAT-015	13. Able to view customized workout list	Pass	-
	UAT-016	14. Able to add customized workout	Pass	-
	UAT-017	15. Able to edit customized workout	Pass	-
	UAT-018	16. Able to delete customized workout	Pass	-
	UAT-019	17. Able to generate workout recommendation	Pass	-
	UAT-020	18. Able to track workout progress	Pass	-
User Profile	UAT-021	1. Able to edit user profile	Pass	-
	UAT-022	2. Able to view user profile	Pass	-
Calculator	UAT-023	1. Able to calculate BMI and Calorie	Pass	-
AI Gym Equipment Recognition	UAT-024	1. Able to capture image	Pass	-

Participant 4:

Name: Lee Yong Khang

Test Starting Time: 10.05pm

Test Ending Time: 10.22pm

Mobile-Based Fitness Application				
Test Module	**Test Case ID**	**Test Scenario**	**Status**	**Comment**
Register	UAT-001	1. Able to register a new account	Pass	-
Login	UAT-002	1. Able to login to the newly registered account	Pass	-
Workout	UAT-003	1. Able to view workout library categories	Pass	-
	UAT-004	2. Able to view workout library list	Pass	-
	UAT-005	3. Able to view workout information	Pass	-
	UAT-006	4. Able to apply search filter on workout list	Pass	-
	UAT-007	5. Able to remove search filter on workout list	Pass	-
	UAT-008	6. Able to apply equipment filter on workout list	Pass	-
	UAT-009	7. Able to remove equipment filter on workout list	Pass	-
	UAT-010	8. Able to apply intensity filter on the workout list	Pass	-

	UAT-011	9. Able to remove intensity filter on workout list	Pass	-
	UAT-012	10. Able to apply target area filter on workout list	Pass	-
	UAT-013	11. Able to remove target area filter on workout list	Pass	-
	UAT-014	12. Able to view workout recommendation list	Pass	-
	UAT-015	13. Able to view customized workout list	Pass	-
	UAT-016	14. Able to add customized workout	Pass	-
	UAT-017	15. Able to edit customized workout	Pass	-
	UAT-018	16. Able to delete customized workout	Pass	-
	UAT-019	17. Able to generate workout recommendation	Pass	-
	UAT-020	18. Able to track workout progress	Pass	-
User Profile	UAT-021	1. Able to edit user profile	Pass	-
	UAT-022	2. Able to view user profile	Pass	Can include more detailed fitness metrics
Calculator	UAT-023	1. Able to calculate BMI and Calorie	Pass	-
AI Gym Equipment Recognition	UAT-024	1. Able to capture image	Pass	-

Participant 5:

Name: Tan Ee Han

Test Starting Time: 11.28pm

Test Ending Time: 11.56pm

Mobile-Based Fitness Application				
Test Module	**Test Case ID**	**Test Scenario**	**Status**	**Comment**
Register	UAT-001	1. Able to register a newaccount	Pass	-
Login	UAT-002	1. Able to login to the newly registered account	Pass	-
Workout	UAT-003	1. Able to view workout library categories	Pass	-
	UAT-004	2. Able to view workout library list	Pass	-
	UAT-005	3. Able to view workout information	Pass	-
	UAT-006	4. Able to apply search filter on workout list	Pass	Can implement recent searches section
	UAT-007	5. Able to remove searchfilter on workout list	Pass	-
	UAT-008	6. Able to apply equipment filter on workout list	Pass	-
	UAT-009	7. Able to remove equipment filter on workout list	Pass	-
	UAT-010	8. Able to apply intensity filter on the workout list	Pass	-

	UAT-011	9. Able to remove intensity filter on workout list	Pass	-
	UAT-012	10. Able to apply target area filter on workout list	Pass	-
	UAT-013	11. Able to remove target area filter on workout list	Pass	-
	UAT-014	12. Able to view workout recommendation list	Pass	-
	UAT-015	13. Able to view customized workout list	Pass	-
	UAT-016	14. Able to add customized workout	Pass	-
	UAT-017	15. Able to edit customized workout	Pass	-
	UAT-018	16. Able to delete customized workout	Pass	-
	UAT-019	17. Able to generate workout recommendation	Pass	-
	UAT-020	18. Able to track workout progress	Pass	-
User Profile	UAT-021	1. Able to edit user profile	Pass	-
	UAT-022	2. Able to view user profile	Pass	-
Calculator	UAT-023	1. Able to calculate BMI and Calorie	Pass	-
AI Gym Equipment Recognition	UAT-024	1. Able to capture image	Pass	-