



Proposal Cover Sheet

Subject Code: CSCI321

Subject Name: Project

Submission Type: Online (PDF)

Project Title: Final Project Report

Team: Blu

Student/Team Name: Dani Ternanni, Noor Ahmed, Fatima Tchantchane , Abdullah Rafiei, Ali Mela Ali

Student Number: 7108771, 7084274, 5560251, 6983935, 7368598

Student Phone/Mobile No. 0501373840, 0564758223, 0521969079, 0503195800, 0543388931

Student E-mail: Dt104, na883, fat120, ar251, ama870 @uowmail.edu.au

Lecturer Name: Dr. May El Barachi

Due Date: 13th of December 2023

Date Submitted: 13th of December 2023

PLAGIARISM:

The penalty for deliberate plagiarism is FAILURE in the subject. Plagiarism is cheating by using the written ideas or submitted work of someone else. UOWD has a strong policy against plagiarism.

The University of Wollongong in Dubai also endorses a policy of nondiscriminatory language practice and presentation.

PLEASE NOTE: STUDENTS MUST RETAIN A COPY OF ANY WORK SUBMITTED

DECLARATION:

I/We certify that this is entirely my/our own work, except where I/we have given fully-documented references to the work of others, and that the material contained in this document has not previously been submitted for assessment in any formal course of study. I/we understand the definition and consequences of plagiarism. We/I declare that the project proposal has not been used in any UOWD courses before and that this project idea is a total new idea of the project team

Signature of Student:

Optional Marks:

Comments:

Lecturer Project Proposal Receipt (To be filled in by student and retained by Lecturer upon return of assignment)

Subject:

Project Title:

Student / Team Name:

Student Number:

Due Date:

Date Submitted:

Signature of Student:

Student Project Proposal Receipt (To be filled in and retained by Student upon submission of assignment)

Subject:

Project Title:

Student/Team Name:

Student Number:

Due Date:

Date Submitted:

Signature of Lecturer:



Contents

CERTIFICATION STATEMENT	8
EXECUTIVE SUMMARY	9
ACKNOWLEDGEMENTS	9
Planning and Feasibility	11
Document	11
INTRODUCTION	12
Problem statement.....	13
Background study	13
Our solution	14
PROJECT OBJECTIVES AND SCOPE	15
OBJECTIVE.....	15
GANTT CHART	15
SCOPE	16
COMPETITOR ANALYSIS	17
TARGET NICHE	18
HIGH LEVEL SYSTEM ARCHITECTURE	19
Software Requirements	21
Hardware Requirements.....	22
LIMITATIONS	23
Challenges	23
Future Improvements	24
Current Features Improvements:	24
Near Future Improvements:	26
Far Future Improvements New Modules:	47
Future Improvements OOP:	48
Software Requirement Document.....	51
Introduction.....	53
Intended Audience and Reading Suggestions	53
Product Scope	54
Overall Description.....	55
PRODUCT PERSPECTIVE	55
Product Functions.....	55
User Classes and Characteristics	56
OPERATING ENVIRONMENT	59

Design and Implementation Constraints.....	60
EXTERNAL INTERFACE REQUIREMENTS	60
USER INTERFACE.....	60
COMMUNICATIONS INTERFACES.....	60
SOFTWARE INTERFACES	61
DOMAIN MODEL	62
System Features (Use Cases).....	62
List of VirtuFit System features:	63
Update Account.....	66
Nonfunctional Requirements	78
Performance Requirements	78
Usability Requirements	79
Reliability	79
Safety Requirements	79
Security Requirements.....	80
Design Document	81
Introduction.....	83
Purpose	83
Scope	83
Overview.....	83
System Overview.....	84
System Architecture	85
Architectural design	85
Physical Architecture.....	86
Decomposition description	87
UML Class diagram (OOP) :	88
Design rationale.....	88
Considering 2-Tier architecture vs 1 or 3-Tier	88
Network architecture	88
Data design	89
ERD diagram	89
Data description.....	89
Data dictionary	90
Component Design.....	92
Activity diagram	92

Sequence diagram.....	102
Human Interface Design.....	114
Overview of user interface	114
Screen Images.....	114
Requirement Matrix.....	125
UML class diagram	126
Test Case Document.....	127
Introduction:	129
Test Plan:	129
Test plan Verification Validation (V & V):.....	129
Test plan Schedule:.....	130
Test plan Requirements Traceability:.....	131
Test plan Test recording procedures:.....	133
Test plan Hardware and Software:	133
Test plan constraints:	134
Unit Testing:.....	135
Login Module:	135
Control Flow Graph (CFG) for Login:	138
Chat bot Module:.....	140
Form Monitoring Module:	143
Acceptance Testing:.....	158
Sign up for VirtuFit	158
Log in to VirtuFit.....	159
Converse with virtual trainer (chatbot).....	160
Create schedule and workout plan.....	162
Start Workout Session	163
View and Edit Schedule	163
Monitor Form	164
Application Screens:.....	178
Login.....	178
Register	179
Choose Trainer	180
Beginning Chatbot	181
Home Page.....	186
Profile Page	187



Side Bar	188
Workouts Description	189
Workout Schedule Page	203
Start Workout Page.....	206

Guidelines

VirtuFit

This technical document is prepared and authorized by team Blu as deliverable of University of Wollongong in Dubai course CSIT321 graduation project.

Team Blu

Fatima Aya Tchantchane 5560251,

Noor Ahmed 7084274,

Dani Ternanni 7108771,

Ali Mela Ali 7368598,

Abdullah Rafiei 6983935

Autumn 2023 Batch

CERTIFICATION STATEMENT

We, **Team Blu**, certify that this paper is the result of our independent effort for the CSIT321 project. We assert that none of the material, ideas, or concepts come from other sources, and those that do are duly acknowledged. This document hasn't been turned in anywhere else. Any outside resources utilized are properly referenced. We affirm the document's conformity to integrity standards and accept academic integrity policies. We attest to our comprehension and compliance by signing below.

Fatima Aya Tchantchane

Noor Ahmed

Dani Ternanni

Ali Mela Ali

Abdullah Rafiei

EXECUTIVE SUMMARY

This document, written by Team Blu, offers comprehensive details on the features and operation of the VirtuFit application. This paper makes use of graphical illustrations, analogies, and thorough explanations of the processes that went into creating the program.

With the help of the VirtuFit applications, people may reach their fitness objectives while lounging in the comfort of their own homes. With the use of this digital platform, users will be able to receive personalized training plans that are tailored to their objectives and schedules. Furthermore, guarantee the security of its users by offering form monitoring capabilities while the user exercises. VirtuFit is a future strategy that encourages people to put their physical health first despite the fast-paced lifestyles they lead.

ACKNOWLEDGEMENTS

Team Blu would like to personally express our thanks to Dr. Mai El Barachi for providing us with this chance to test our abilities and create an application that attempts to address and improve the current problems facing the globe.

Furthermore, we would like to express our sincere gratitude to Professor Farhad Oroumchain and Dr. Patrick Mukala for their valuable direction and assistance in using optimized pathways and technologies to help us accomplish our aim throughout the project.

Finally, we would like to express our gratitude to Bishir Zamout, our WISP mentor, for his astute advice and support during the project, which enabled us to surpass our expectations and achieve outstanding outcomes.

Planning and Feasibility Document

Planning and Feasibility
Document



CSCI321
Team Blu

Dani Ternanni, Fatima TchanTchane, Noor
Ahmed, Ali Mela Ali, Abdullah Rafiei

INTRODUCTION

Problem statement

Physical wellbeing and fitness are important aspects for every human being; they have proven to improve brain activity, decrease the probability of diseases, and provide strength to muscles and bones, improving quality of life. Yet, despite the ubiquity and commonality of gyms and fitness awareness, many people continue to confront challenges with availability and access to achieving their fitness goals.

Background study

After thorough analysis, we determined that the highest rated obstacle faced by individuals while going to the gym is lack of time. Around 42% of people reported in a survey that their biggest setback for incompetence at their physical wellbeing is lack of time in their daily routine. Whereas, ranking up to 27%, the second most common reason for individuals aged 18–32, was lack of confidence; statistics reveal that nearly fifty percent of Americans feel reluctant to work out in front of other people. Concluding, 27% of adults worldwide fail to get sufficient physical activity.

Moreover, another significant challenge in conservative parts of the world is the lack of women-only gyms, or the schedule of women-only hours being inconvenient for people. Due to the following, people fail to recognize the necessity of monitored gymnastics, resulting in bad posture while exercising at home and a lack of awareness about what regimen is appropriate for them. Extensive studies have shown that poor posture can lead to more problems than good, such as further muscle and joint damage, spinal dysfunction, and back pain.

Our solution

Team Blu offers VirtuFit, a cutting-edge solution aimed at addressing the issues that individuals in the twenty-first century face. With innovative technology, VirtuFit is an android application that lets users experience a gym at their own pace and convenience. There are three primary components to the application: AI-integrated chatbot that replaces the conventional method of using surveys to collect user data; individually tailored exercise regimens for each user based on their objectives and circumstances; and, finally, a form monitoring function that guarantees user safety.

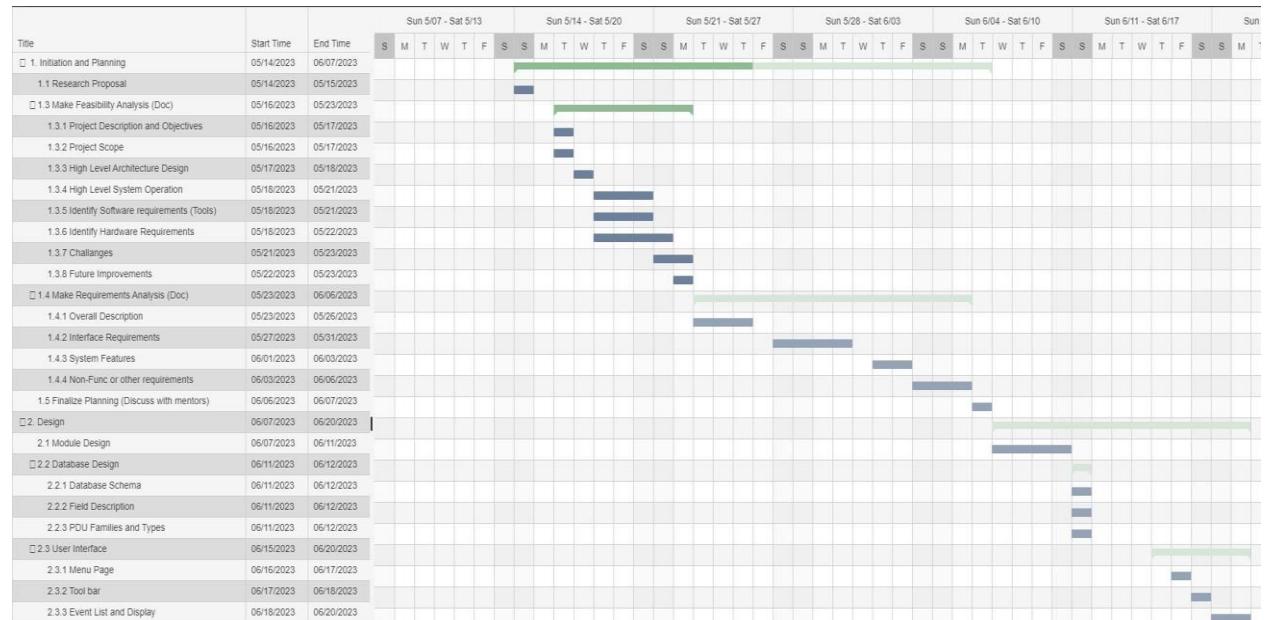
PROJECT OBJECTIVES AND SCOPE

OBJECTIVE

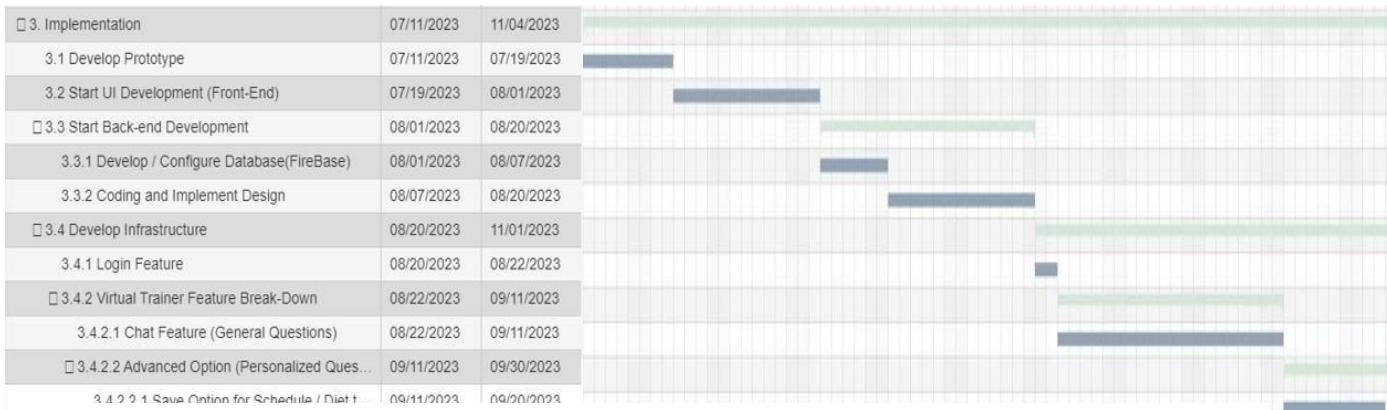
Team Blu noticed a gap between consumers and fitness apps after completing an extensive study, which was a lack of progress analysis and monitoring as they exercised. As a result, Blu decided to address this issue by developing an app that provides individuals with all the benefits of a gym without requiring numerous programs. Our fitness app aims to provide individuals with personalized fitness plans, which include conversations with a virtual trainer that will provide them with a workout regime. As the consumer implements their regime, their posture and movements will be monitored thoroughly to achieve concentrated results and avoid unsupervised routines causing health issues. The application will employ React Native for front-end development as it provides improved efficiency and development speed across platforms. Users can use the application to its maximum potential by using a camera with at least 8 megapixels or higher to identify their movements while exercising. The major goal of this program is to keep users from having too many applications and to create a one-stop shop for all their needs. Ensuring minimum user input while providing maximum user satisfaction.

GANTT CHART

Semester 1:



Spring 2023



Summer 2023

41	3.4.2.2.2 Advice Changes and Recommen...	09/20/2023	09/30/2023	
42	3.4.2.3 Analyze Exercise and calculate	09/30/2023	10/20/2023	
43	3.4.3 Evaluate Results calculate Progression L...	10/20/2023	10/31/2023	
44	3.4.4 Motivate User	10/31/2023	11/04/2023	
45	□ 4. Testing	11/04/2023	11/09/2023	
46	4.1 Develop Test Cases	11/04/2023	11/09/2023	
49	5. Loop between Implementation and Testing	11/09/2023	12/07/2023	

Autumn 2023

SCOPE

PURPOSE

The goal of the fitness app is to offer users individualized fitness regimens, interactions with virtual trainers, in-depth workout tracking, recommendations, and personalization choices. It aims to enable users to accomplish their exercise goals successfully and safely. We strive to support users through their fitness journey by providing customized training regimens and engaging virtual trainer sessions, ensuring correct form and technique to prevent health risks. The Blu application will also offer tailored suggestions and modifications to improve users' outcomes through ongoing monitoring and analysis of their progress.

Additionally, we provide customization choices for future goals and a welcoming community to build user motivation and pleasure. Blu's goal is to encourage and equip people to lead healthy lives.

COMPETITOR ANALYSIS

Feature/Company	Our product	FitBod	Muscle booster	Freeletics	Strong
Strengths	Form monitoring and guidance	Recovery heatmap	User friendly, Suitable for all skill levels	Offers a great variety of exercises	Cheap, balanced for all skill levels, easy to use
Weaknesses	Does not offer diet plans	Minimal guidance, impractical for beginners	Limited guidance	Limited for advanced gym goers	Limited workouts and guidance
Pricing (dollars)	9.99/month	15.49/month	5.99-19.99/month	14.99-20.00/month	7.49/month
Onboarding Experience	Conversate with user to gather info	Requires user to answer a questionnaire	Requires user to answer a questionnaire	Requires user to answer a questionnaire	Simple, minimal steps required

TARGET NICHE

Our product targets people of all ages who might need the following:

- Getting in shape but needing motivation.
- A new workout routine.
- Assistance and guidance about work outs.
- A flexible workout routine due to a busy schedule.
- Workout plans that caters to their health conditions.
- An advanced fitness app integrated with modern technologies.
- Individuals uncomfortable working out in public.

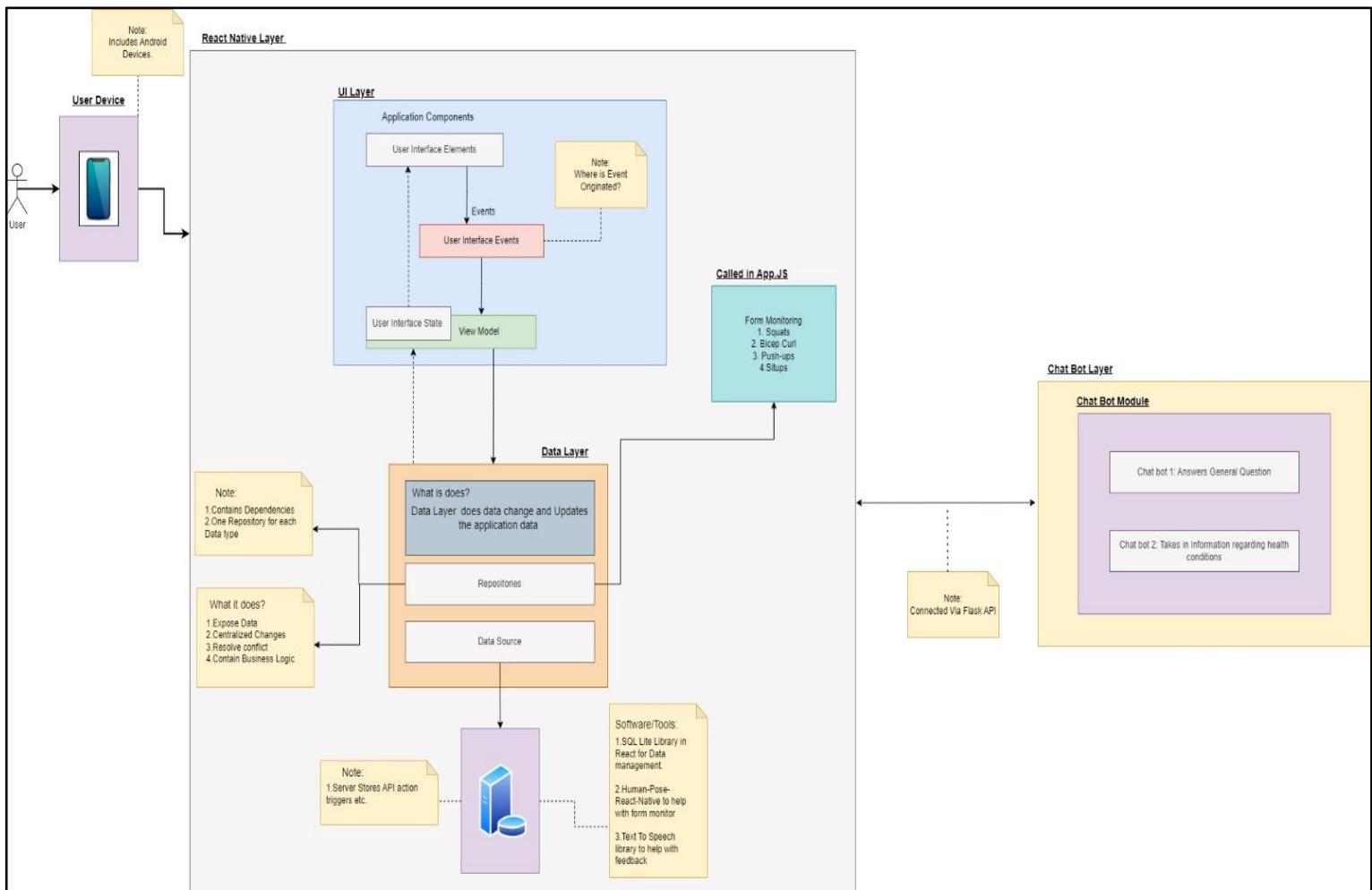
Our main goal with this application is to accommodate users with different needs and circumstances, all at one place. With the fast-paced lifestyle being adapted all over the world it is significantly important to keep in mind that the needs of people all differ greatly despite having the same solution. Regardless of circumstances, every individual aims to enhance their physical health at a certain point of their life, whether it may be due to their health conditions, interest, or body standards invested by their surroundings. Our application focuses to provide them with the utmost benefit of their investment into the application, by personalizing every individual's experience and providing them with their desired results.

Below is a persona for a 9-to-5 marketing intern named Dila, who faces personal and social conflicts while wanting to maintain her fitness goals. Dila is one example of many cases that led us to the solution of developing the VirtuFit app.

HIGH LEVEL SYSTEM ARCHITECTURE

It can be seen in the high-level system architecture that the UI (front-end) layers consist of application components and view model components; These components are mostly done in the design and implementation phases of the project. On the other hand, repositories and data sources are components of the data layer (back end). They work together to retrieve the data changes required and update the application view model data.

Although most modules are inside react-native layer, the chatbot layer is a backend back-end layer that is connected to the front-end via Flask API server.



List of key front-end components:

- Client-side: This is the location where processes are run. All application execution strategies are made on the end-user device.
- Design: done on React Native; all UI elements and UI features are made here.

List of key back-end components:

- Local Database: All data from chat bot, form monitoring, and UI components are stored on the local device.
- API Layer: This layer should send and receive data via an interface.
- Flask API: Connects Chat bot to Front-End.

How it would work General Question:

- The user opens the mobile application and views the UI view model, which is on the client-side.
- Users would ask general questions about the Virtu Fit app.
- The question gets processed through Natural language processing (NLP).
- The processed question gets sent to the required API to retrieve the necessary answers.

How it would work Personal Question (Based on Feature):

- The user opens the mobile application and views the UI view model, which is on the client-side.
- Users would then ask the Virtu Fit app to create a schedule for a specific goal. Example: "I would like to lose 10 kg."
- The virtual trainer would then ask the user to choose home or gym workouts.
- Then Virtual trainer would require the users free or available times they would like to go to the gym,

- Based on these two inputs, the virtual trainer would create a schedule to satisfy the user's needs.

Software Requirements

Tools and platforms

Platforms:

- **React Native (front-end)**

React Native is an application that will be used to build our front-end and design what our application will look like. It works on IOS and Android by automatically transforming the same code built using JavaScript into a framework that will allow it to actively make applications on two different platforms. We will use it in our application to make it a cross-platform application as well as to build all the components of our UI layer.

- **SQLite**

SQLite is a serverless open-source relational database management system providing with local storage solutions supported by SQL syntax.

Tools:

- **Flask**

Flask is a web tool that makes request handling and API management much simpler. It can act as a middleware between mobile apps and AI services. We will use Flask to ease the back-end development process by receiving data in Flask from the mobile application and processing the data using trained AI models to generate the desired output. Once we have our desired output, the new data is sent back to the mobile application and is ready for use and display.

- **Open AI API**

The Open AI API can be used to build and refine the perfect chatbot. It is also used to define what we expect the user to ask. Once the range of questions that the user gets to ask is clearly defined. The API will be used to send user entries to the chatbot, and an answer will be

provided. In our case, we will limit the user to only ask questions related to exercise and fitness.

Libraries:

- **React-Native-Human-Pose**

Used to check user movement and once it detects user will apply a model which displays 17 joints on the user. These joints each have different X, Y, Confidence values. This information can be used to understand what the user is doing.

- **React-Native-TTS**

A text to Speech library which uses the phone's speaker to speak out texts.

- **NLP**

Understand Users' generalized questions. Breakdown questions which allow the VirtuFitness assistant to make easier searches and get the correct answer for the question.

Hardware Requirements

System setup hardware

- Android Mobile Phone of 3-4 GB RAM
- Camera with 12 megapixel and above
- Server
- CPU with enough processing power

Specialized hardware

- Phone stand

The Specialized tool is required by users so that they have access to special features such as calculate calorie burn or record and upload their form and get feedback from VirtuFit Assistant.

LIMITATIONS

As our project entered the development phase, team Blu encountered some restrictions and difficulties related to the technologies, libraries, and APIs that were being used. The team Blu encountered a more significant obstacle in the shape of incompatibility; although our initial idea was to create a cross-platform application, we were only able to accomplish their objective with an Android application. In addition, team Blu had to switch between several libraries, including the form monitoring library, to adjust to react's coordination because of the usage of react native and its incompatibility with preferred libraries.

Lastly, the application's benefits are confined to the user's own will. Whereas the program will aid the user's desire for physical fitness, it will not be completely trustworthy in achieving the user's goal because the application can only monitor the user while they use it. Anything beyond the application's recommendation is the user's responsibility to commit to consistency.

Challenges

1. Pose Detection Form Monitoring:

A better AI pose detection model can be used which is more accurate. But choosing the right model based on the specs of user device and making sure it is still accurate it going to be a difficult task.

2. Chatbot:

The initial aim was to create a single chatbot that would ask users for information and offer advice on common questions. However, during the initial testing phase, it became clear that using just one chatbot would not be sufficient, consequently leading us to built two. The first chatbot's objective is restricted to new users at their initial sign-in and is to gather user data to generate a fitness plan based on the data the user enters. To respond to general inquiries, the second chatbot will remain constant during their VirtuFit experience.

3. Switch from SQLite to Firebase:

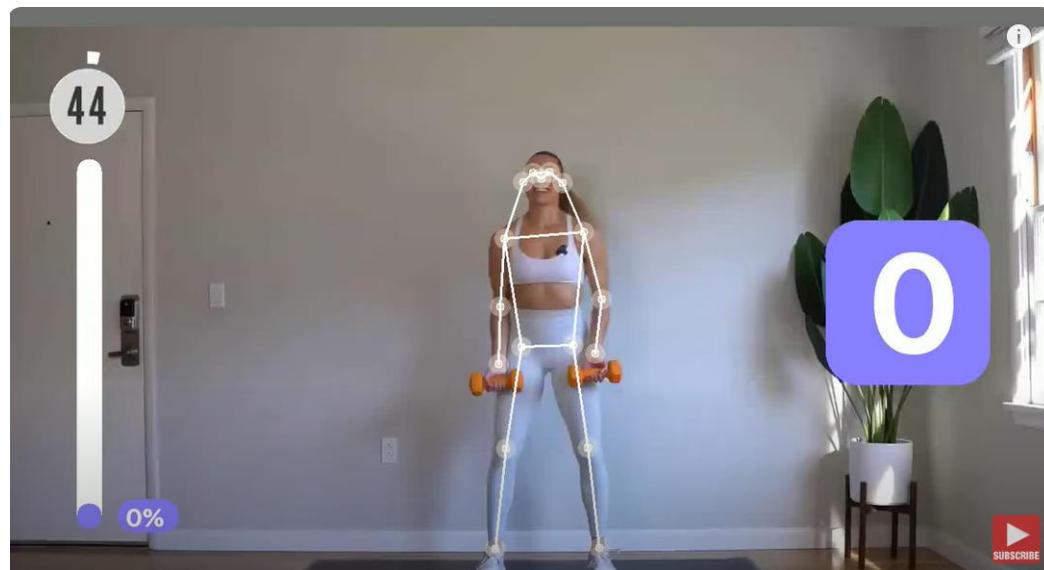
Similarly, during the implementation phase team Blu decided to opt for a local database management system as opposed to the initial decision of using Firebase. This decision was based upon efficiency and resource availability.

Future Improvements

Current Features Improvements:

Form Monitor:

- Progress Loading Bar for REP-
A loading bar will gradually increase as the user approaches the completion of a repetition. Once the satisfactory angle is reached, the progress bar will reach its climax at 100%. This feature enables users to visually comprehend their status in the exercise without depending on text-to-speech feedback from the trainer.



- Form Status confirmation-
A tick arrow can visually show the user if they are in the correct position and distance if not then a cross arrow will be displayed.
- Turning On/Off Text to Speech -
Having a button which can help user turn on/off feedback from TTS when needed.



- Use of outside Form monitoring tools-
Having the device connect to a server for form monitoring can improve accuracy because it reduces resource load from the device and because better Pose Detection can be used for example: BlazePose model which has 33 joints and can detect more than one user on the screen.

Chatbot:

- Use of Text To speech on chatbot-
Users don't have to read the response from the chat, instead a text to speech would read it out for the user.
- Sending question to chat bot using voice-

Whenever the user communicates with the chatbot for any inquiries they can use their voice instead of typing in text.

Workout Page:

- Clear Display-

In the workout page which shows all available workouts in the application there needs to be a clear explanation of these exercises in bullet points so that they are easier to understand.

Database

- Use of Database server:

Switching to Firebase or any other data server hosting application will be more useful in the future for the app in long term. This is because, it allows users to transfer data onto new devices and not lose progress.

Near Future Improvements:

Forgot password

Forgot password	
Function Name	Recover account through forgot password.
Description / Goal	Users should be able to change their passwords if they forgot their login password.
Inputs	<ul style="list-style-type: none"> • User Email/Username • Method of verifying user either via phone number or Email or answer personal questions.
Source	User should press forgot password button

Outputs	New tab is opened, and users can change their password.
Destination	User account module in database affected.
Action / Steps	<ul style="list-style-type: none"> • Step 1: Users will click on the forgot account password button and a new tab is opened. • Step 2: The user must choose an option on how to recover their account. Options could be through (Emails, Phone number, Personal question) • Step 3: After the user manages to bypass the security option a new link is sent which allows the user to change their password.
Requires	User must be able to bypass the recover account security option
Pre-condition	<ul style="list-style-type: none"> • Account must exist • Users must be able to answer or verify their identity through recovery options
Post-condition	The user's password for their account is updated. Users can now login to account using the updated information.
Side-effects	N/A

Exceptional Scenario	<ul style="list-style-type: none"> User doesn't know the answer to any recovery option then they have to communicate with an admin or tech support.
-----------------------------	--

Display Schedule progress

Display Schedule Progress	
Function Name	Display schedule progress
Description / Goal	User can monitor their progress along the assigned schedule
Inputs	Button touch from user
Source	User selection
Outputs	Schedule is displayed for the user
Destination	Schedule layer / frame

Action / Steps	<ul style="list-style-type: none"> • Step 1: User opens the app on the mobile device. • Step 2: User logs into the application • Step 3: User selects schedule option • Step 4: User views their schedule
Requires	User is required to have an ongoing workout plan assigned
Pre-condition	<ul style="list-style-type: none"> • User must have an existing account • User must have a valid membership • User must be logged in
Post-condition	Schedule is viewed for the user
Side-effects	The page viewed by the user will be changed to the schedule page.
Exceptional Scenario	If the user does not have a workout plan yet, if the schedule page is viewed empty with a message displayed saying there is no schedule created yet.

Display calorie loss / gain progress

Calories loss / gain progress

Description / Goal	User is able to monitor their calorie loss / gain progress
Inputs	Button touch by user
Source	User selection
Outputs	Calorie loss / gain progress is shown to user
Destination	Calorie loss / gain progress layer / frame
Action / Steps	<ul style="list-style-type: none"> • Step 1: User opens the app on the mobile device. • Step 2: User logs into their account • Step 3: User selects calories option • Step 4: User views their calorie loss /gain progress
Requires	User is required to have a ongoing workout plan assigned
Pre-condition	<ul style="list-style-type: none"> • User must have an existing account • User must have a valid membership • User must be logged in
Post-condition	Calorie loss / gain progress is viewed for the user

Side-effects	The page viewed by the user will be changed to the calorie page
Exceptional Scenario	If the user does not have a workout plan yet, if the calorie page is viewed empty with a message displayed saying there is no schedule created yet.

Display duration of workout remaining

Display timer of workout	
Function Name	Provide the user with a timer showing the remaining duration of the workout
Description / Goal	Users will be able start their workouts and time themselves and the duration of their workouts.
Inputs	Users are required to click the start workout button.
Source	User clicks on start workout.
Outputs	Workout timer is on and displayed on the phone screen. The timer starts counting the duration of the workout.

Destination	Timer layer is displayed as a widget.
Action / Steps	<ul style="list-style-type: none"> Step 1: Users will click on their schedule tab and a screen will display the workouts for this week. Step 2: Users will click on today's workout and choose to start the workout button. Once this is done a timer is displayed and counting of timer is started,
Requires	Users are required to have a workout plan in schedule or create one to start the workout timer.
Pre-condition	<ul style="list-style-type: none"> User account must exist. User must be logged in. User must have a valid membership. User must have a workout plan at that time, or he should add a new workout timing and add it to the schedule.
Post-condition	<ul style="list-style-type: none"> Timer is on and counting has started. User can pause the timer and take any breaks.
Side-effects	User is doing a workout therefore the streak is incremented.
Exceptional Scenario	<ul style="list-style-type: none"> If the user has no plan now, we would not be able to start the timer. Therefore, a plan must be made on the spot then he/she can start the workout.

Display Motivational Quote

Motivational quotes	
Function Name	Display motivational quotes
Description / Goal	To motivate users by providing them with daily quotes.
Inputs	Famous motivational quotes
Source	Online platforms
Outputs	Motivational quotes displayed on application
Destination	Pop up message on application / notification
Action / Steps	<ul style="list-style-type: none"> • Step 1: Application receives quote from online quote generator or from the saved quotes. • Step 2: Application displays a quote to users to help with motivation.
Requires	The user must allow the display of motivational quotes.

Pre-condition	<ul style="list-style-type: none"> • User accounts must exist. • User must be logged in to the account. • User must have a valid membership • User must have motivational quote notification on.
Post-condition	<ul style="list-style-type: none"> • Receive daily quote from VirtuFit app.
Side-effects	<ul style="list-style-type: none"> • The Notification module will be activated.
Exceptional Scenario	<ul style="list-style-type: none"> • Users can have the quote turned off making the feature inactive.

Provide Workout Challenges

Workout Challenge	
Function Name	Challenge users.
Description / Goal	Provide new challenges to users to push them further if they think the workout was very easy.
Inputs	User confirms the addition of challenges in his/her workout.
Source	User confirmation to challenges.

Outputs	User workout duration is increased. Quote from virtuFit commending and praising the user for accepting challenges.
Destination	Schedule module is changed in the system, and a new quote is used for the user.
Action / Steps	<ul style="list-style-type: none"> ● Step 1: Users will confirm the addition of challenges after finishing their workout by clicking on the add challenge button. ● Step 2: a screen displays duration of the challenge phase
Requires	<ul style="list-style-type: none"> ● User confirmation is required after initial workout is completed.
Pre-condition	<ul style="list-style-type: none"> ● Users must have an account. ● Users must be logged in ● Users must have finished the initial workout only then can a bonus challenge workout be added.
Post-condition	<ul style="list-style-type: none"> ● New challenge is added, and a workout screen is displayed waiting for the user to start the challenge phase.
Side-effects	N/A
Exceptional Scenario	N/A

Provide affirmation to motivate users

Provide affirmation to users	
Function Name	Enquire user about the workout
Description / Goal	Provide motivational words and positive affirmations to keep users focused on the workout.
Inputs	Work out details such as time.
Source	Schedule information provided by database.
Outputs	User is told positive words and small tasks such as how much time is left in the workout is mentioned by the VirtuFit trainer keeping the user focused on the set.
Destination	Schedule module is giving information to the virtuFit trainer so that the bot can make appropriate comments during the workout such as (Time to change exercise and keep up the good work, almost there just a few more reps).
Action / Steps	<ul style="list-style-type: none"> • Step 1: User must allow the VirtuFit trainer to make affirmations to the user. • Step 2: VirtuFit trainer must be able to provide user affirmations based on the workout.

Requires	Requires the user to allow the trainer to give positive affirmation.
Pre-condition	<ul style="list-style-type: none"> • User's account must exist. • The user must be logged in. • The user must allow the bot to make affirmations.
Post-condition	Users are informed by the trainer about how much of the workout is completed and provide user comments such as ("good job", "almost there") to keep them focused in the workout.
Side-effects	<ul style="list-style-type: none"> • Users might get ahead of the schedule and finish the workout quickly.
Exceptional Scenario	<ul style="list-style-type: none"> • If the user turns off the affirmation option, then the comments cannot be made by the trainer.

Display Workout Streak

Streak workout	
Function Name	Count workout streak

Description / Goal	Count the number of times the user consistently follows the schedule and workout plan and display the number
Inputs	The user clicks on the start workout button
Source	User starts the workout
Outputs	Counter is added to the streak and the streak is always displayed
Destination	The UI display workout counter is affected.
Action / Steps	<ul style="list-style-type: none"> • Step 1: User starts a workout plan normally. • Step 2: then the streak counter is added until after the user clicks start workout and finishes the workout
Requires	User to start and finish the workout
Pre-condition	<ul style="list-style-type: none"> • User accounts must exist. • User must be logged in. • Users must start the workout and complete all exercises

Post-condition	Streak counter is increased, and users can view how consistent they are with the schedule.
Side-effects	UI of the schedule is affected because of streak UI changing constantly and counter is added every time.
Exceptional Scenario	Users can cancel a workout beforehand, not affecting the streak.

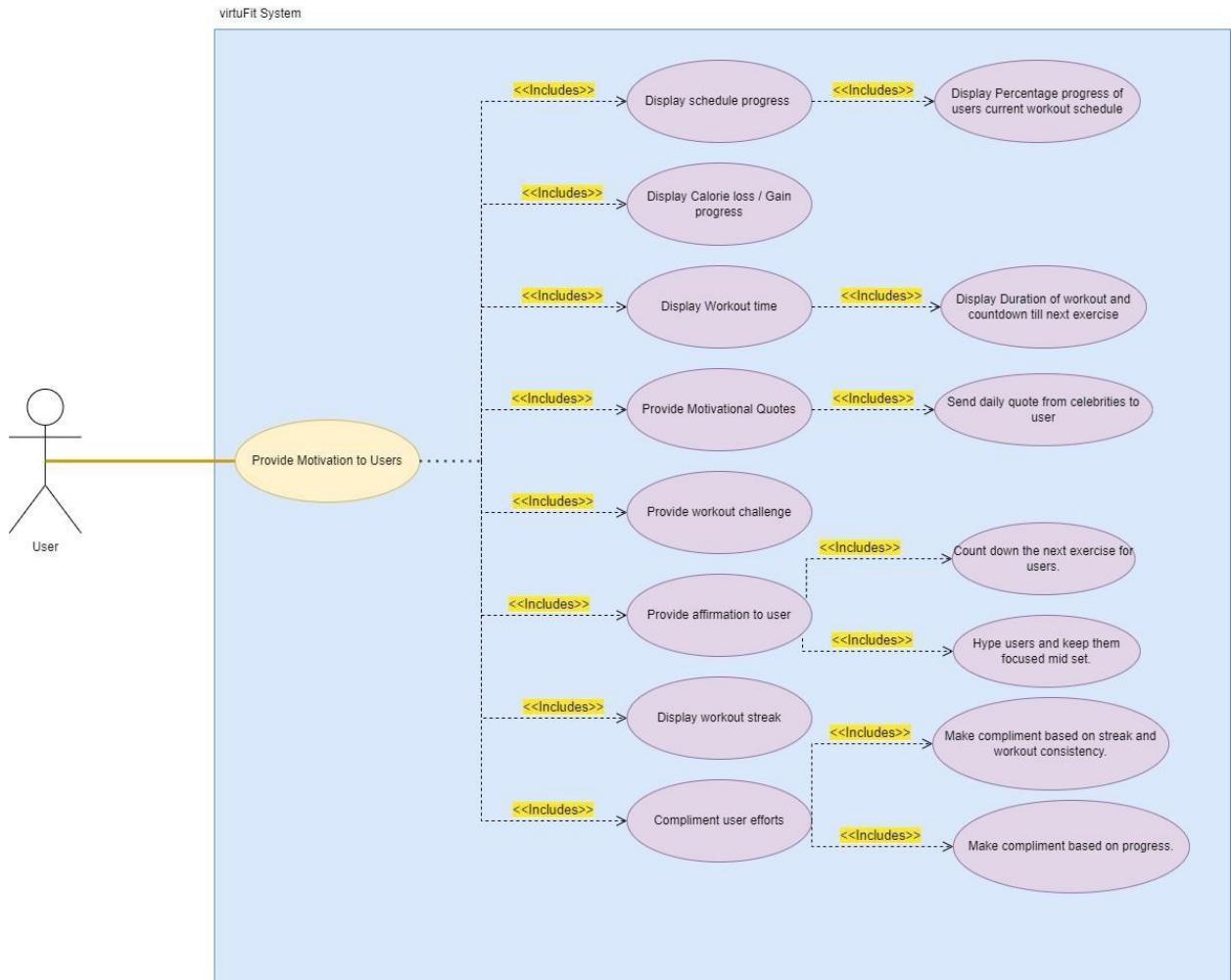
Comments by trainer Recognize small effort and make comment to user

Compliment user efforts	
Function Name	Positive compliment and comments to user
Description / Goal	Recognize efforts of users such as consistently following the plan or users taking on challenges and making positive comments about them.
Inputs	User schedule information like streak or number of challenges completed.

Source	User schedule data.
Outputs	Trainer makes conversation and comments about the user.
Destination	Conversation module and user schedule data is affected.
Action / Steps	<ul style="list-style-type: none"> ● Step 1: VirtuFit trainer takes profile information such as how many streaks the user has and how many challenges have been completed. ● Step 2: the trainer makes comments and compliments based on users' achievements.
Requires	Users must already have some profile information or achievement.
Pre-condition	<ul style="list-style-type: none"> ● User's account must exist ● Users must be logged in ● Users must have some achievement or streak.
Post-condition	Conversation is made with users and compliments are mentioned
Side-effects	More conversation options with users

Exceptional Scenario	N/A
-----------------------------	-----

Provide motivation use case overview:



Provide advice and recommendations

Advice users	
Function Name	Provide alternative solutions to meet user goals.
Description / Goal	Users should be able to mention any displeasure with the schedule or exercises provided by the VirtuFit app. Alternative recommendations can be suggested.
Inputs	User mentions the issue in the workout to the VirtuFit trainer.
Source	Through conversation or the schedule users can enquire advice or recommendations about workout.
Outputs	If the new advice or change is accepted will implement the change to the workout.
Destination	Conversation layer is affected, and the schedule frame.
Action / Steps	<ul style="list-style-type: none"> ● Step 1: Users can ask for advice or mention the issues they face during the workout. ● Step 2: VirtuFit app will provide new solutions and recommendations which the user can accept or reject. ● Step 3: accepted recommendations are implemented in the workout.

Requires	Users must accept or reject the advice or recommendation so that it is either implemented or rejected.
Pre-condition	<ul style="list-style-type: none"> • User's account must exist. • User's must be logged in. • User's must already have cleared basic communication with a virtual trainer.
Post-condition	New advice is implemented to workout.
Side-effects	If advice or recommendation is related to the schedule any changes must be changed automatically.
Exceptional Scenario	Users can be unsatisfied with all the advice and recommendations only then will a manual workout be allowed for the user.

Payment

Payment	
Function Name	Pay membership

Description / Goal	User can pay for their membership
Inputs	User payment details
Source	User
Outputs	Payment transcript or bill
Destination	UI (Payment page), Database
Action / Steps	<ul style="list-style-type: none"> • Step 1: User logs into the application • Step 2: Select membership option. • Step 3: If user has a valid membership still, they can select extend option. • Step 4: If user has an invalid membership, they can select join option. • Step 5: User then enters their payment details. • Step 6: User confirms the transaction
Requires	User to have a valid online payment method such as visa or PayPal
Pre-condition	User must have an existing account. User must be logged in

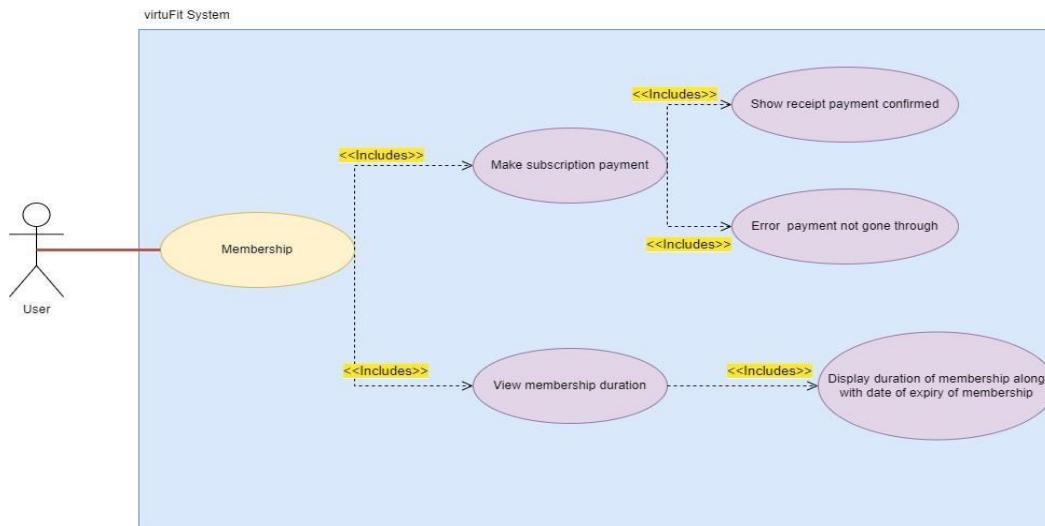
Post-condition	User has a valid membership
Side-effects	Database updates user information
Exceptional Scenario	In case of a failed transaction, the application will notify the user

View Current Membership

View Membership period	
Function Name	Display membership period
Description / Goal	Display information such as which date membership expires and how many days left till expiration.
Inputs	Once payment is confirmed it will start the countdown till expiry.
Source	User confirms payment to VirtuFit application.
Outputs	Duration of membership and date of expiry is displayed for users.
Destination	Payment module/database is used to start the countdown.

Action / Steps	<ul style="list-style-type: none"> Step 1: User will confirm payment in the application. Users will sign up for an account and login to their respective account. In the settings membership tab, there is a feature that displays membership duration and expiry date.
Requires	User is required to confirm the payment.
Pre-condition	<ul style="list-style-type: none"> User must pay for account. User must sign up for account.
	User then logs in to the account.
Post-condition	Displays users' membership information and countdown till expiry is started.
Side-effects	Countdown module is initiated after payment is done.
Exceptional Scenario	N/A

Membership use case overview:



Far Future Improvements New Modules:

It is crucial to come up with future improvements that can be integrated into the project, which will help in enhancing and advancing the project. In addition, it will also make the project have sufficient requirements. Below are some of the improvements we can work on after completing the application.

1. Diet Planning:

Incorporating personalized diet plans based on users' fitness goals, dietary preferences and their physical characteristics. This feature will aid users in picking the righteous dietary and healthy meals on daily basis.

2. Virtual Reality (VR) or Augmented Reality (AR) Integration:

The integration of VR and AR improves user experience and provides real-time visualizations in the fitness app, to ensure the users are engaged.

Virtual Reality transforms fitness enthusiasts into digitally immersive environments, thus amplifying their workout experiences. For instance, users might perform a yoga routine on a virtual beach in the metaverse, or experience boxing in a realistic form, or cycling over mountainous terrains, all from the comfort of one's home. Such unique applications can bring an element of excitement and novelty to otherwise routine workouts. Furthermore, VR enables the development of interactive fitness regimes, creating an illusion of participating in a live class.

This brings in elements of motivation and social interaction to the workout experience, making it more effective (Shokurova, 2019).

Augmented Reality has the potential to revolutionize fitness by providing visual demonstrations of workouts within the user's environment. Such visual aids can enhance the accuracy of the user's movements and contribute to a safer workout regime. Additionally, AR can introduce dynamic challenges within a user's immediate surroundings to encourage more physical activity. This seamless blending of the real and the digital promises a more engaging and interactive fitness experience.

3. Social tools:

incorporating tools like connecting with other users to motivate one another and share accomplishments.

4. Mental Health Features:

A key component of wellness is mental health. A more complete health tool might be offered if functions like meditation exercises, stress tracking, or mood logging were added. Additionally, since sleep plays a vital role in psychological well-being, tracking

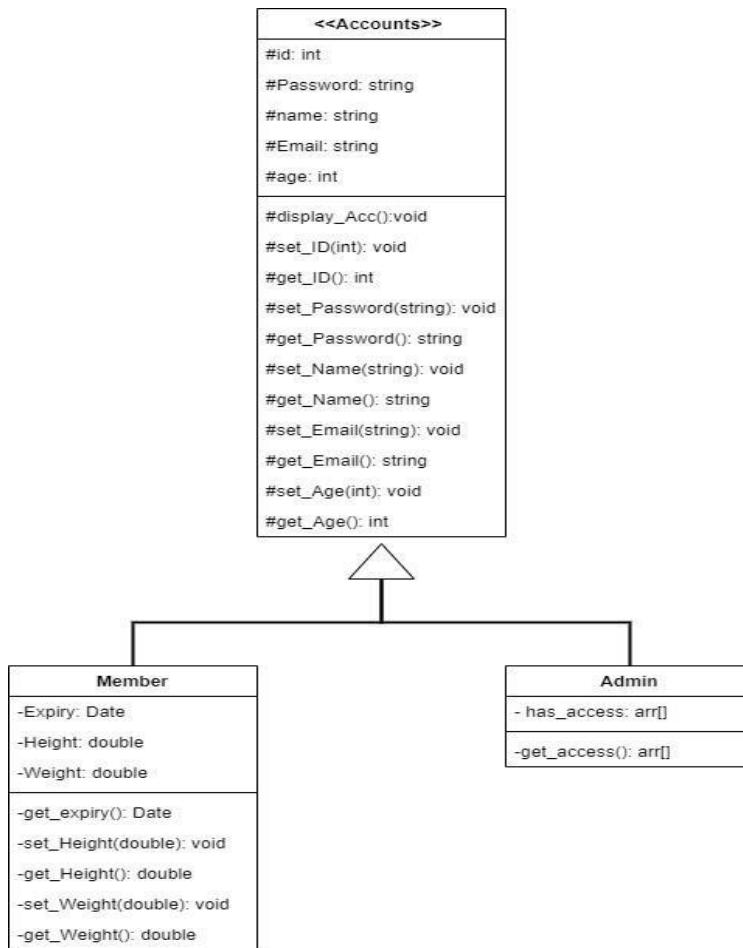
your sleep patterns can be another tool to monitor your mental health. Poor or irregular sleep may be a sign of mental stress, such as sadness or worry (Shokurova, 2019).

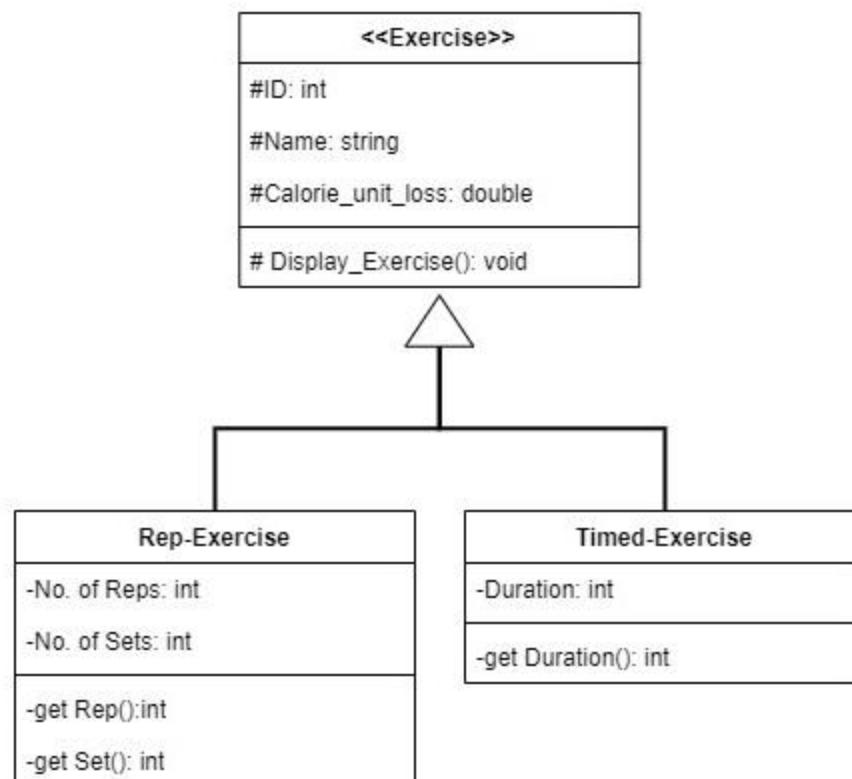
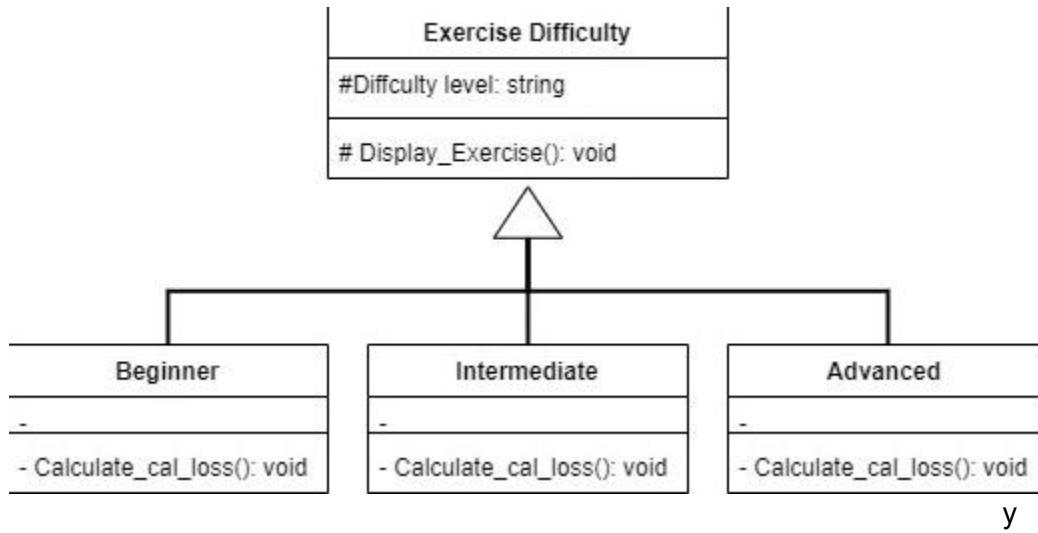
Future Improvements OOP:

OOP can be implemented later to keep standard coding, and to make the code understandable.

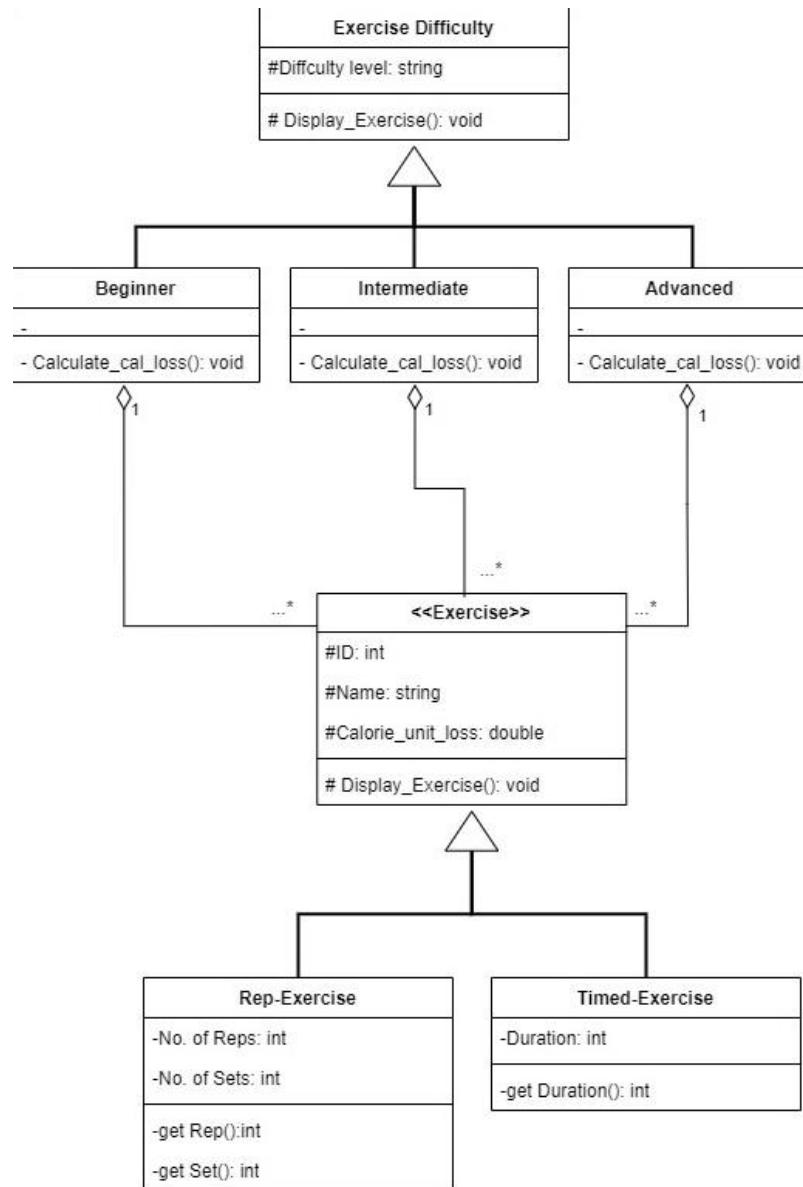
UML Class diagram (OOP)

Generalization & Polymorphism:





Aggregation:



Software Requirement Document



CSCI321 REQUIREMENTS

Team Blu

Dani Ternanni, Fatima TchanTchane, Noor
Ahmed, Ali Mela Ali, Abdullah Rafiei

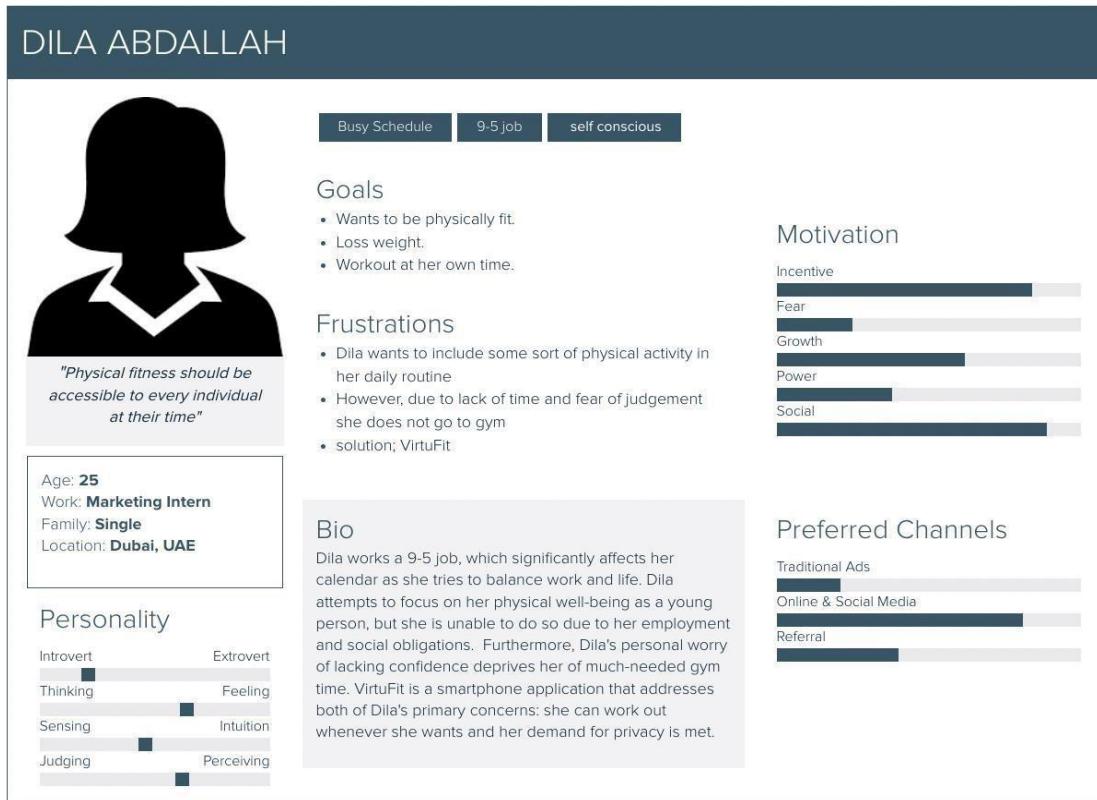
Introduction

The purpose of this project is to address the challenges faced by individuals when it comes to supervising their physical fitness. After thoroughly studying the reasons and issues raised when it comes to the negligence of working out and visiting the gym, we concluded that we should develop an application that tackles these concerns. VirtuFit is a mobile-based application integrated with artificial intelligence and computer vision to provide its users with the ultimate gym experience in the comfort of their homes and via their mobile phones. The application is envisioned to be tailored with minimal user input and serve them with benefits including personal customization for every individual, tracking user progress, monitoring their workouts, and providing them with an AI integrated chatbot available 24/7.

Intended Audience and Reading Suggestions

This document briefly explains the needs of our application VirtuFit, such as the external interface requirement, domain model, application features, and non-functional requirements, for all interested stakeholders of the VirtuFit application, such as its potential users, individuals interested in the backend development of an application and seeking knowledge regarding the implementation of technologies such as computer vision and AI, and team Blu. The VirtuFit application is available to every individual seeking fitness-related advice and benefits. The program is designed with the objective of making physical training accessible to everyone, independent of personal responsibilities, at their own place and time.

Below is a persona for a 9-to-5 marketing intern named Dila, who faces personal and social conflicts while wanting to maintain her fitness goals. Dila is one example of many cases that led us to the solution of developing the VirtuFit app.



Product Scope

The VirtuFit program places a strong emphasis on each user's unique experience, which is why the user is prompted to log into their account when they first launch the application. Once the user has submitted their credentials, the database will double-check the user's account's availability. The user will be greeted by a variety of VirtuFit functionalities after they have successfully accessed their privileges, including interactions with the AI chatbot, personalized workout plans and schedules, computer vision integrated feedback, and much more. The VirtuFit app is designed with extensive AI testing and training to speak with users and deliver appropriate feedback and answers to their questions. In addition, customers of the Virtufit application can change their schedule and exercise plan at any time via the workout schedule screen when it need due to sudden events.

Overall Description

PRODUCT PERSPECTIVE

The Virtufit mobile application augments existing fitness apps; it has unique features and functionalities that other fitness apps lack, providing users with high engagement and personalized users experience. The product focuses on the user's preferences and needs to reach wellness. The users in the mobile app will be directed to the virtual trainer, which is the chatbot, to share their goals, any constraints like health conditions, and their schedule. This product is specifically designed for individuals who prefer to train and workout at home due to personal reasons. Based on the user's preferences, the procedural programming system will generate a personalized workout schedule and plan. The user will use a standard phone stand to capture their movements and postures during fitness activities, facilitating the recording and sharing of their performance. Computer vision technology will be employed to analyze the user's movements, further ensuring the accuracy and correctness of their exercise form.

Product Functions

1. Sign up for VirtuFit account.
 - 1.1 Create Customer account.
2. Login to VirtuFit account
 - 2.1 Login to User account
 - 2.2 Forgot password.
3. Settings
 - 3.1 Update and change user information/accounts.
 - 3.2 Delete user account.
 - 3.3 Change Password
- 4 Motivate users.
 - 4.1 Display Schedule progress
 - 4.2 Display Calorie loss / gain progress

- 4.3 Display Duration of workout remaining
- 4.4 Display Motivational Quote
- 4.5 Provide Workout challenges.
- 4.6 Provide Affirmation
- 4.7 Display workout streak
- 4.8 Complimenting user efforts
- 5 Chat with VirtuFit AI trainer
 - 5.1 Inquire information about user.
 - 5.2 Create workout schedule.
 - 5.3 Create workout plan.
 - 5.4 Provide advice and recommendation.
 - 5.5 Respond to fitness or general questions.
- 6 Start workout.
- 7 Change/Edit schedule.
 - 7.1 Edit schedule time/date.
- 8 Display Exercises and how to perform them.
 - 8.1 Home workout body weight.
 - 8.2 Home workout using equipment.
- 9 Form correction
 - 9.1 Monitor user performance.
 - 9.2 Provide feedback on user's form.
- 10 Membership
 - 10.1 Payment.
 - 10.2 View current membership.

User Classes and Characteristics

Function-to-user relationships will be used to decide which features are suitable for a particular user type. There is one designated user class used in our work: user. The following table illustrates the functional permissions corresponding to the mentioned user groups.

Function	Global user
Sign up for VirtuFit account	X
Create Customer account	X
Login to VirtuFit account	X
Login to User account	X
Settings	X
Update and change user information/accounts	X
Delete user account (done by User)	X
Change Password (Forgot password)	X
Motivate users	X
Display Schedule progress	X
Display calorie loss / gain progress	X
Display Motivational Quote	X



Provide workout challenges	X
Display workout streak	X

Comments by trainer Recognize small effort and make comment to user	X
Chat with VirtuFit AI trainer	X
Inquire information about user	X
Create workout schedule	X

Create workout plan	X
Provide advice and recommendation	X
Respond to fitness or general questions	X
Start workout	X

Change/Edit schedule	X
Edit schedule time/date	X
Edit workout plan	X
Display Exercises and how to perform them	X
Home workout body weight	X
Home workout using equipment	X
Form correction	X

OPERATING ENVIRONMENT

The environment in which the mobile application operates will be diverse. Users will be able to use the program on a variety of android devices. To improve the quality of the user experience, the application will take advantage of the hardware features of mobile phones, such as cameras, and RAM. Since most of the features don't require internet access the app would work fine offline. Internet access is only required to access the virtual trainer, get workout advice.

The application will automatically adjust its resolution based on the different screen sizes and orientations, providing reliable and consistent performance for a variety of different Android phones. A crucial component of the operating

environment will be security. The application will use local DB on the user's device. Because of this it becomes more secure as the application would also work offline.

Design and Implementation Constraints

Our fitness app will be using react native CLI therefore, one major constraint is having to develop the same features on the IOS platform. Another major constraint is the use of local DB, this can cause compatibility issues when user switches or changes device.

EXTERNAL INTERFACE REQUIREMENTS

USER INTERFACE

The mobile app will have a simple user interface that enables users to login and communicate with a virtual trainer. The interface will make it easier to access the application's functionalities, resulting in a more seamless user experience. Users will be able to obtain custom training programs, workout recommendations, and progress tracking by logging in with their credentials. The user interface will have a bunch of different functionalities that enable them to interact with the virtual trainer. Once the user is logged in, their home screen will be displayed. This home screen will have features that allow the user to communicate with the virtual trainer, check their personalized training schedule, view their workout plan, and track their progress. During the exercise, the user will also be able to use the form monitoring feature to get feedback on their performance.

COMMUNICATIONS INTERFACES

When a user converses with the virtual trainer, their inputs are sent from their mobile device to the flask server, which is then processed. The mobile application will get the required replies from the virtual trainer when the server has processed the user's input. This method of communication provides a user engaging experience.

Furthermore, for getting feedback from the AI about your exercise, the data will be sent from the mobile application to a server. This server will then access APIs of

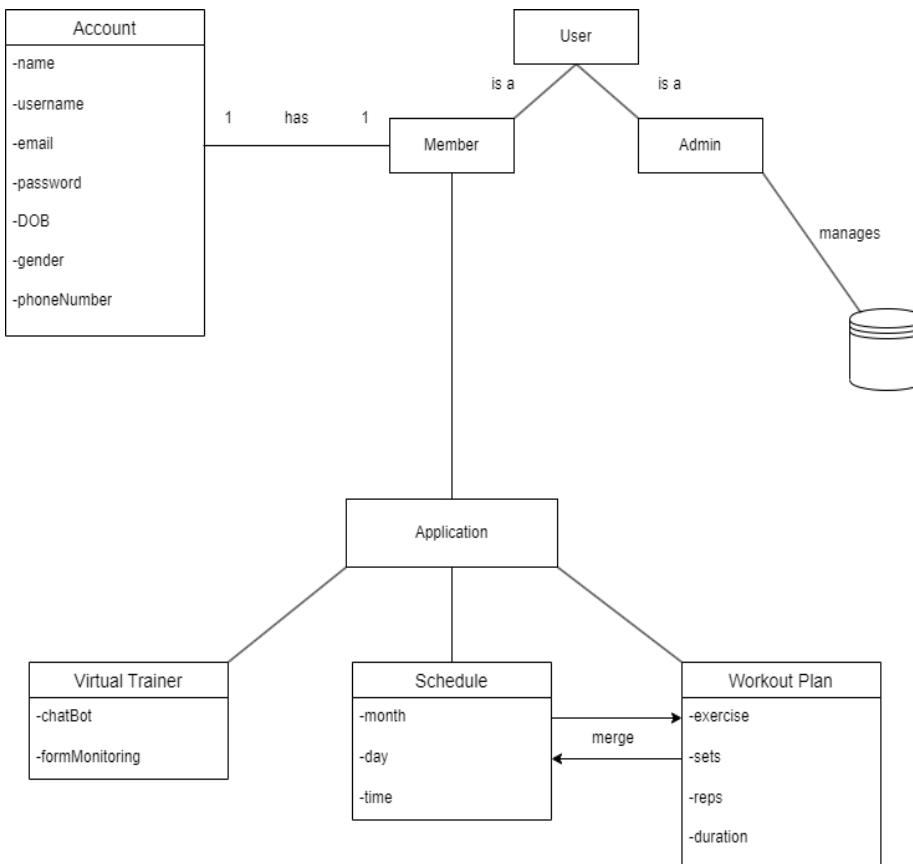
different AI services using flask, to process the data. Once we have the processed data, it will be sent back to the application and can be used or displayed to the user. Lastly, the application prevents all attempts of SQL injection by making use of anti-injection procedures.

SOFTWARE INTERFACES

The mobile application will use multiple software components and platforms to provide the desired functionalities. React Native will be the leading platform and framework for mobile app development. SQLite will be used as the backend platform for the authentication of users, training schedule storage, storage of exercises, and user progress. It provides data storage and retrieval services, which will boost the application's security as it will be harder for malicious users to get access.

Flask has also been used to create a web API that integrates with the mobile application and takes care of features like chatbot API communication, form correction and feedback. The use of this API will make it easier to communicate with external systems. Additionally, users will be able to communicate with the virtual trainer using our chatbot, which will be trained using the OpenAI API. This API will be used to train our model to answer general questions about fitness and exercise with accurate information.

DOMAIN MODEL



System Features (Use Cases)

This section of the report provides an overview of use cases along with a description of the use cases and functional requirements of VirtuFit's features and process. Furthermore, this section can provide insight as to which user has access to which features, and which features branch into more detailed features.

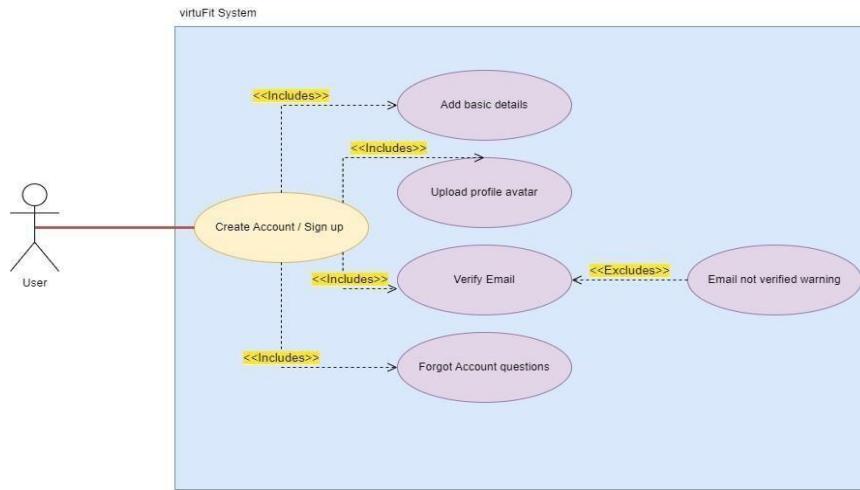
List of VirtuFit System features:

Sign up for VirtuFit Account

Create Customer account

Account Sign up	
Function Name	Create a new Virtufit account
Description / Goal	Users should be able to create a personal account in the app.
Inputs	<ul style="list-style-type: none"> ● Name ● Username / Email ● Password ● Confirm password ● Age
Source	The source of all the inputs mentioned will come from the user.
Outputs	New account created and added to database
Destination	The user table in the database is affected and a new user is added.
Action / Steps	<ul style="list-style-type: none"> ● Step 1: User opens the app on the mobile device. ● Step 2: User clicks on sign up button in the VirtuFit app. ● Step 3: User inputs valid information for account creation.
Requires	Valid user information is required to execute this feature.
Pre-condition	<ul style="list-style-type: none"> ● An already existing email ● The VirtuFit app should already be installed
Post-condition	New user account is created,
Side-effects	A new account information is added to the database.
Exceptional Scenario	All fields must be filled correctly to register and create a new account.

Sign up use case overview:

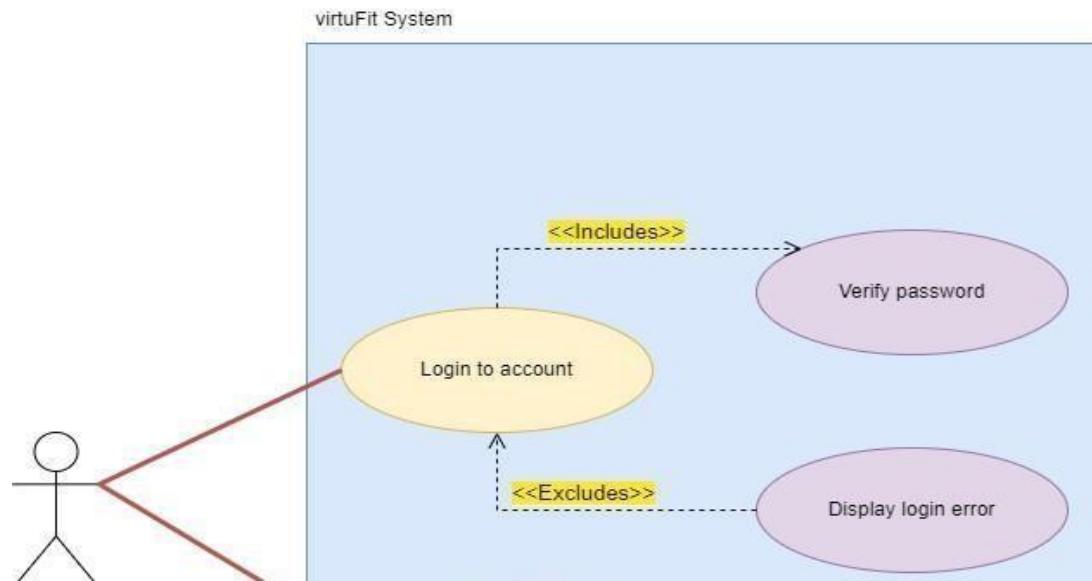


Login to VirtuFit Account

Login	
Function Name	Login to existing account
Description / Goal	Users should be able to successfully login to their account
Inputs	Credentials <ul style="list-style-type: none"> • Username / Email • Password
Source	The source of all the inputs mentioned will come from the user.
Outputs	Successful login: user is granted access to their account
Destination	Database confirms user User will be logged into the main screen of the application
Action / Steps	<ul style="list-style-type: none"> • Step 1: User opens the app on the mobile device. • Step 2: User clicks on the login button. • Step 3: User inputs valid credentials. • Step 4: User is granted access to their account.

Requires	Valid user credentials are required
Pre-condition	<ul style="list-style-type: none"> • Users must have an existing account on the application. • The VirtuFit App should be already installed.
Post-condition	User is logged in to the application
Side-effects	Active users increased by one.
Exceptional Scenario	Users can request to change password in case of invalid credentials

Login use case overview:



Update Account

Update and change user information/account

Update User information	
Function Name	Change profile information
Description / Goal	Users will be able to change user profile information such as (username, age etc.)
Inputs	User inputs updated information and confirmed the changes.
Source	User will edit the details in the profile and confirm the changes done to the profile.
Outputs	Users' new information overrides the previous data, and the new information is saved and displayed in the user's profile.
Destination	The user table in the database is affected and the new changes are made, and the table is updated.
Action / Steps	<ul style="list-style-type: none"> • Step 1: User will press on the profile button and the user's information will be displayed. • Step 2: Users can choose to edit any profile detail such as (Username, age etc.) <p>Step 3: once the changes are done, users can choose to save or cancel the changes.</p>
Requires	<ul style="list-style-type: none"> • Users confirmation to the changes is a must otherwise the changes will not be saved.
Pre-condition	<ul style="list-style-type: none"> • User account must exist. • User should be logged into the account to make the changes
Post-condition	User profile information is updated to the new changes.
Side-effects	User table in the database is called upon and an update is done to change the information.
Exceptional Scenario	<ul style="list-style-type: none"> • Users must have a stable connection during this change; any disconnection can lead to a refresh in the page causing the unsaved changes to disappear. • After the user is done editing information if he/she does not confirm the change the new data will not override the old data.

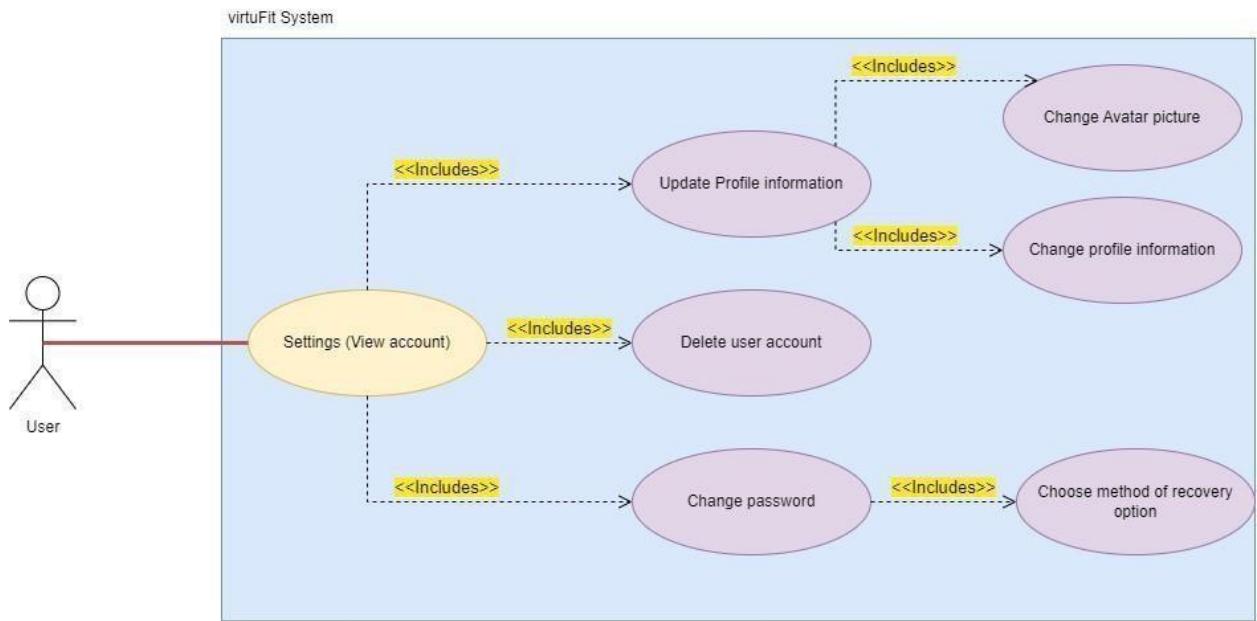
Delete user account (done by User)

Delete Accounts	
Function Name	Delete user account
Description / Goal	Successfully delete a user's account
Inputs	Button touch from user
Source	User confirmation
Outputs	Account is deleted
Destination	Account is deleted from the database
Action / Steps	<ul style="list-style-type: none"> ● Step 1: User opens the app on the mobile device. ● Step 2: User logs in to the application ● Step 3: User selects delete account option from settings. ● Step 4: User gets a confirmation message. ● Step 5: If yes is selected, account is deleted
Requires	User is required to confirm deletion
Pre-condition	<ul style="list-style-type: none"> ● User must have an existing account. ● User must have a valid membership. ● User must be logged in
Post-condition	User account is deleted
Side-effects	User information is deleted from the database
Exceptional Scenario	N/A

Change Password

Change Password	
Function Name	Change account login information
Description / Goal	Users should be able to change their accounts password for security purposes
Inputs	<ul style="list-style-type: none"> • Old password • New password
Source	User should press change password button
Outputs	New tab is opened, and users can change their password
Destination	User's account information in the database is affected
Action / Steps	<ul style="list-style-type: none"> • Step 1: The user must select the change password option. • Step 2: user must enter old password. • Step 3: user must enter new password. • Step 4: user confirms new password.
Requires	Accurate old password.
Pre-condition	<ul style="list-style-type: none"> • Account must exist. • User must be logged in to use settings option.
Post-condition	The user's password for their account is updated.
Side-effects	N/A
Exceptional Scenario	User doesn't know the answer to any recovery option then they have to communicate with an admin or tech support.

Setting use case overview:



Chat with VirtuFit AI trainer

Inquire information about user

Chat with virtual trainer	
Function Name	Inquire information about user
Description / Goal	To gather all the necessary information about the user including their age, medical conditions, schedule, and goal.
Inputs	User inputs their texts via the keyboard on screen
Source	Screen keyboard
Outputs	Virtual trainer responds back with text.

Destination	Conversation page / frame
Action / Steps	<ul style="list-style-type: none"> • Step 1: User opens VirtuFit application. • Step 2: User logs into the application • Step 3: User clicks chat option. • Step 4: User converses with the “virtual trainer”
Requires	User must be on the conversation page with the virtual trainer
Pre-condition	<ul style="list-style-type: none"> • User must have an existing account. • User must be logged in
Post-condition	A conversation between user and virtual trainer will exist.
Side-effects	Virtual Trainer is up to date on user needs
Exceptional Scenario	User enters the conversation page but does not engage in a conversation, after some time the virtual trainer will ask a question to confirm with the user

Create workout schedule

Workout Schedule Creation	
Function Name	Create workout schedule
Description / Goal	Application builds a specialized unique workout schedule for specific user
Inputs	User details.
Source	Conversation app
Outputs	Schedule is created
Destination	Database saves new schedule for user

Action / Steps	<ul style="list-style-type: none"> Step 1: User engages with the virtual trainer through a conversation. Step 2: All info gathered will be fed to rule based algorithm. Step 3: The algorithm will build a personalized schedule for user. Step 4: Schedule will be available for user to view
Requires	User to provide their details to the application through conversation
Pre-condition	<ul style="list-style-type: none"> User must have an existing account User must be logged in
Post-condition	Schedule is created
Side-effects	N/A
Exceptional Scenario	In case the schedule is not satisfying the user, it can be changed via edit schedule option.

Create workout plan

Workout Plan	
Function Name	Create workout plan
Description / Goal	To create a workout plan that includes all necessary exercises imbedded along the schedule / timeframe for the user
Inputs	Schedule User details.
Source	Conversation, schedule
Outputs	Specialized workout plan
Destination	Workout plan / regime page
Action / Steps	<ul style="list-style-type: none"> Step 1: User logs into the application Step 2: User converses with the virtual trainer to gather information and build schedule. Step 3: Details fed to the rule-based algorithm to build workout plan for user.

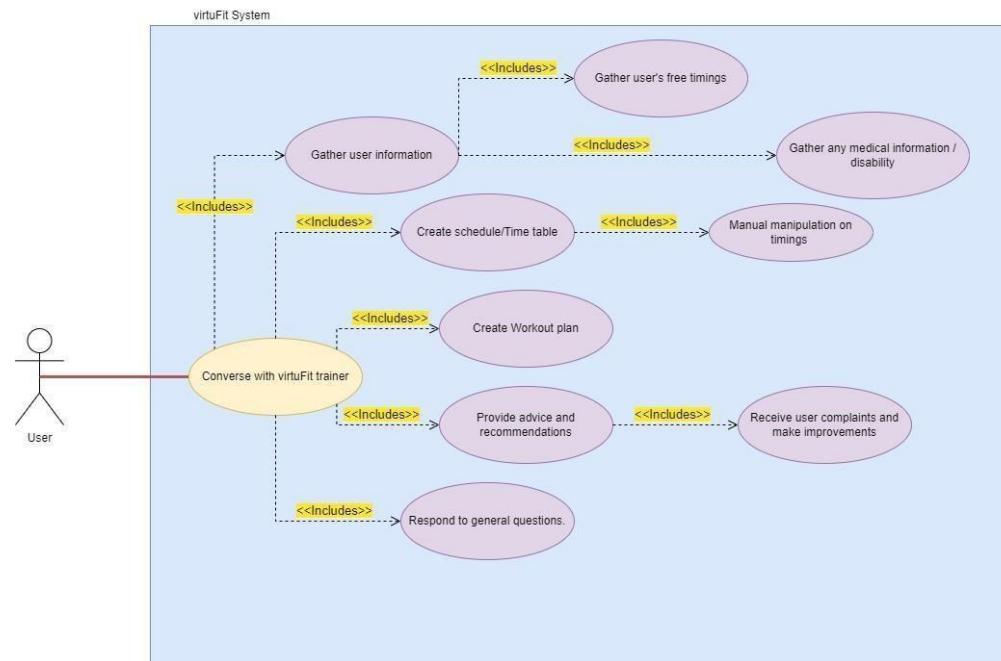
	<ul style="list-style-type: none"> Step 4: Algorithm creates the workout plan for the user
Requires	User to select create workout plan option.
Pre-condition	<ul style="list-style-type: none"> User must have an existing account. User must be logged in. User must provide necessary details through conversation User must have a schedule ready
Post-condition	Workout plan must be available
Side-effects	N/A
Exceptional Scenario	N/A

Respond to fitness or general questions

Chatbot - Answer Fitness questions	
Function Name	Respond to fitness related inquiries
Description / Goal	To respond to any fitness questions, the user asks the virtual trainer on the conversation page
Inputs	User questions
Source	Keyboard on the screen
Outputs	Answers to the questions are provided for the user
Destination	Conversation page on UI

Action / Steps	<ul style="list-style-type: none"> Step 1: User engages with the virtual trainer through a conversation. Step 2: User asks the virtual trainer fitness related questions. Step 3: Trained model will provide answers to the user questions
Requires	User to ask a question
Pre-condition	User must have an existing account. User must have a valid membership. User must be logged in
Post-condition	Correct answers are provided for the user
Side-effects	N/A
Exceptional Scenario	In case the user asks a question unrelated to fitness, the virtual trainer should state that it only answers fitness related questions

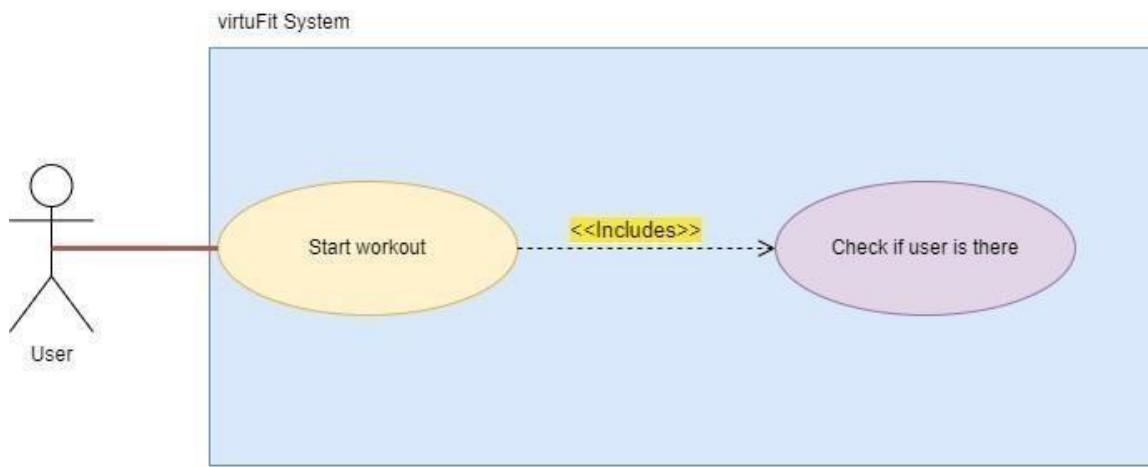
VirtuFit trainer use case overview:



Start workout

Start workout	
Function Name	Start workout
Description / Goal	User starts the workout and the workout plan for the day begins
Inputs	Button touch
Source	User selection
Outputs	Workout structure for the day is shown
Destination	UI (workout page)
Action / Steps	<ul style="list-style-type: none"> • Step1: User logs in to the application. • Step 2: User selects start workout option. • Step 3: Workout of the day begins with the timer/reps and sets ready for each workout
Requires	User to have an existing workout plan
Pre-condition	<ul style="list-style-type: none"> • User must have an existing account. • User must be logged in
Post-condition	User completes the day's workout
Side-effects	N/A
Exceptional Scenario	If the user is busy and cannot workout, they edit schedule and shift their workout.

Start workout use case overview:



Change/Edit Schedule

Edit schedule time/date

Edit schedule	
Function Name	Edit schedule
Description / Goal	User can edit their own schedule manually
Inputs	Edit Button touch
Source	User selection
Outputs	Editable schedule
Destination	UI (schedule page)
Action / Steps	<ul style="list-style-type: none"> • Step 1: User logs into the application • Step 2: User selects view schedule option.

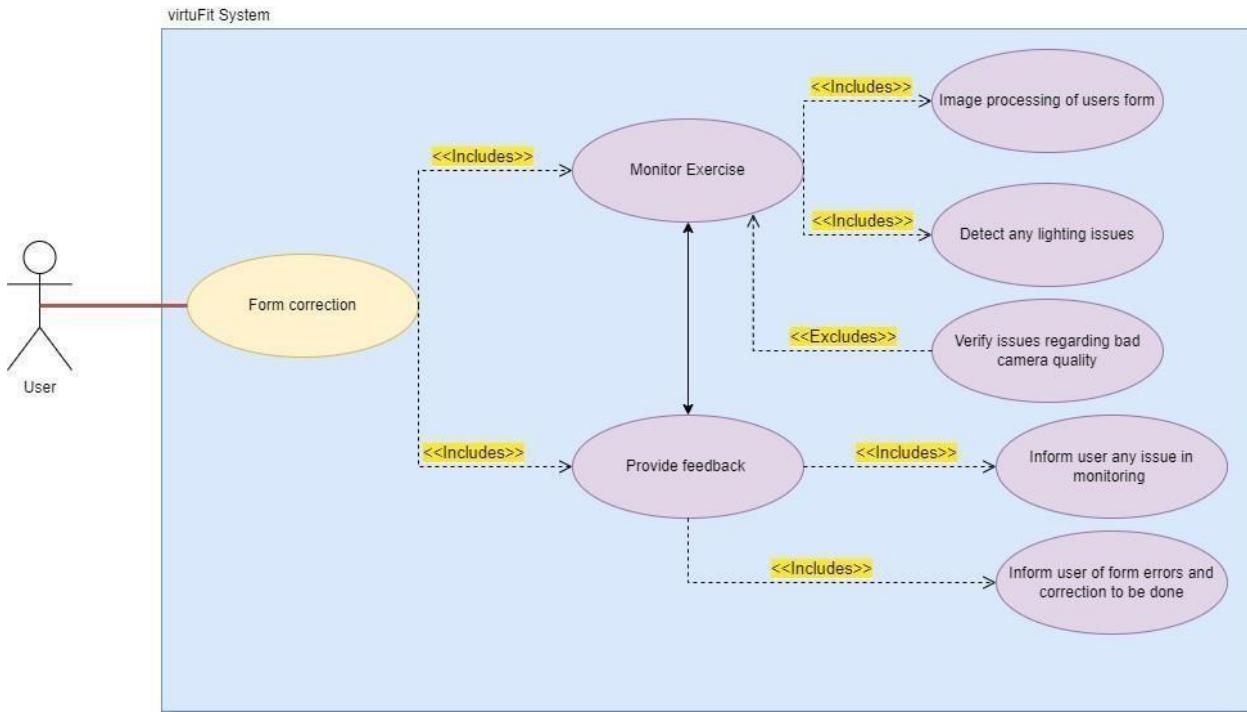
	<ul style="list-style-type: none"> • Step 3: User selects edit schedule option. • Step 4: User edits their schedule accordingly
Requires	User to have an existing schedule
Pre-condition	<ul style="list-style-type: none"> • User must have an existing account • User must be logged in
Post-condition	Schedule is updated
Side-effects	The workout plan adjusts to the updated schedule
Exceptional Scenario	N/A

Provide feedback on user's Form

Form Monitoring	
Function Name	Provide feedback on workout form
Description / Goal	To give accurate feedback on users form through computer vision if they perform the exercise incorrectly. The pre trained pose detection model will be using angles and position of joints to correct user form.
Inputs	Livestream camera of user performing the exercise.
Source	User, camera
Outputs	Detailed feedback
Destination	UI

Action / Steps	<ul style="list-style-type: none"> Step 1: User selects the monitor form option. Step 2: User is shown guidelines to perform the monitoring. Step 3: User performs the exercise following the guidelines. Step 4: User actions are captured and processed with angles and joints calculated. Step 5: Virtual trainer provides verbal feedback as well as textual feedback.
Requires	<p>User to be amid their workouts.</p> <p>User must follow the guidelines for the feature to function the way it is supposed to</p>
Pre-condition	<p>User must have an existing account.</p> <p>User must be logged in</p>
Post-condition	Sets and Reps of user during exercise is stored and user moves to next exercise.
Side-effects	N/A
Exceptional Scenario	If one of the guidelines are not met, the application will tell that to the user

Form correction use case overview:



Nonfunctional Requirements

Performance Requirements

Certain functional requirements of the VirtuFit application require a lot of resources in terms of power, storage, processor and RAM than a mobile phone can provide. Therefore, to improve performance simple features that require less resources will be on the client side whereas, more complex features are implemented using server request and API's this way there is a lower chance of delay between UI elements and response, and lower chance of overconsumption of user's device resources which can lead to fewer crashes.

One such example of a Virtufit requirement which requires a lot of resources is the Chatbot. This is because, constant internet access is needed to answer general question. Therefore, having this module connected to a server using API will automatically handle the response and requests which as a result, will reduce the load on the device reducing delays and crashes. However, not all requirements can be stored in server's even if they have a high resource requirement. An example of this is AI Form monitoring. Although it will reduce the load to have the Form detection in a different server, it will severely hinder the App's usability. This is because, Form detection module of the VirtuFit application should provide feedback whenever the user wants to start workout exercise regardless of whether the user has internet or not. Therefore, having the form detection in the user's device will improve usability.

Usability Requirements

The VirtuFit app has a very intuitive layout that has been carefully designed to improve the overall user experience. The usage of informative infographic icons is one of the main elements enhancing its usability. These icons work as visual aids, providing a brief description of each application button's purpose. VirtuFit guarantees that users may easily navigate through the many functions without any confusion, even if they are unfamiliar with fitness apps, by employing such a design technique.

Furthermore, the VirtuFit application gains additional ease from the user-friendly screen navigation. The layout is simple and uncomplicated when users first arrive, which reduces the learning curve and frees them up to concentrate on their fitness journey rather than figuring out a complicated interface. The user-centric design of the panels makes sure that all tasks, such as tracking accomplishments and creating fitness goals, can be completed with a few taps or swipes.

Reliability

The application uses OpenAI API to offer consumers a chatbot that is available around the-clock to answer their questions. We will train the virtual trainer to respond to fitness specific queries and develop schedules and regimes that fit best according to their scenario to give users accurate results that are tailored to their demands and focused on their fitness.

Safety Requirements

Utilizing computer vision, the VirtuFit program uses the client's camera to track their form. VirtuFit makes sure to not save any camera data to protect user privacy and foster trust. The application will delete all recorded video from their database as soon as the user receives the necessary feedback during the exercise. This requirement was put in place while considering users from diverse backgrounds and providing them comfort of choice with their appearance while they workout.

Security Requirements

Utilizing computer vision, the VirtuFit program uses the client's camera to track their form. VirtuFit makes sure to not save any camera data to protect user privacy and foster trust also, all form monitoring exercises will be giving feedback live stream and will not store or record any images of the user performing the exercise.

Design Document



CSCI321

Design Document

Team Blu

Introduction

Purpose

This report will give a detailed description of the VirtuFit design by showcasing its architectural background. This document includes numerous aspects of the project, including high-level and low-level design, the system architecture's structure, and the user interface design. These designs ensure the project is well structured and within the development scope. A Software Design document is needed in this project to satisfy the user's needs and system requirements, streamline the development process, and enhance communications between the team members. It also helps deliver a high-quality and practical fitness application that meets the user's preferences and needs. In addition, diagrams such as the class diagram, activity diagram, and data flow diagram are included in this document to exhibit the development process and provide an excellent visual representation.

Scope

Our software, VirtuFit, is designed specifically for individuals who perform home workouts. The app features a virtual trainer that communicates with users to assist them with their fitness goals. Furthermore, VirtuFit includes a Live-stream feature which provides real time feedback. Due to the description of the software, it is crucial to guarantee that the user is fully engaged in effective workouts within appropriate durations and in a comfortable environment in which the software will serve all these things to the user. VirtuFit serves as a workout tool and a comprehensive fitness companion that assists users in executing the right exercises precisely.

Overview

This Software design documentation covers all the design aspects. Section System Architecture demonstrates the user's interaction with the system or interface, and based on the user's requirements, a feature in the Virtufit application is implemented. The database is used to store information. The following sections contain the class, activity, data flow, and sequence diagrams. In addition, Section Human Interface Design includes all the interface screens of the system.

System Overview

VirtuFit is a mobile application to help users work out by conversing with a AI fitness trainer. This is done in hopes of achieving users' fitness goals. The VirtuFit application has several crucial features.

Main features of the VirtuFit application:

1. Workout Schedule:

The workout schedule will help inform users what workouts are available during the week.

2. Chat with a VirtuFit trainer:

Several features to help users work out are in this chat module.

Conversing with the trainer can allow users to inquire about general fitness questions, receive advice to help achieve their goals, and communicate with the trainer to help develop workout plans and workout schedules.

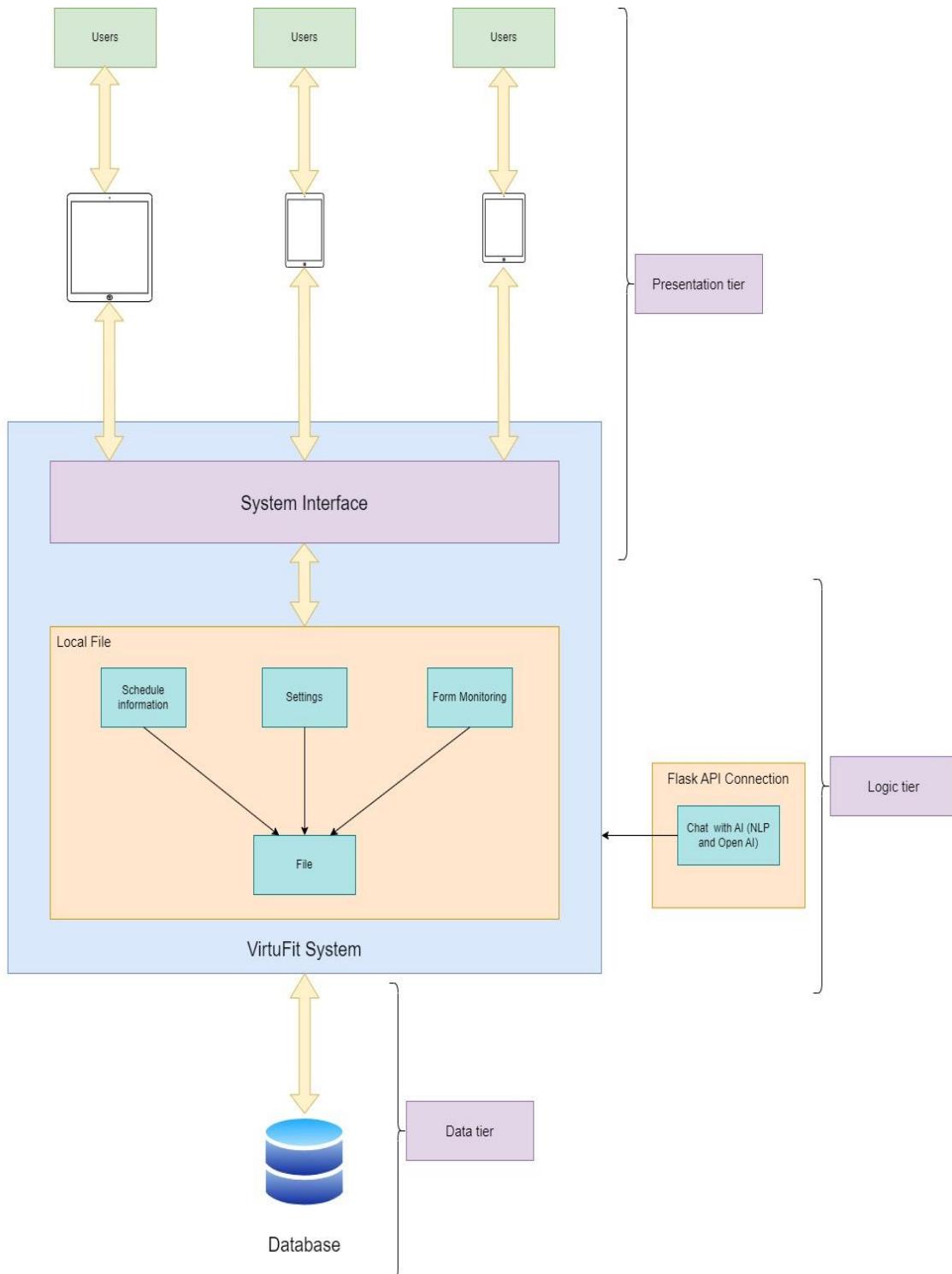
3. Form correction:

During workouts, provide feedback to users regarding exercise form. This is done after the user's movement is monitored, and feedback is sent back to the user.



System Architecture

Architectural design



Three Layer architecture:

Presentation layer:

- The VirtuFit application can be presented on many android devices.
- VirtuFit UI development will use React Native, which allows small parts of the logic layer to be implemented in the presentation layer.

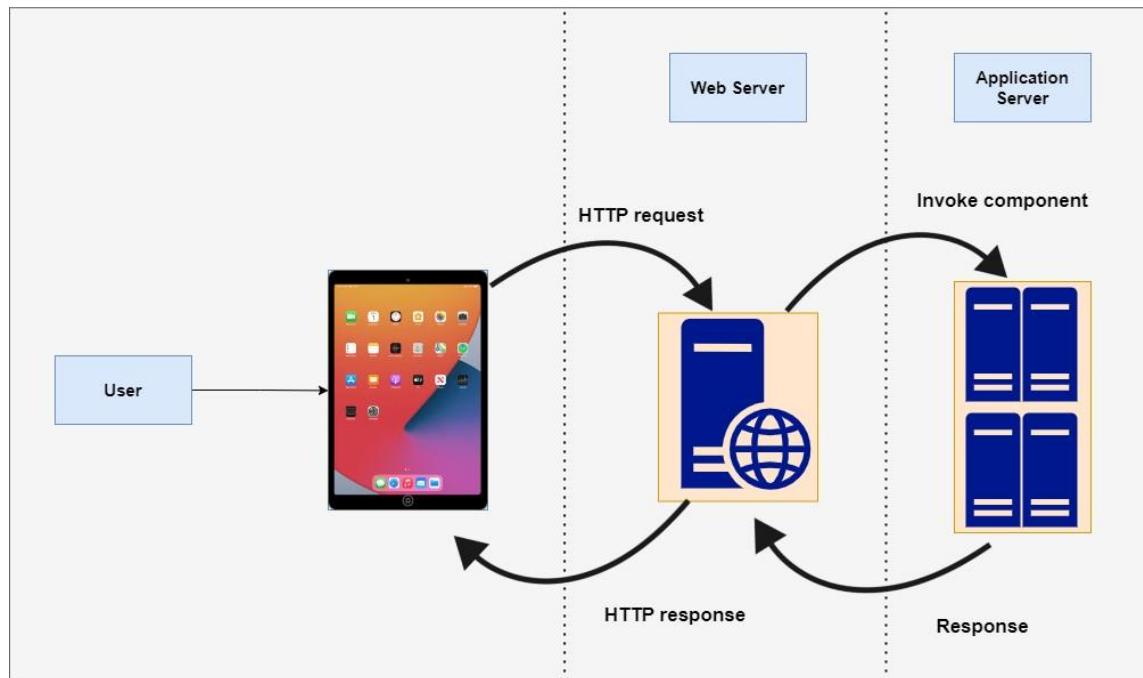
Logic layer:

- This layer handles most of the features of the VirtuFits application, such as API's, processing payments, etc.
- This layer is also responsible for communicating with the presentation tier and making sure requests such as schedule changes or schedule information are processed.
- Currently this layer is made within the local device or users.

Data layer:

- The data layer of the VirtuFit application is planning to use SQL lite to authenticate and store user information securely.
- Communicating with the logic layer to send data such as Schedule changes or authenticating user through local DB.

Physical Architecture



The physical architecture follows a 2-tier architecture. Tier 1 is the web server; Tier 2 is the application server.

Most of the features of VirtuFit is implemented locally this includes the workout schedule and the form monitor. However, the chatbot is connected using flask API which sends the request and receives response from server.

Web Server:

- Used for making requests when a user interacts with the VirtuFit Application chatbot and receives the response from the application server.
- This HTTP request is sent by the Flask API.

Application server:

- Handling and managing said requests.
- Sends and Receives requests from and to Chatbot Module.
- Application server is on Flask API.

Decomposition description

In the architectural design, there are several modules: the system interface, the local files, which contain all crucial features of the VirtuFit application, and lastly, the database.

- **System interface:** The system interface is in the presentation layer; this is where all UI elements and UI aspects of VirtuFit application are visible to users and interactions can be made.
- **Local files:** The local file in the architecture contains the main features of the VirtuFit application, and this module can be broken into different subsystems such as schedule information, Setting, and chatting with virtual trainer.
- **Database:** The main idea is how data information is stored, mostly including user information, schedule information, membership status information, etc. all of which are done locally currently.

UML Class diagram (OOP) :

Part of Future Improvements

Design rationale

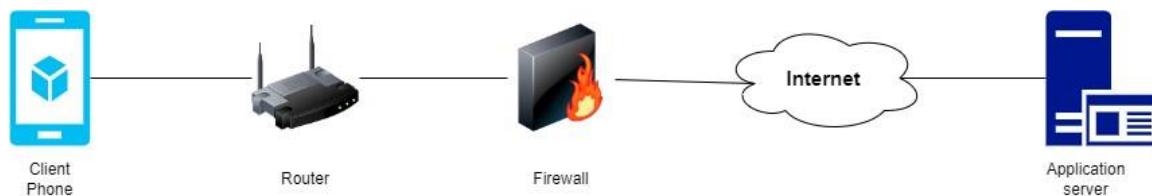
Considering 2-Tier architecture vs 1 or 3-Tier

2-Tier vs 1 3 Tier	
Advantages	Disadvantages
It is easy to maintain 2 tier architecture	Performance degrades when user increases
Easy to modify architecture	
Communication is fast	

The architectural design as well as the physical architecture both follow a 2-tier data architecture. There is one certain cost when following a 2-tier architecture compared to a 1 or 3 Tier such as Performance issues when users increase however, the benefits of 2-tier architecture such as being easy to maintain, easy to modify which is good for prototype projects, and communication being faster far outweighs those costs.

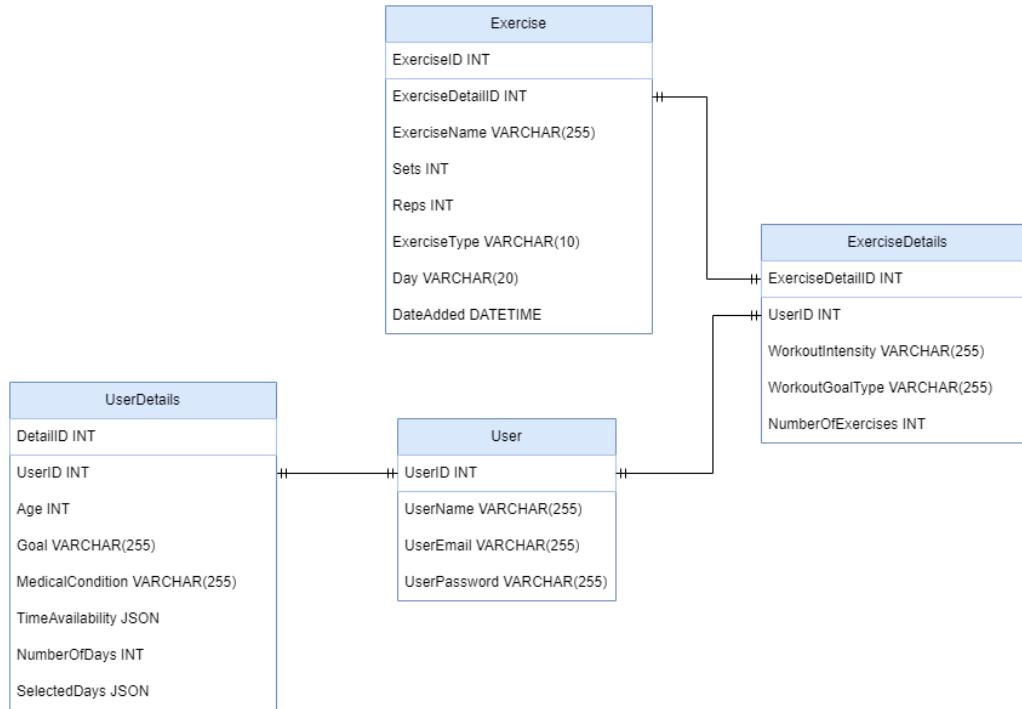
VirtuFit is a mobile application which in development will go through various prototypes. Because of this having 2 tier architecture makes it easier to modify making it less costly in the future.

Network architecture



Data design

ERD diagram



Data description

We first have users that interact with the application. These users all have their login credentials like username, user email, and user password. Once the user opens the app initially, they interact with a chatbot that stores data into user details table. The data that we extract from the user is their age, goal, medical condition, if any, and time availability. Once we have collected that data and the user opens home page, we run procedural programming code that adds to the user Details table the number of days the user will be working out in and the selected days. We will also run procedural programming code to generate a workout plan. This plan we will store it in the exercise details table responsible for the user. We will store data like workout insensity, workout goal, and number of exercises per day. Once we have all of this, we will add data to the exercise table based on the userdetails and exercise details. The exercise table will have data such as the exercise name, number of sets and reps, exercise type, day, and date added. These exercises found in the table will be displayed to the user once they open schedule or press on start workout.

Data dictionary

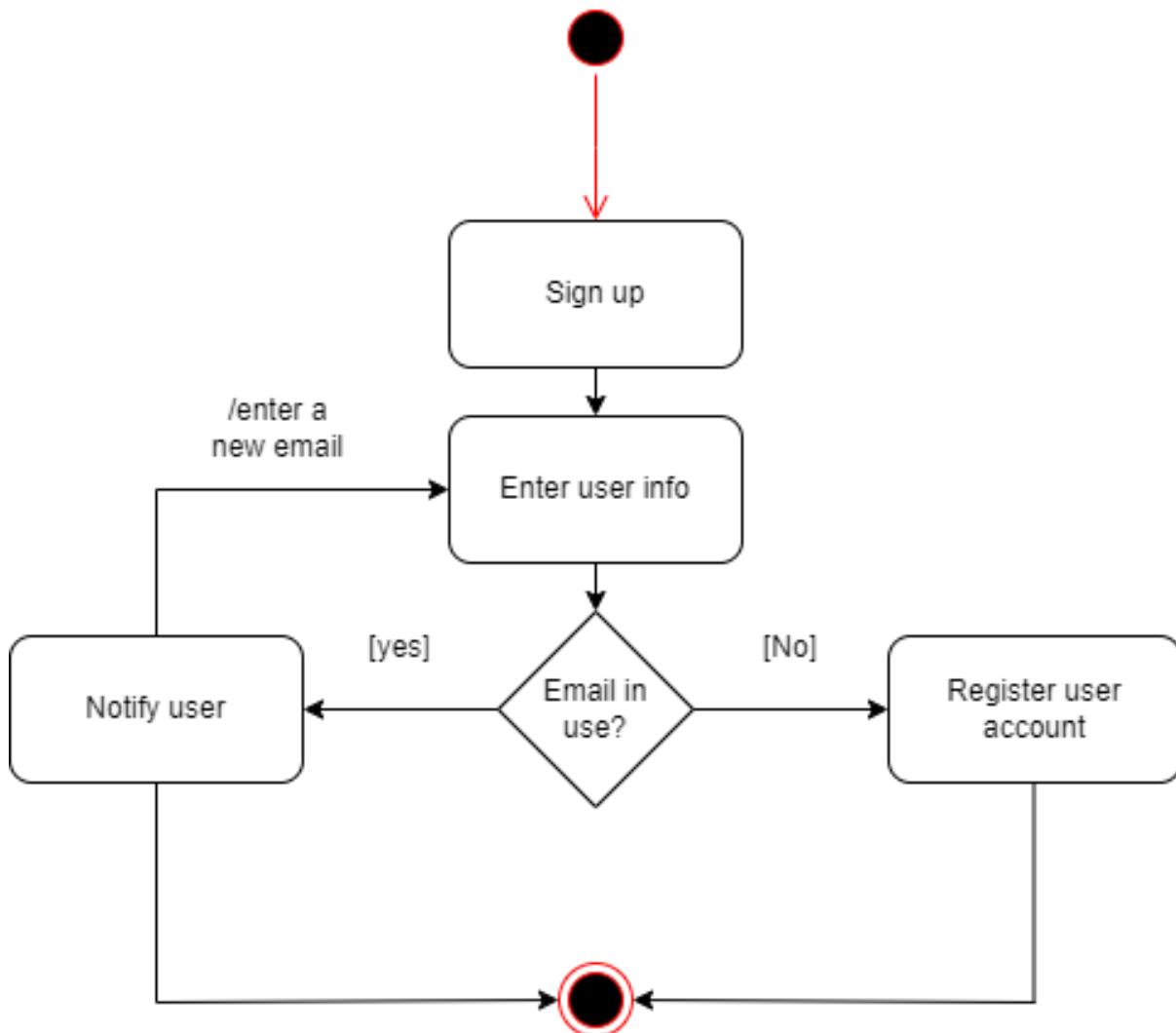
	Field	Type	Null	Default
User	UserID	INT	No	AUTOINCREMENT
	UserName	VARCHAR	No	
	UserEmail	VARCHAR	No	
	UserPassword	VARCHAR	No	
UserDetails	DetailID	INT	No	AUTOINCREMENT
	UserID	INT	No	AUTOINCREMENT
	Age	INT	No	
	Goal	VARCHAR	No	
	MedicalCondition	VARCHAR	No	None
	TimeAvailability	JSON	No	
	NumberOfDays	INT	No	
	SelectedDays	JSON	No	
	ExerciseDetailID	INT	No	AUTOINCREMENT
ExerciseDetails	UserID	INT	No	AUTOINCREMENT
	WorkoutIntensity	VARCHAR	No	
	WorkoutGoalType	VARCHAR	No	
	NumberOfExercises	INT	No	
	ExerciseDetailID	INT	No	AUTOINCREMENT
Exercise	ExerciseID	INT	No	AUTOINCREMENT
	ExerciseDetailID	INT	No	AUTOINCREMENT
	ExerciseName	VARCHAR	No	
	Sets	INT	No	

	Reps	INT	No	
	ExerciseType	VARCHAR	No	
	Day	VARCHAR	No	
	DateAdded	DATETIME	No	CURRENT_TIMESTAMP

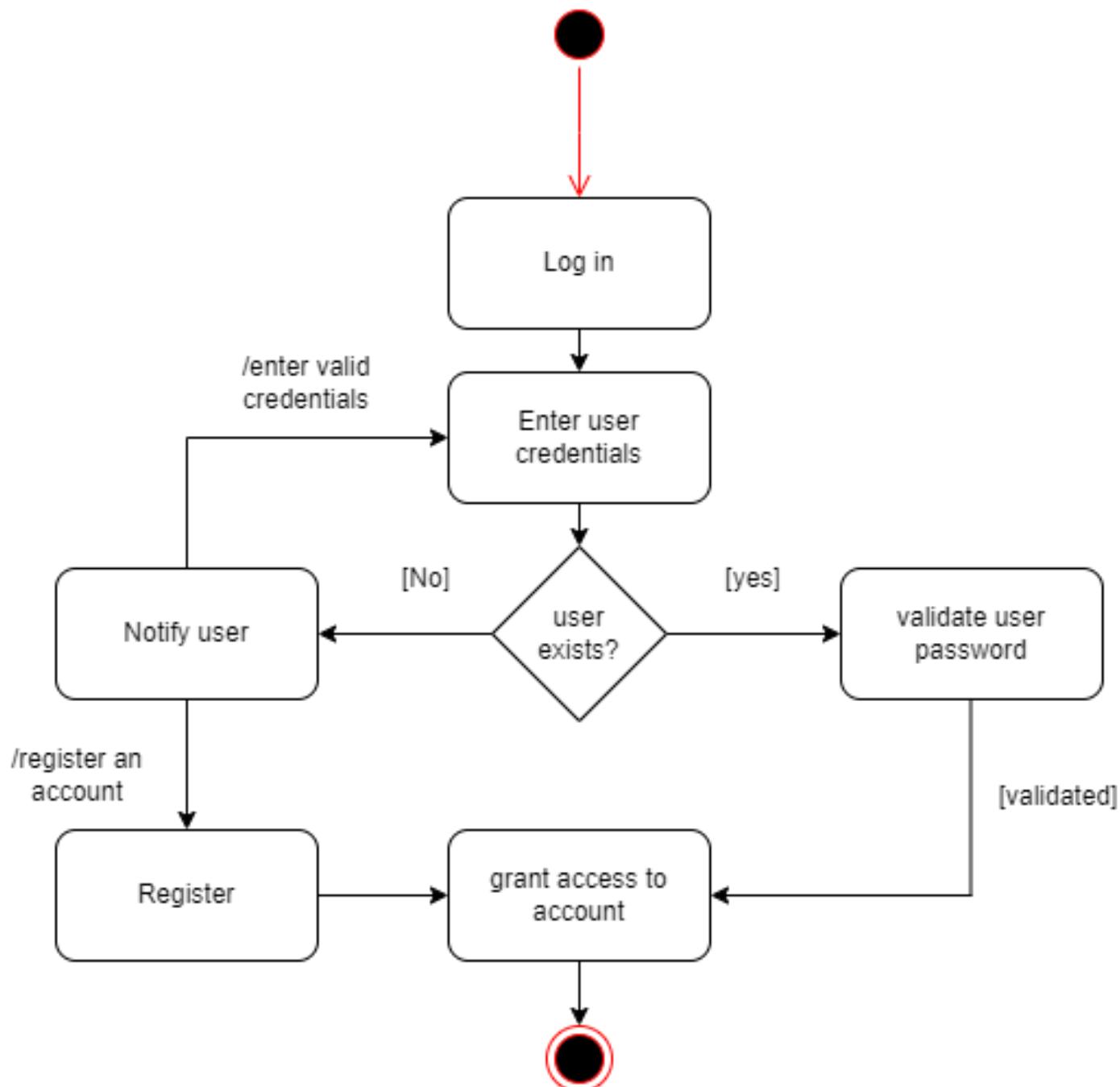
Component Design

Activity diagram

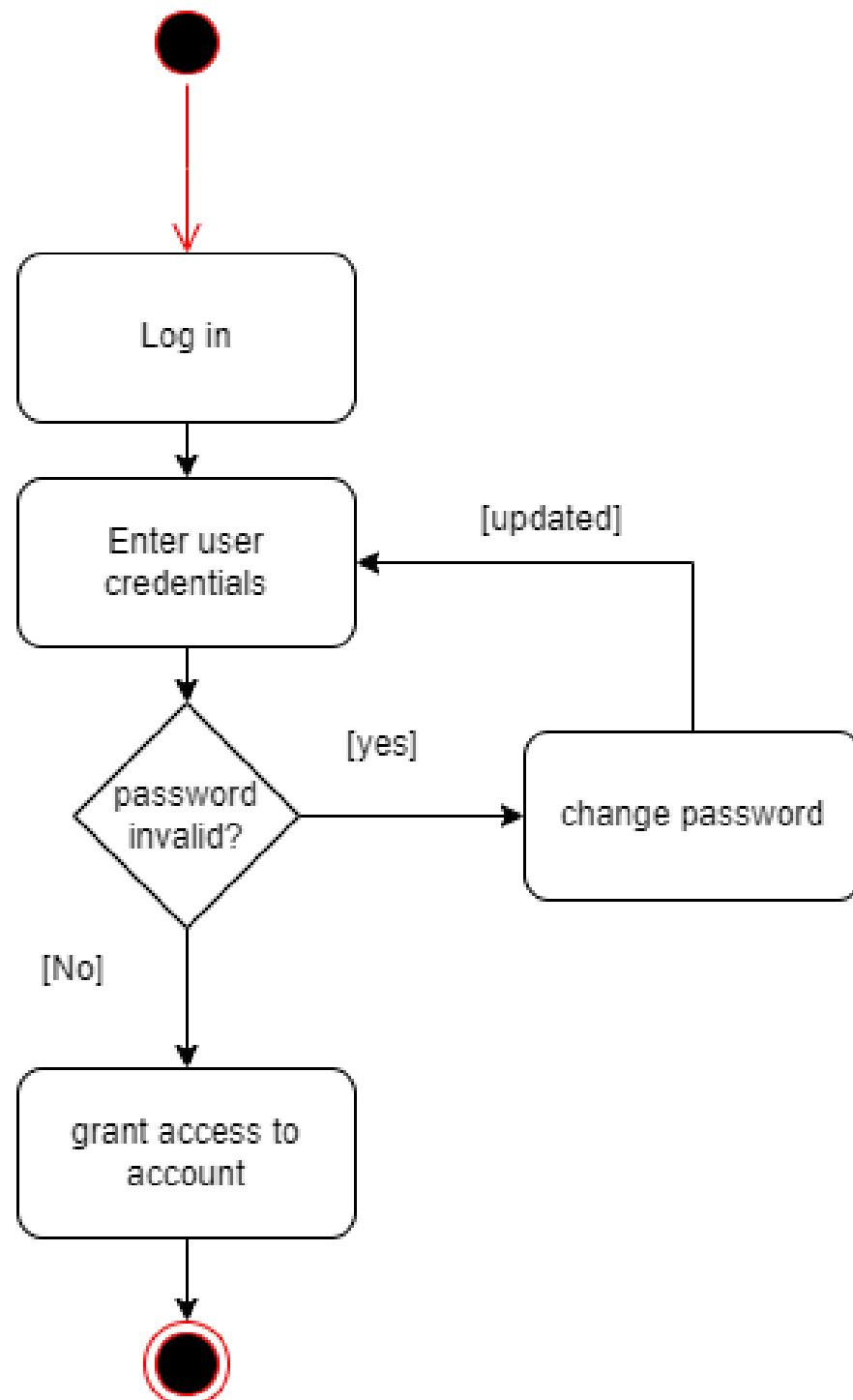
Sign up for VirtuFit account



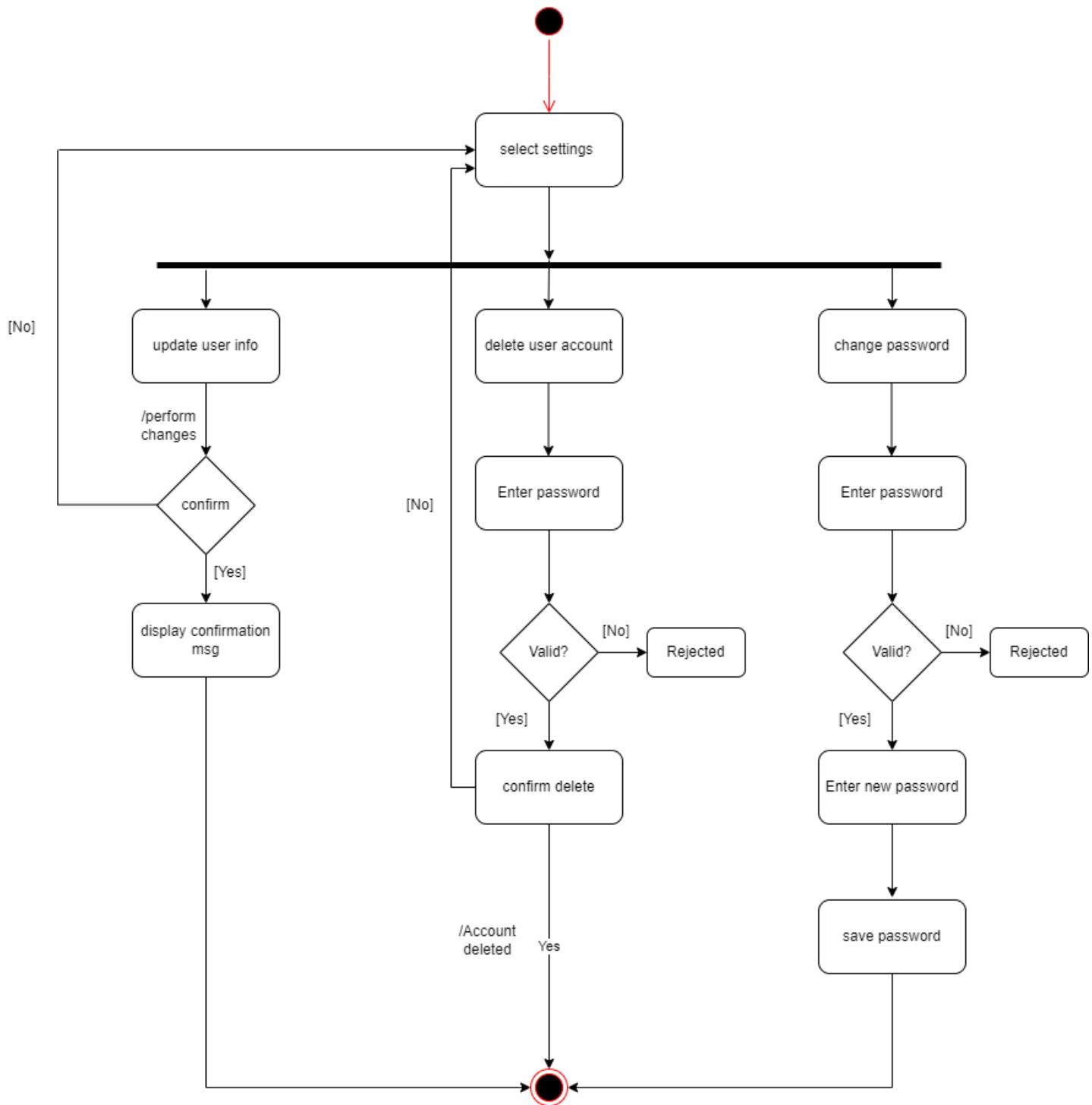
Login to VirtuFit account



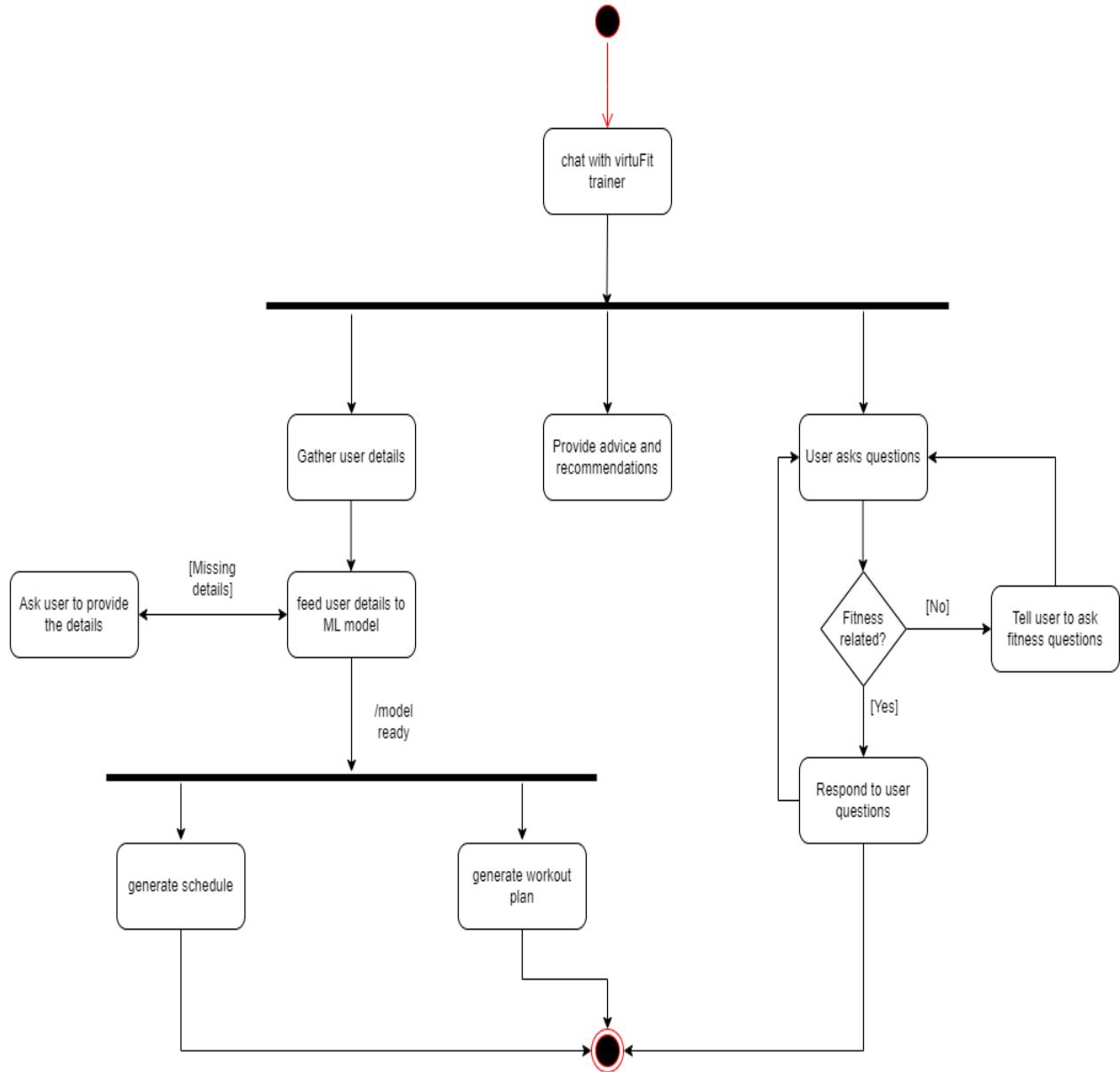
Forgot your Password



Settings

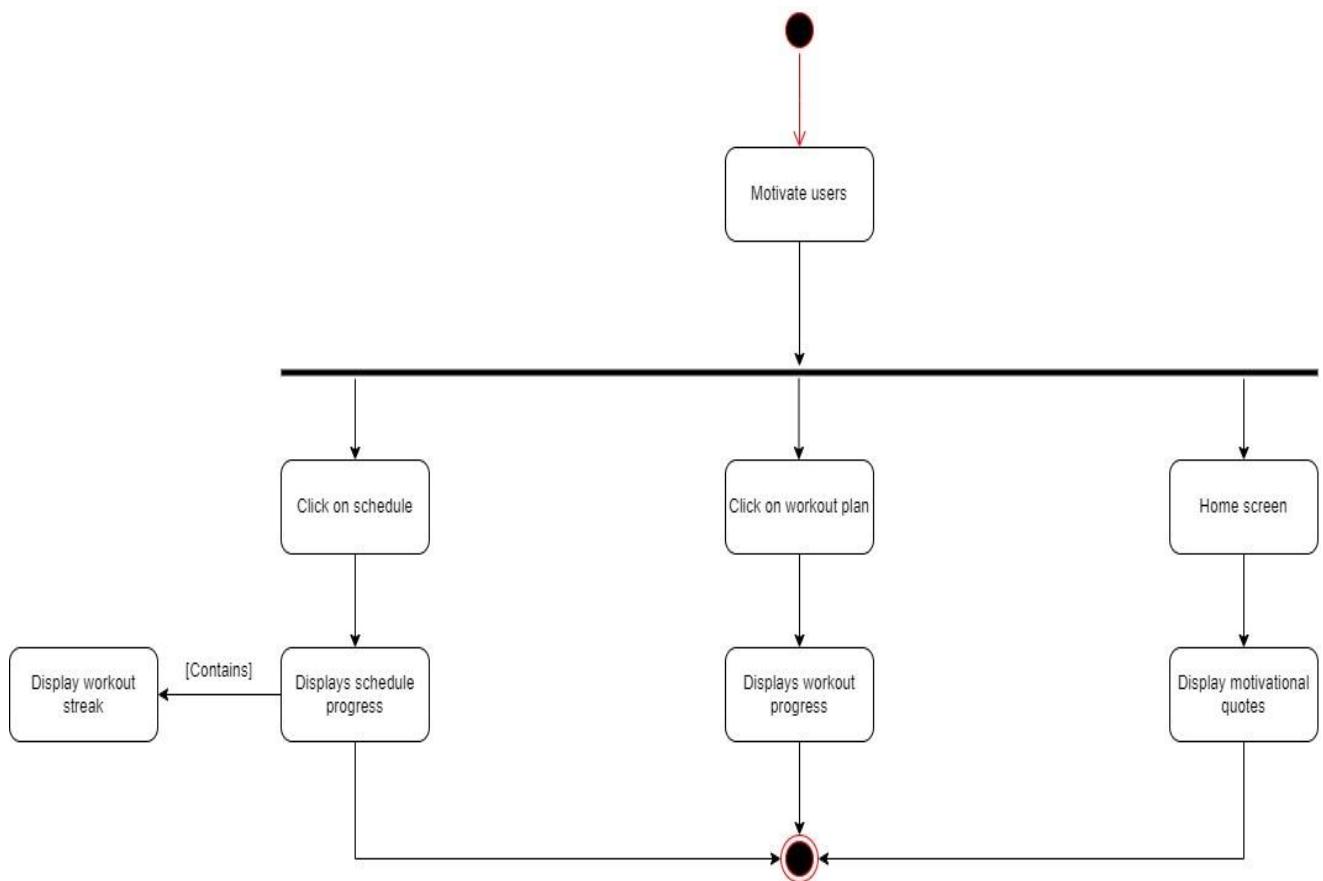


Chat with virtuFit trainer



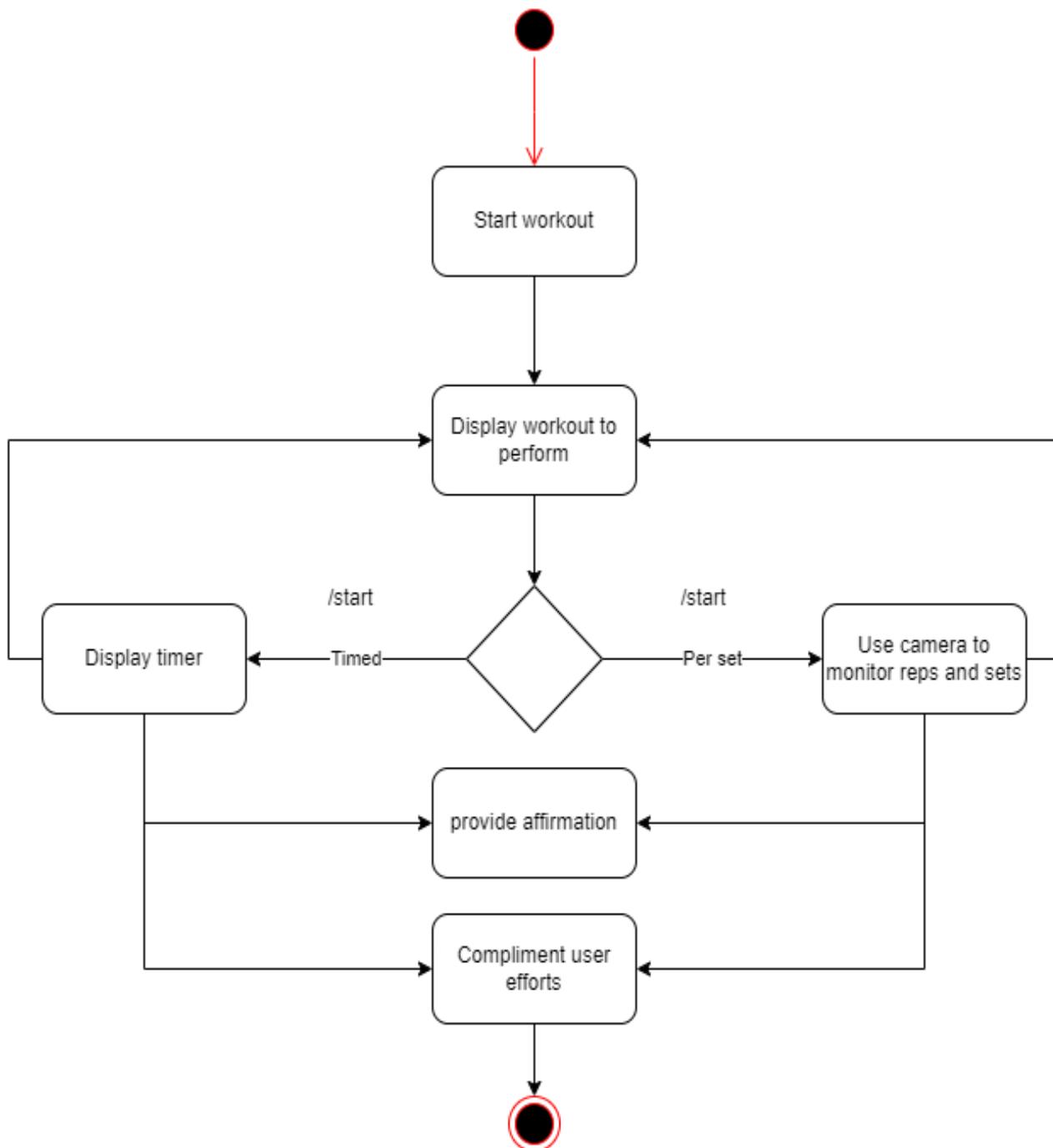


Motivate Users



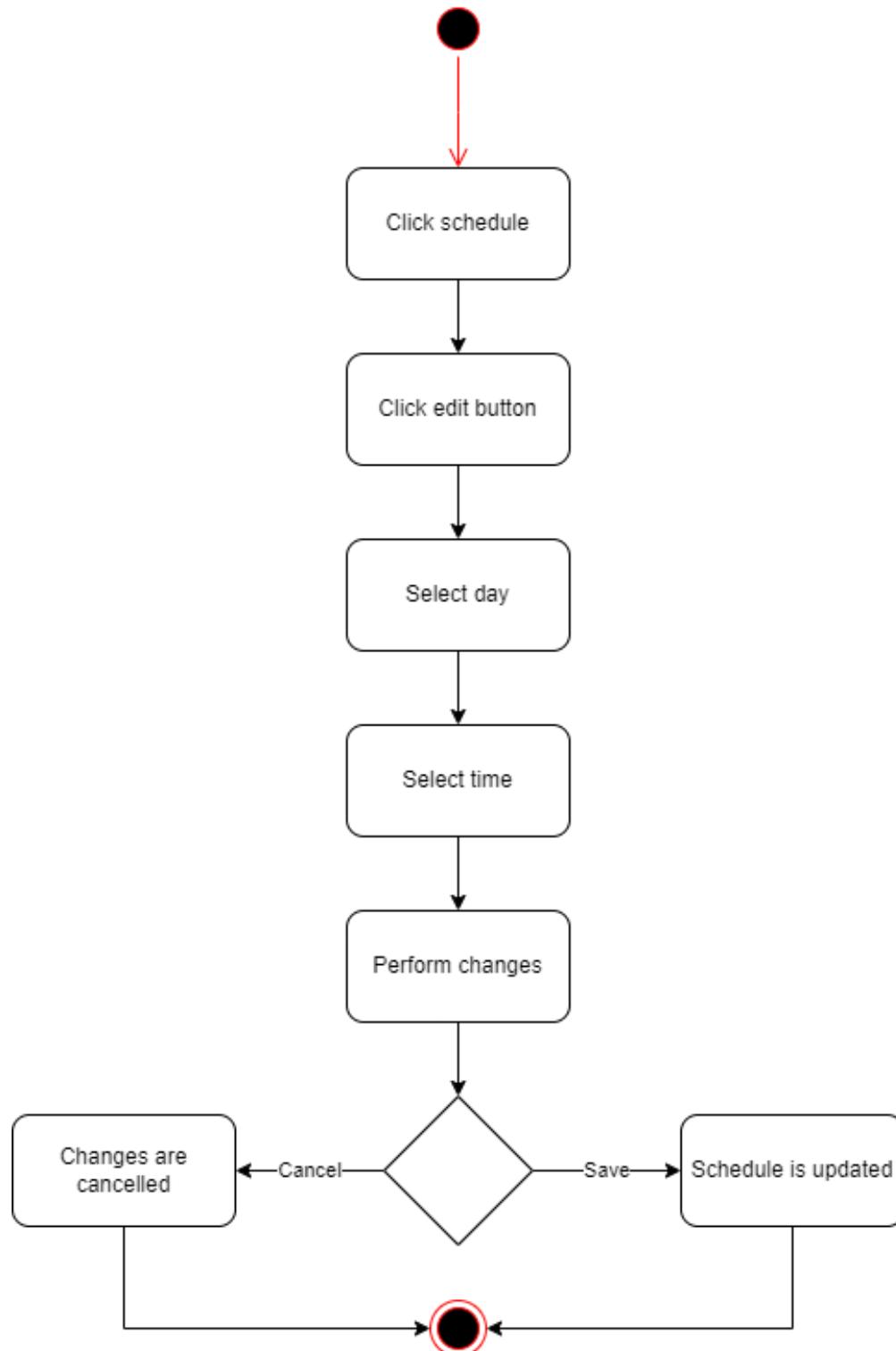


Start Workout

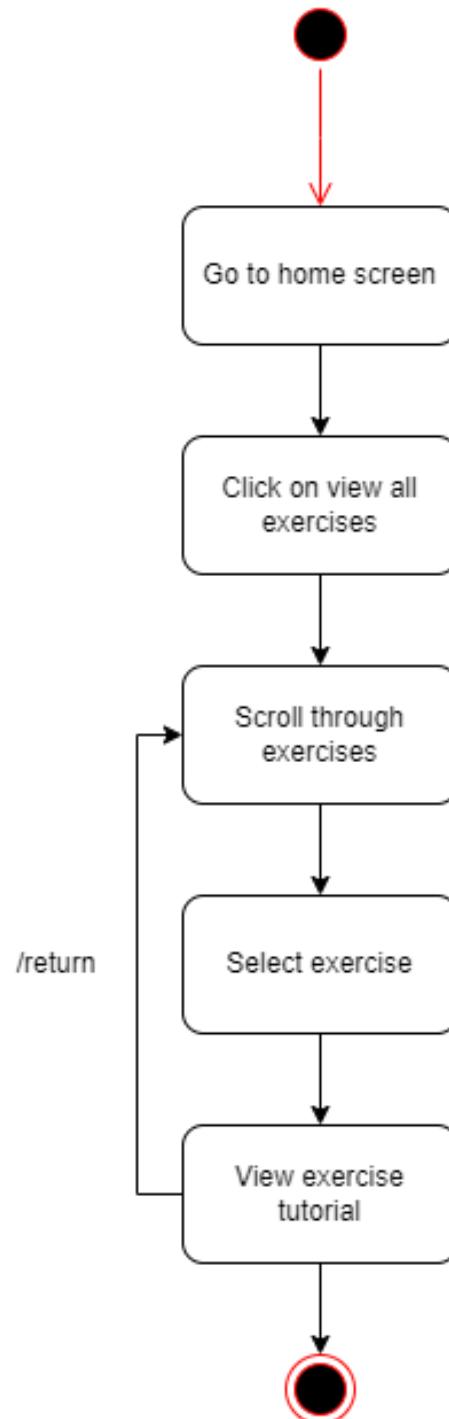




Edit Schedule

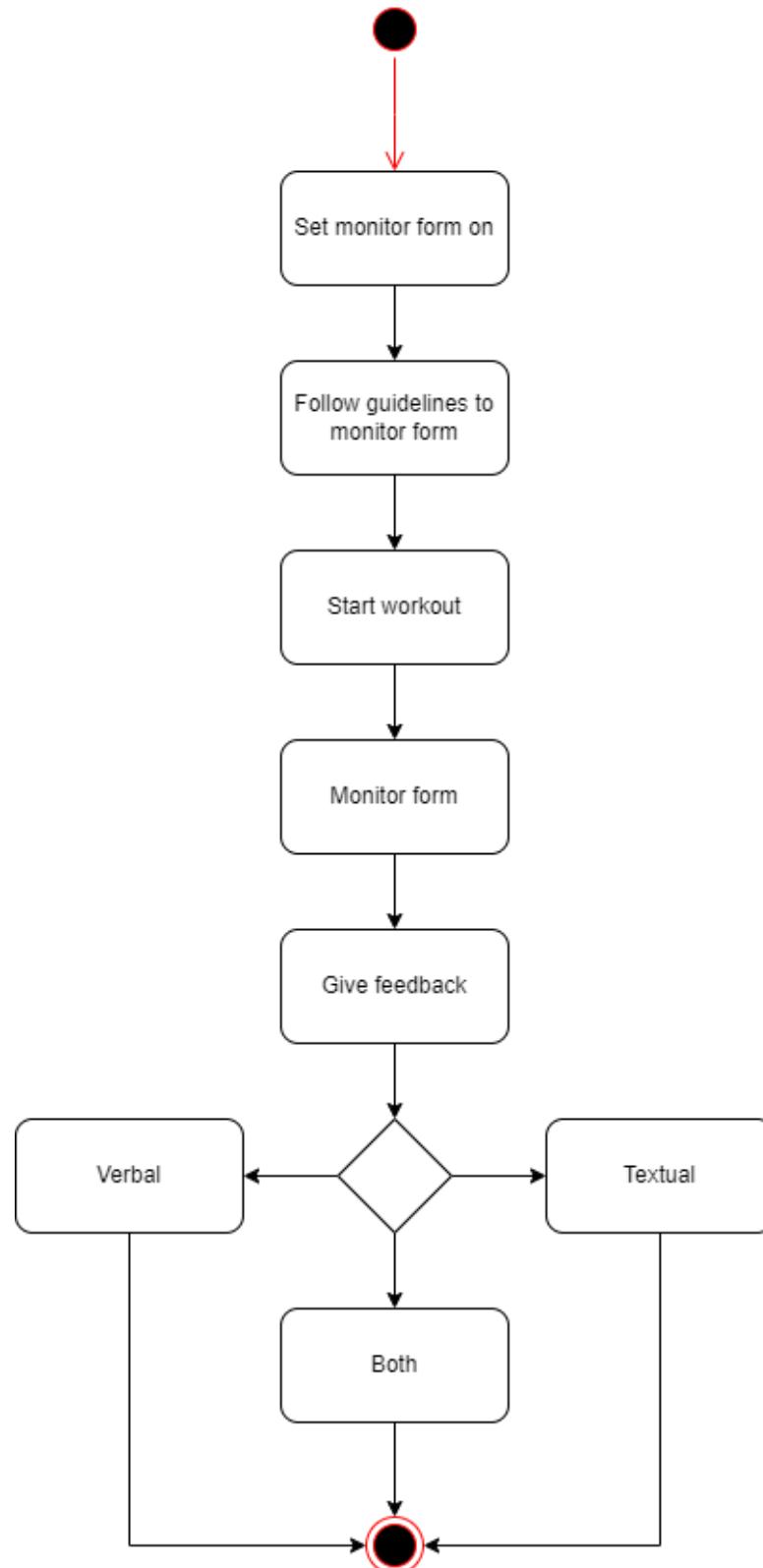


View Exercises and their tutorial



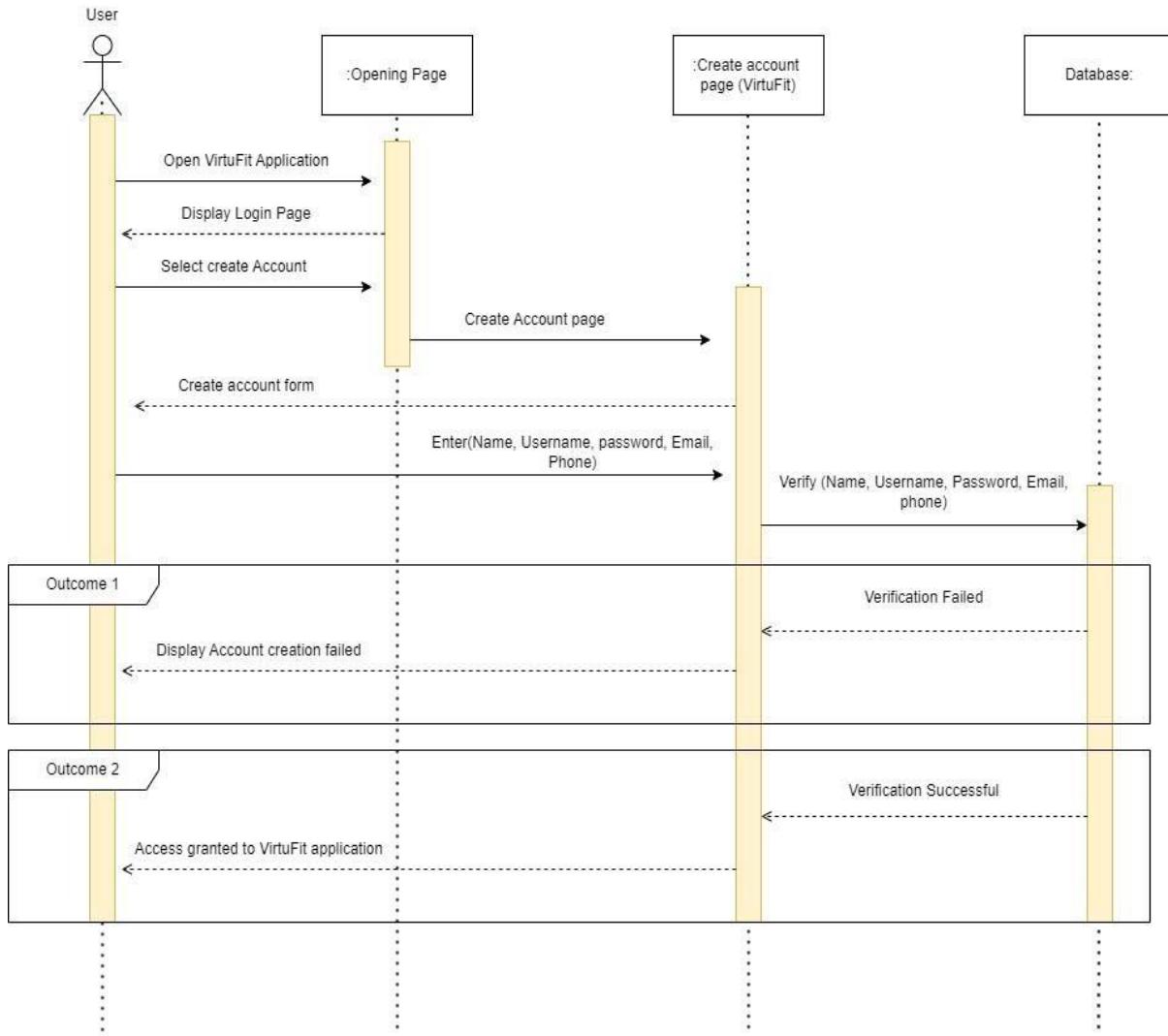


Monitor Form

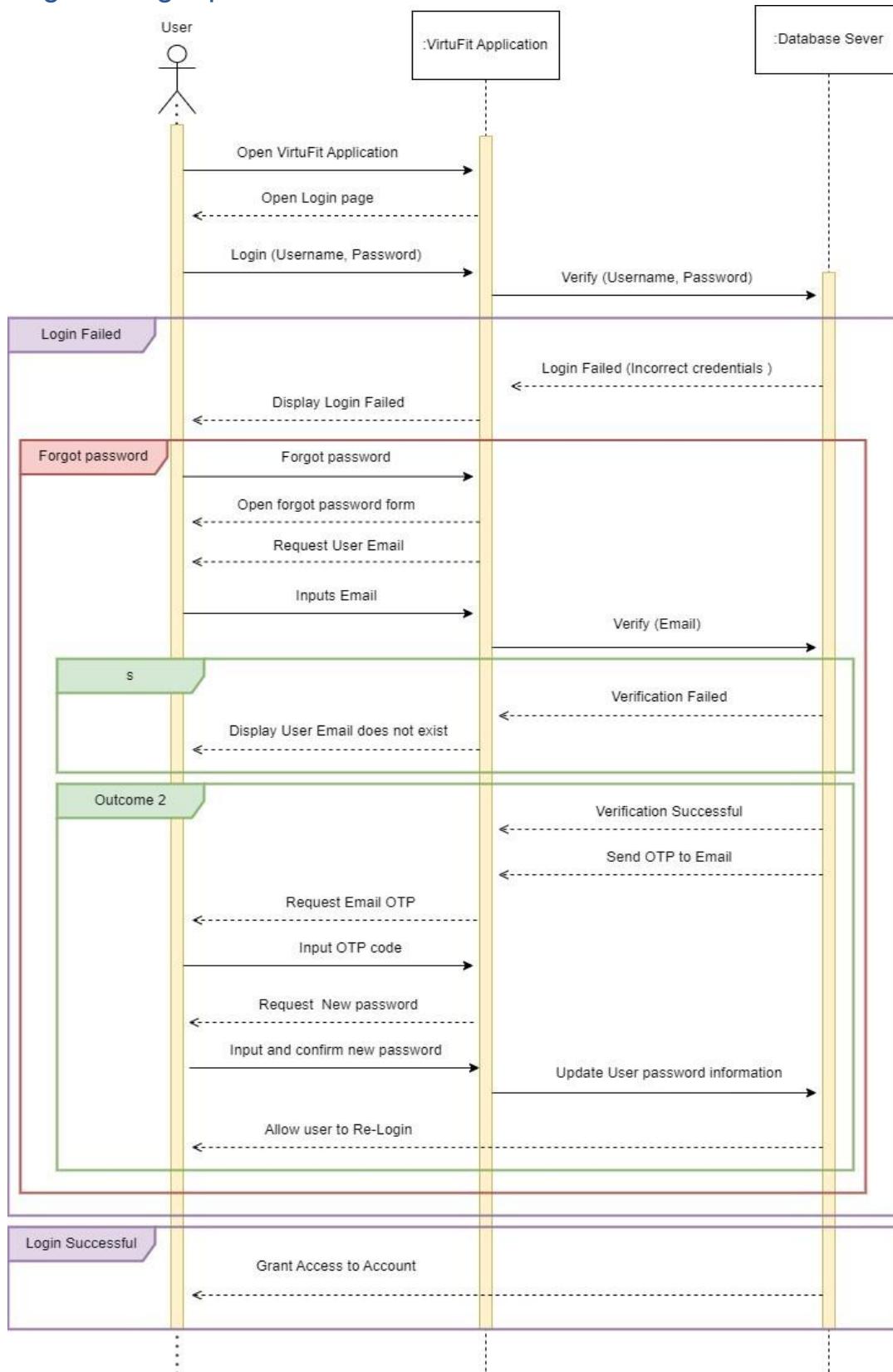


Sequence diagram

Create account

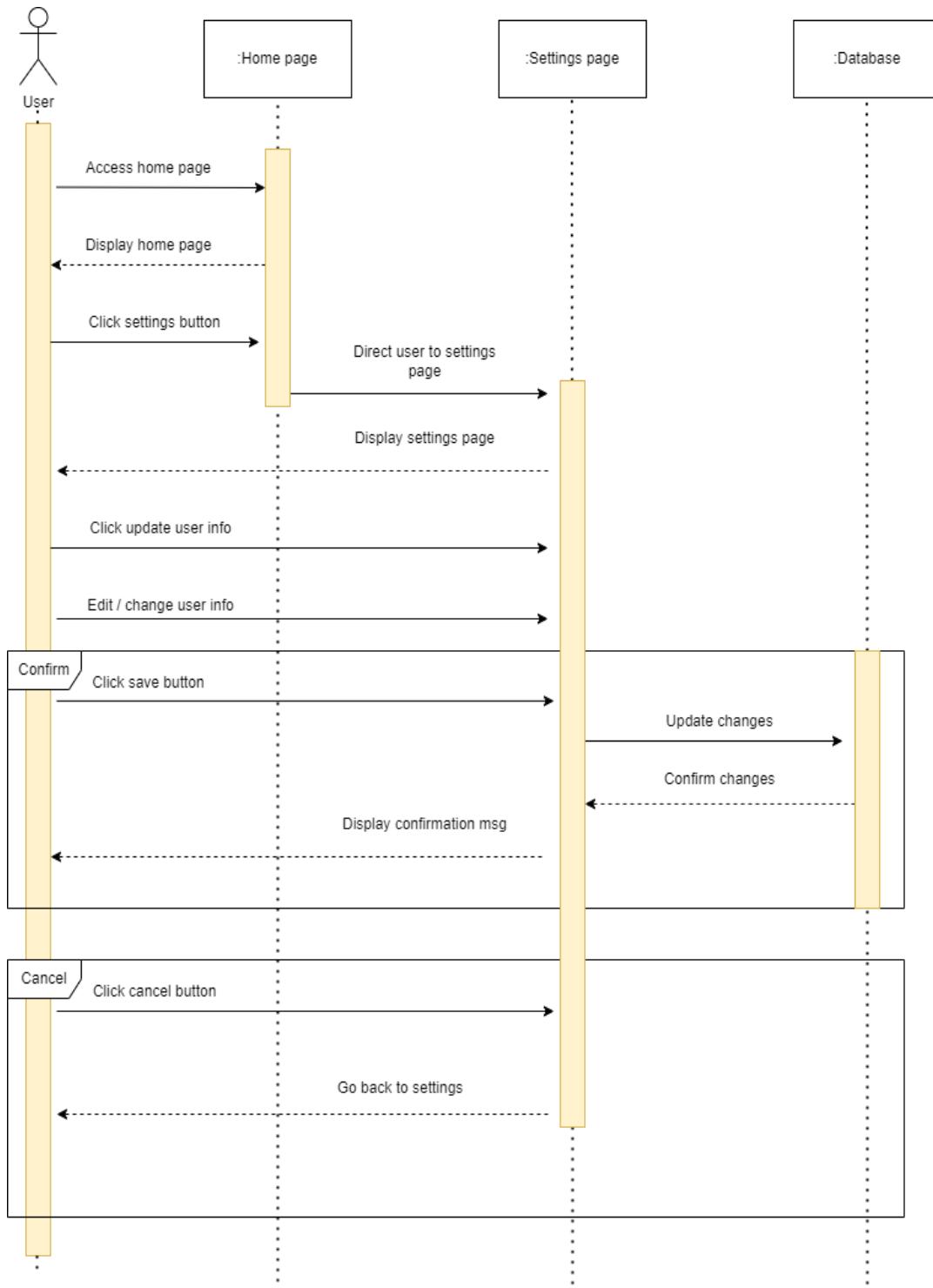


Login / Forgot password

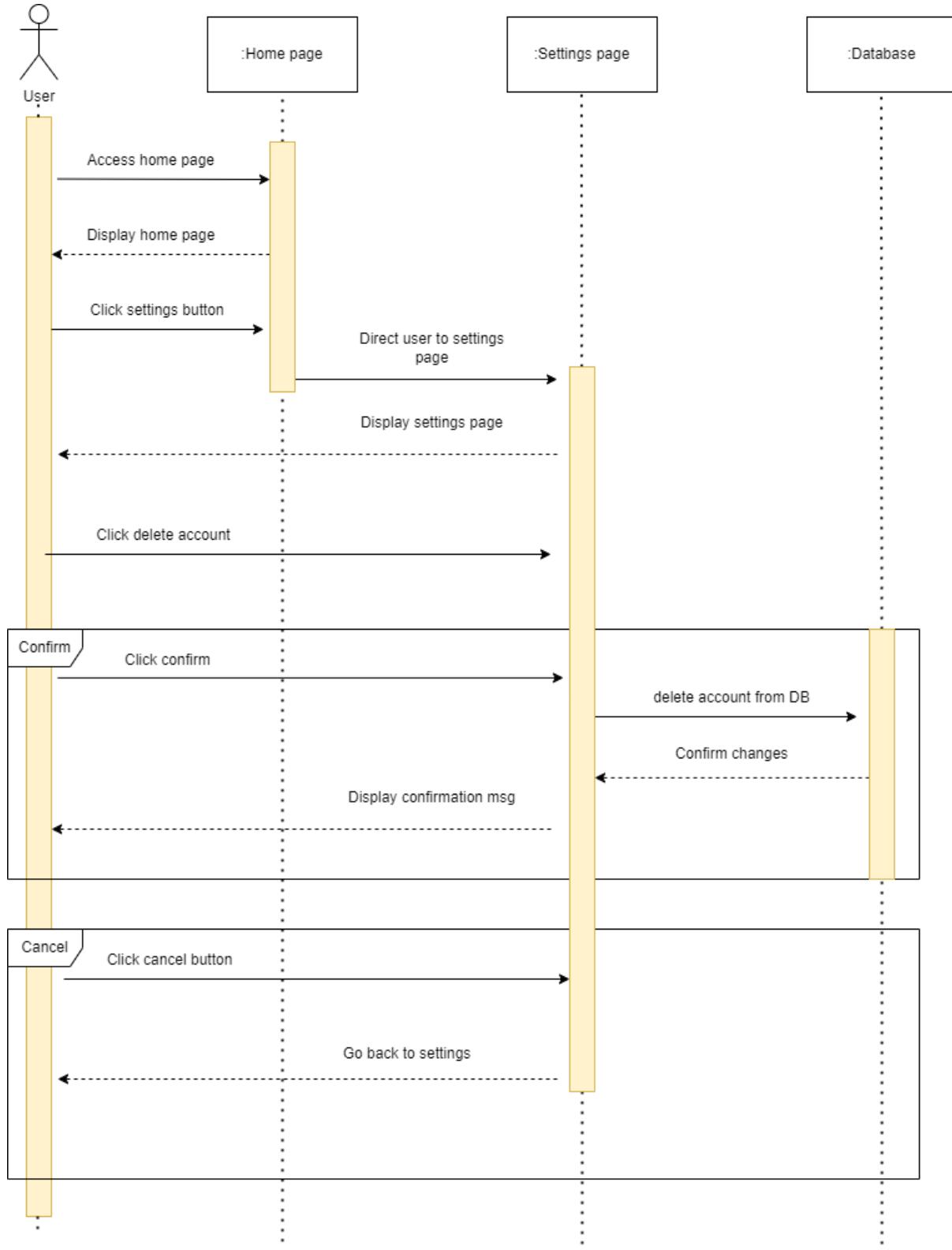


Settings

Update user information

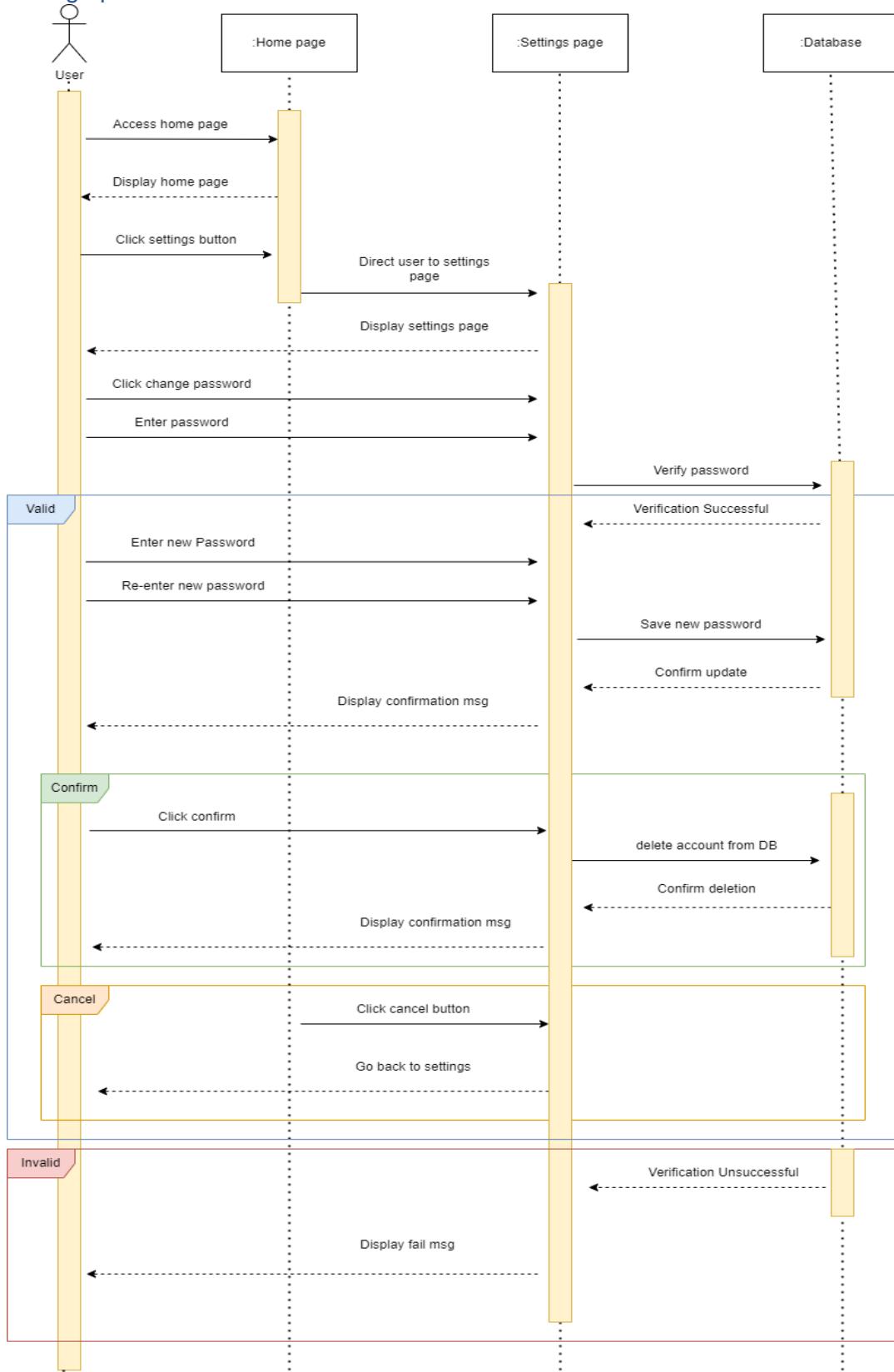


Delete user account





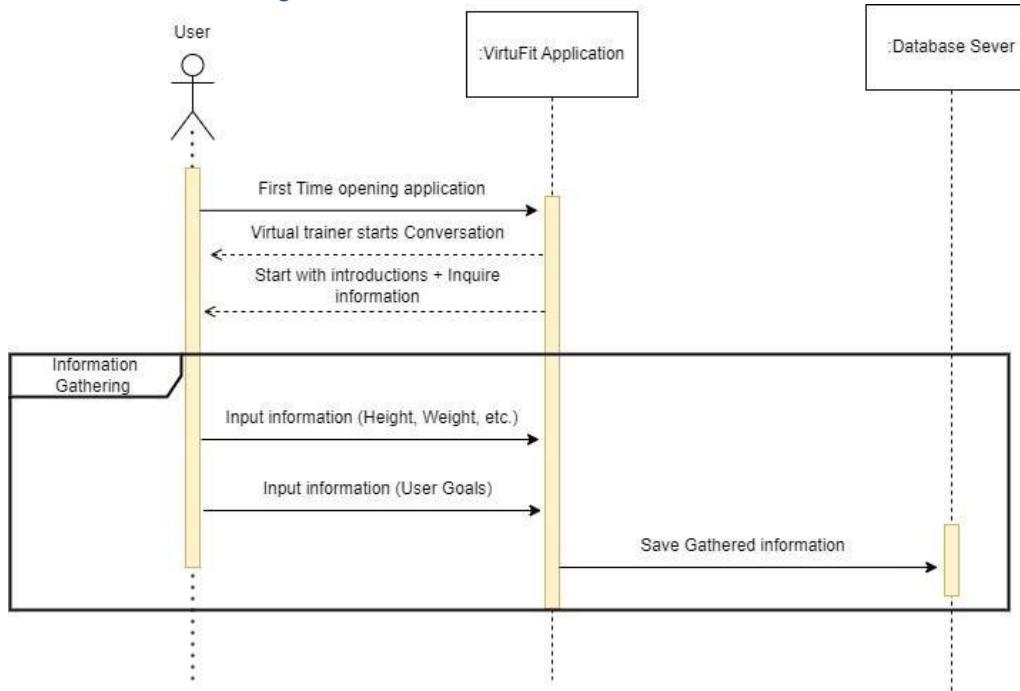
Change password



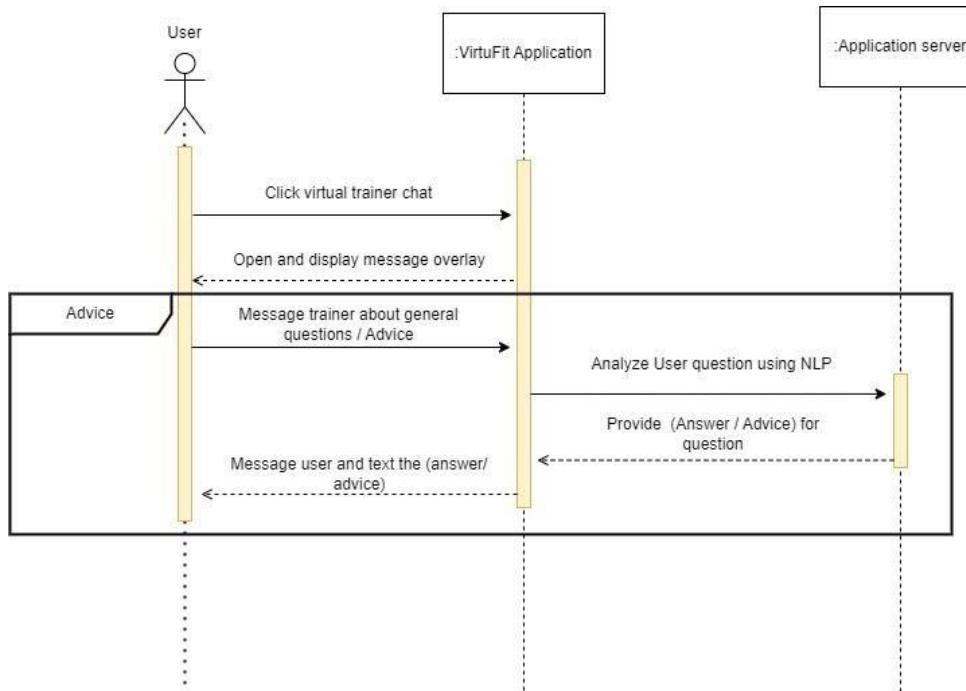


Chat with Virtual trainer

Information Gathering

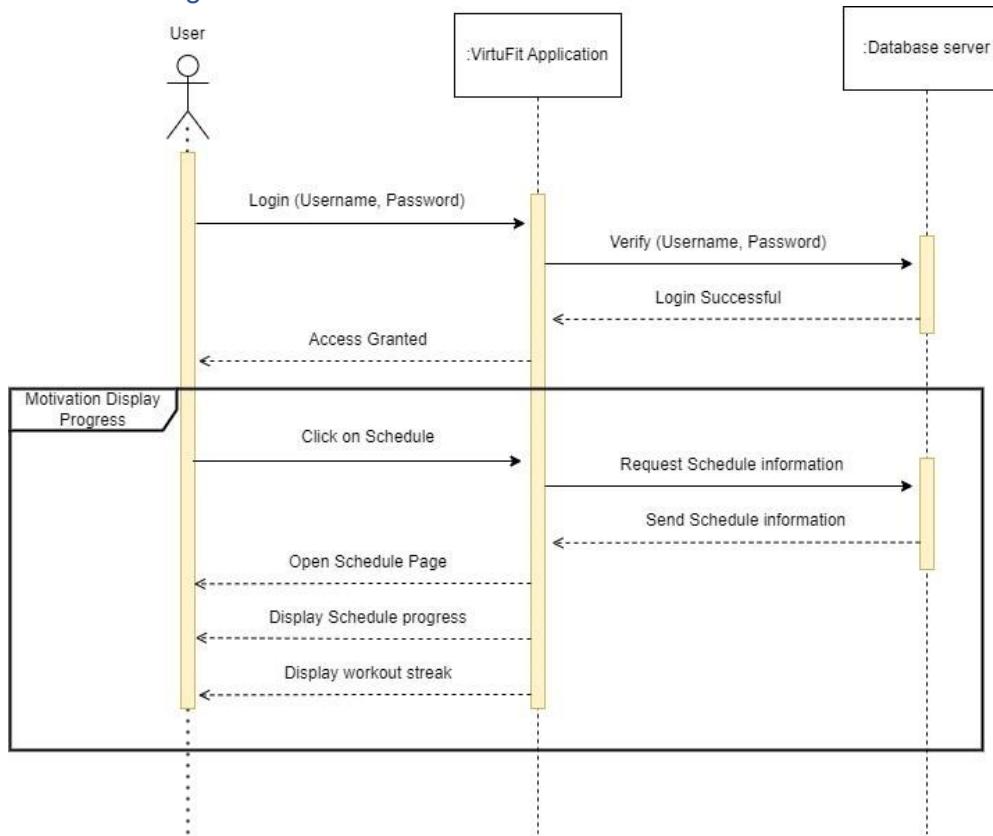


Advice/ Recommendations/ ask general questions

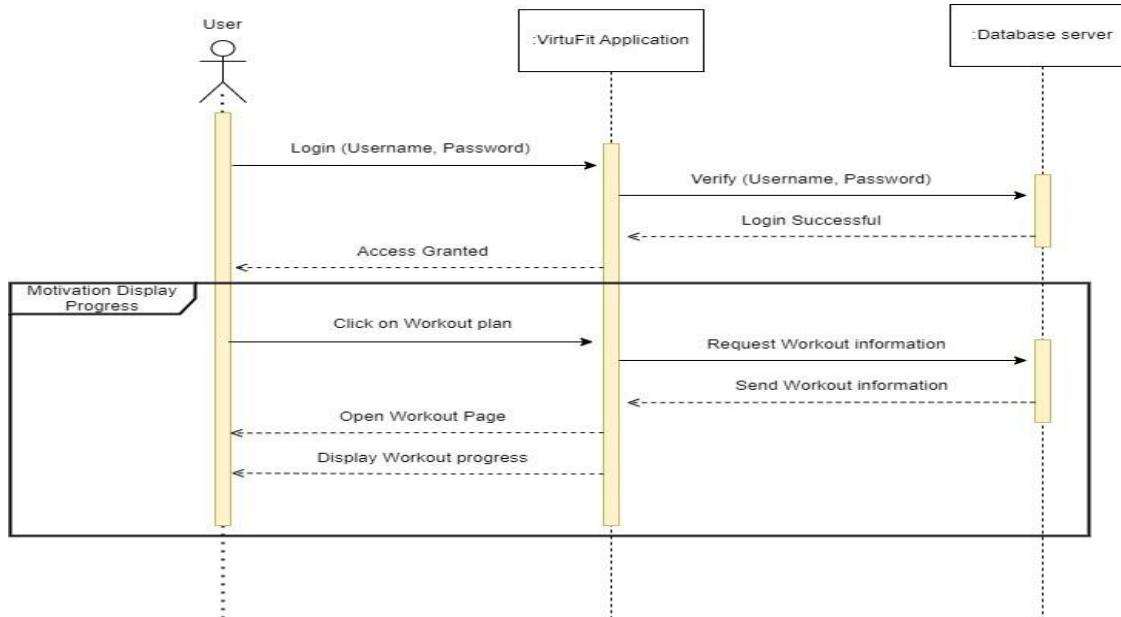




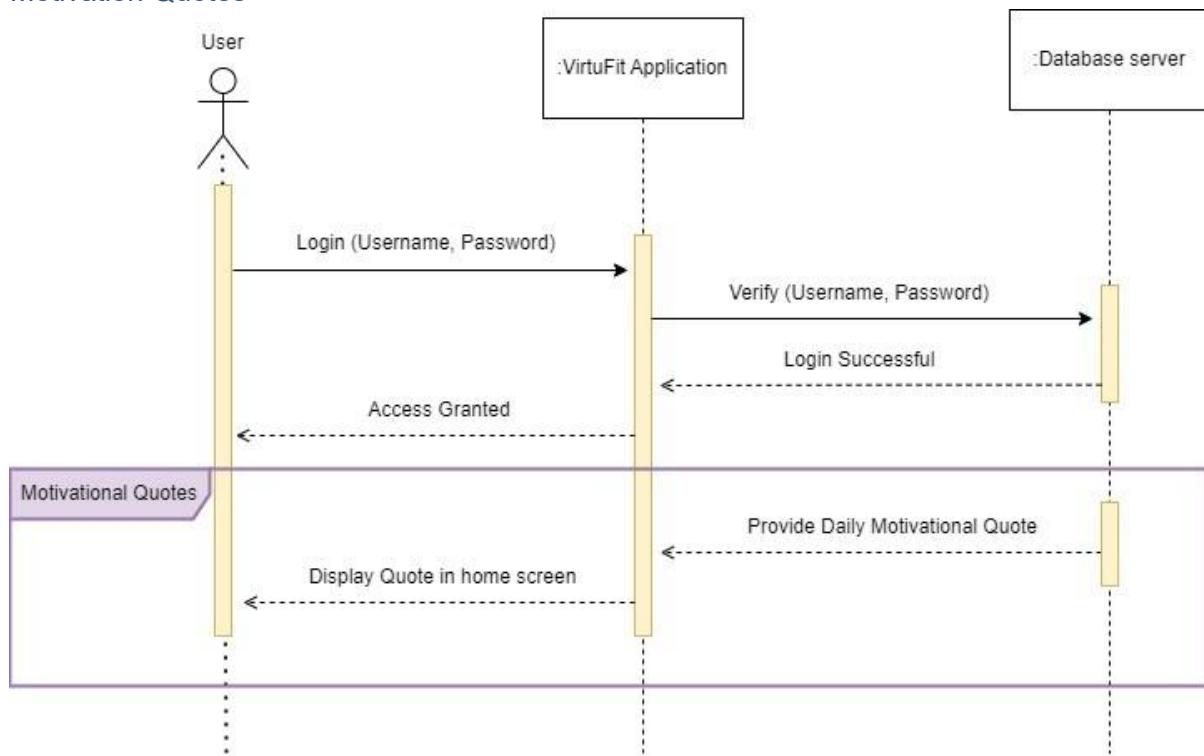
Motivate User Schedule Progress



Workout Plan progress

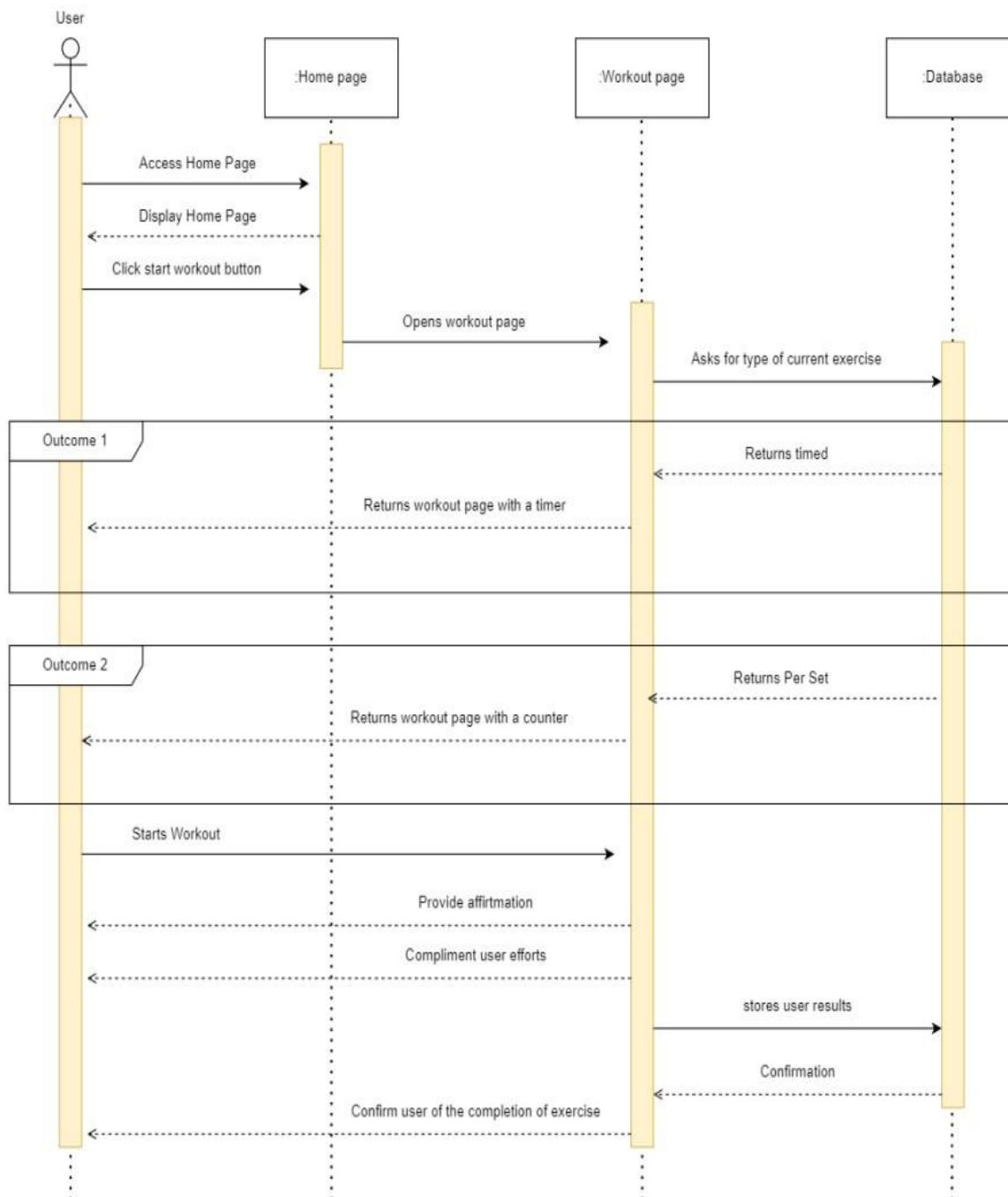


Motivation Quotes

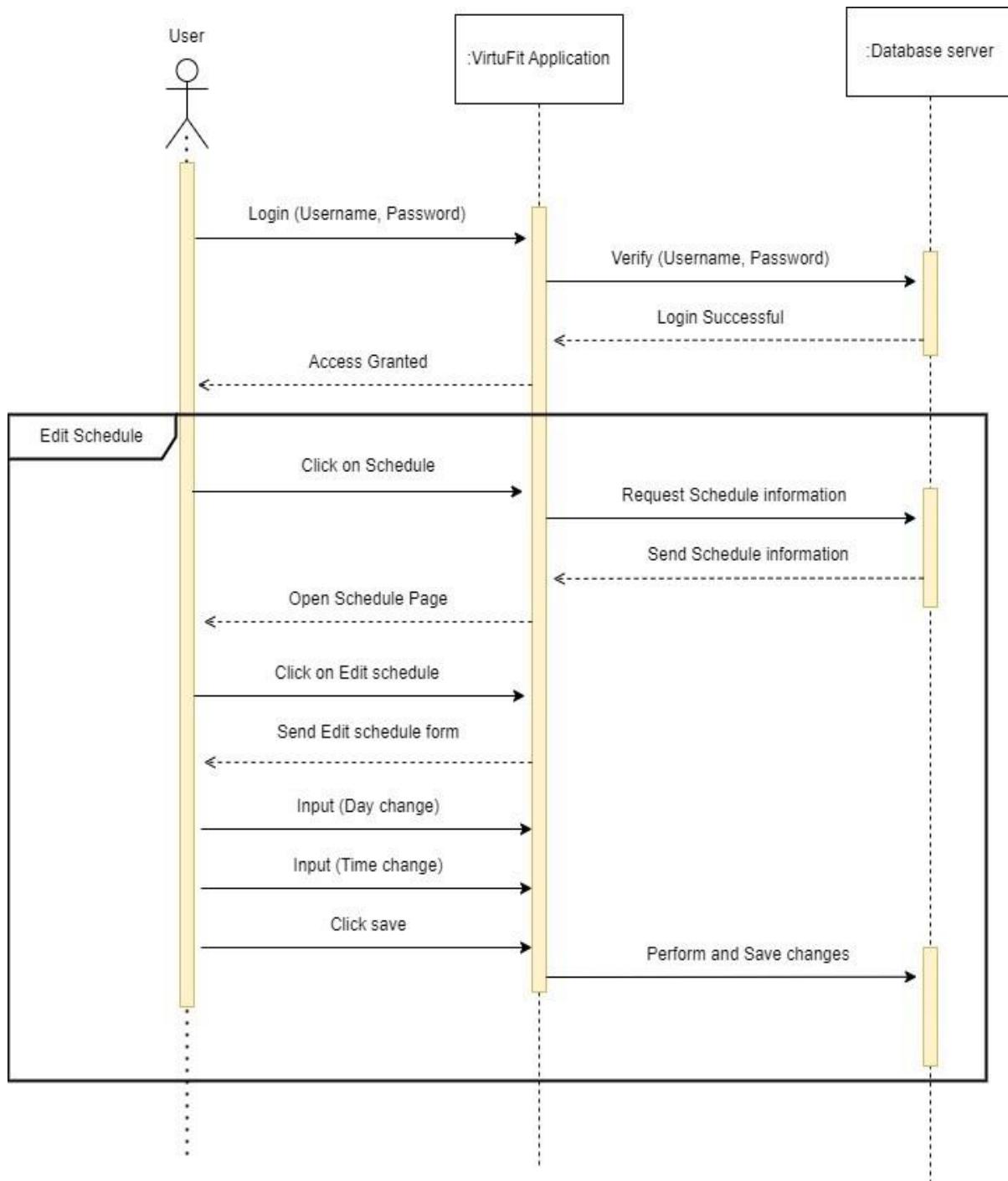




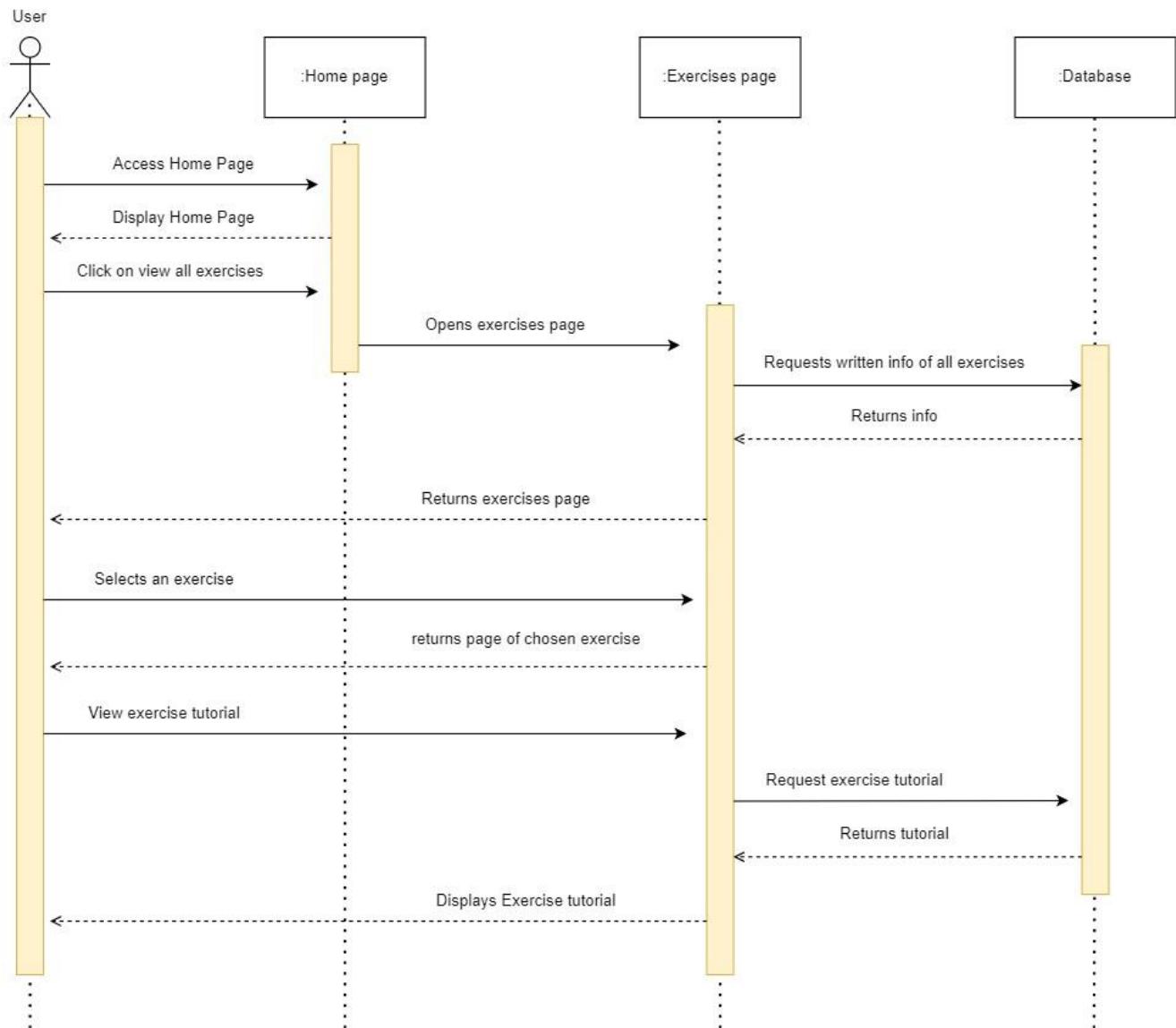
Start Workout



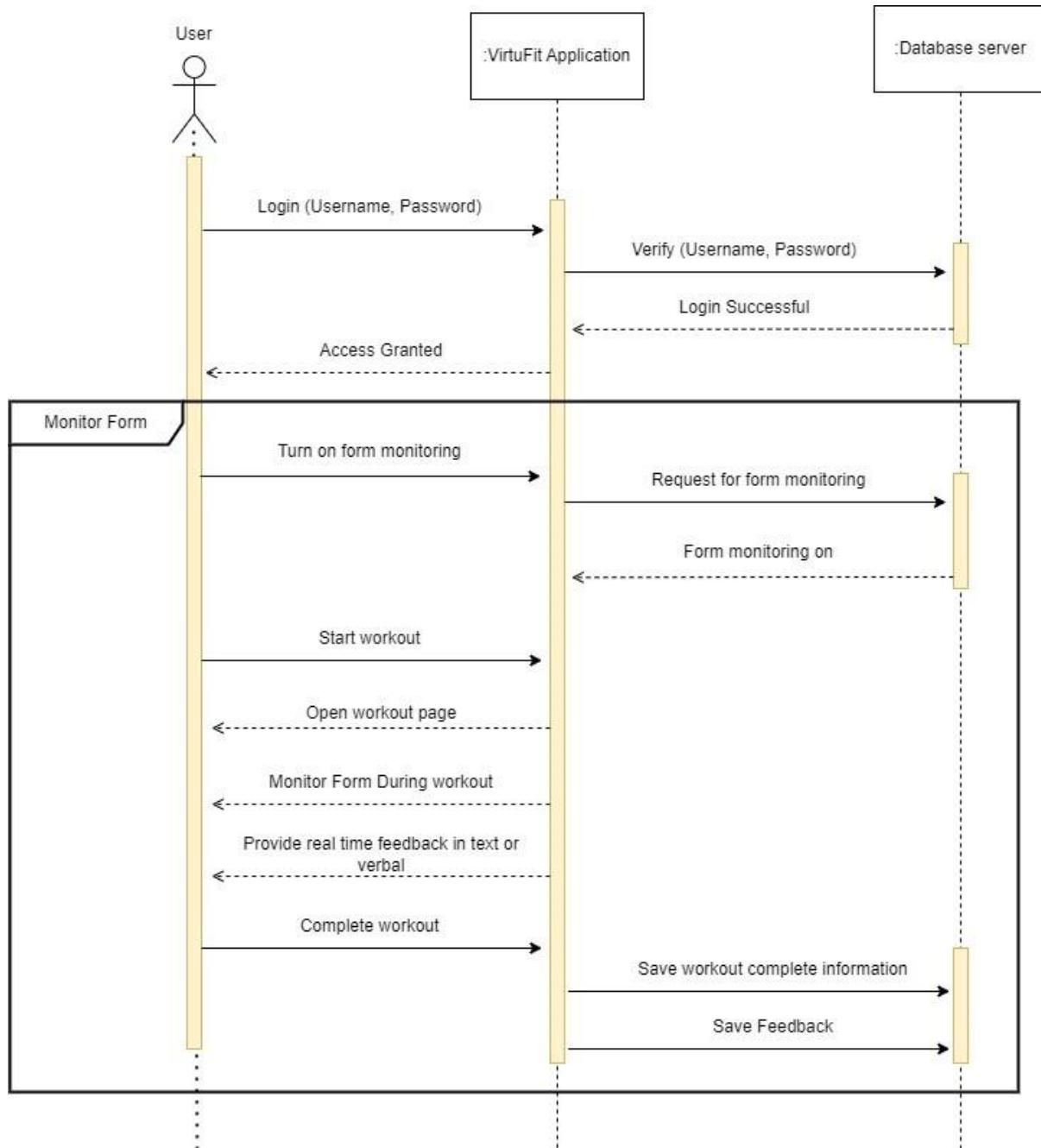
Edit Schedule



View Exercise and their Tutorial



Monitor Form



Human Interface Design

Overview of user interface

The Virtufit application currently consists of 15 screens. Each screen is made with thorough user application interaction consideration; minimal input yielding maximum user satisfaction.

Screen Images

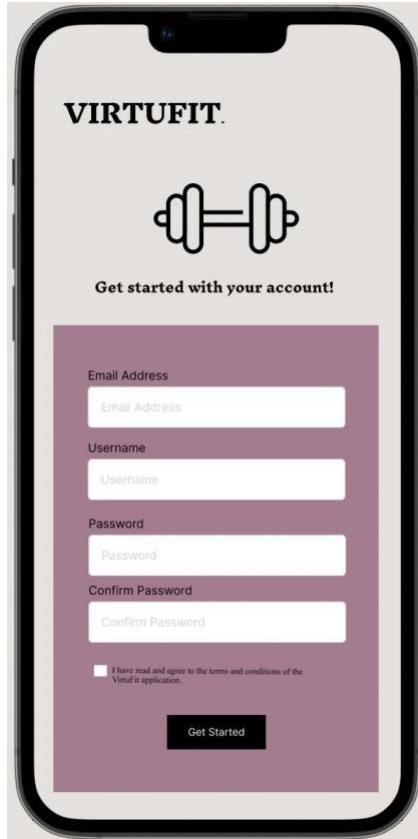
- 1- The Virtufit mobile application main screen, displaying two options for user:
sign up or Log in



2- Login screen



3- Sign up screen



4- In case the user forget their password, Forgot password screen



5- Users can choose which avatar they prefer as their personal trainer.



6- Once user log in they are greeting with home screen of virtufit.



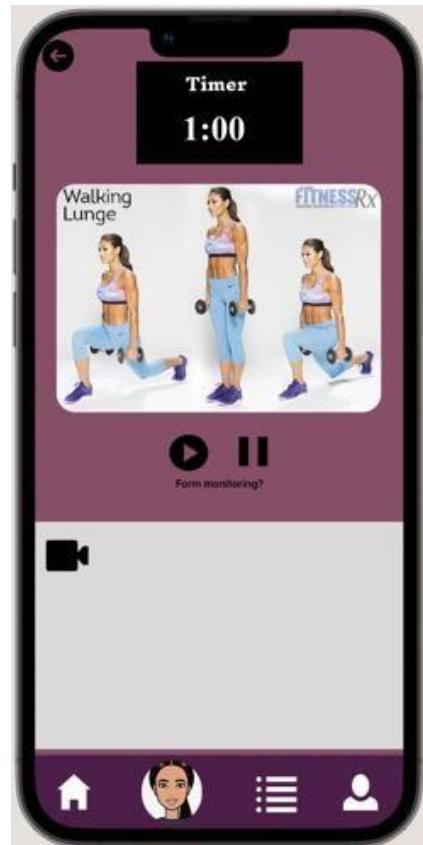
6.1- view schedule screen.



6.1. 2- edit workout plan



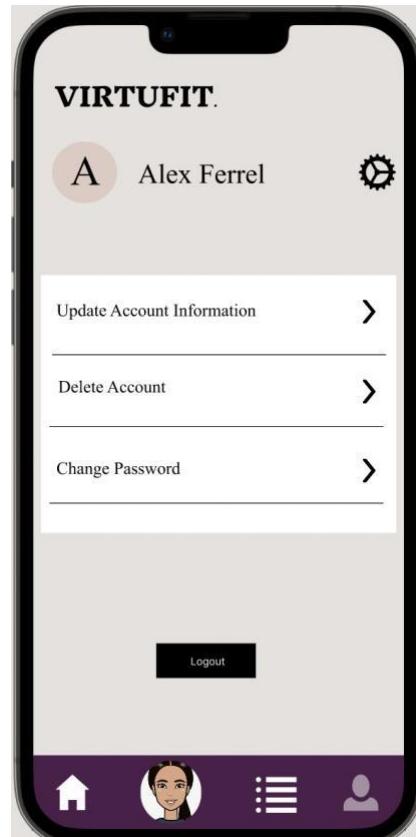
start workout



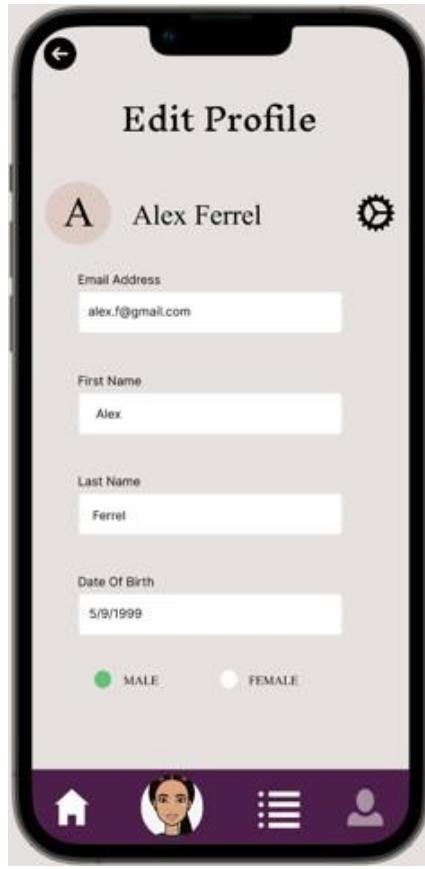
7- Virtual trainer chatbot



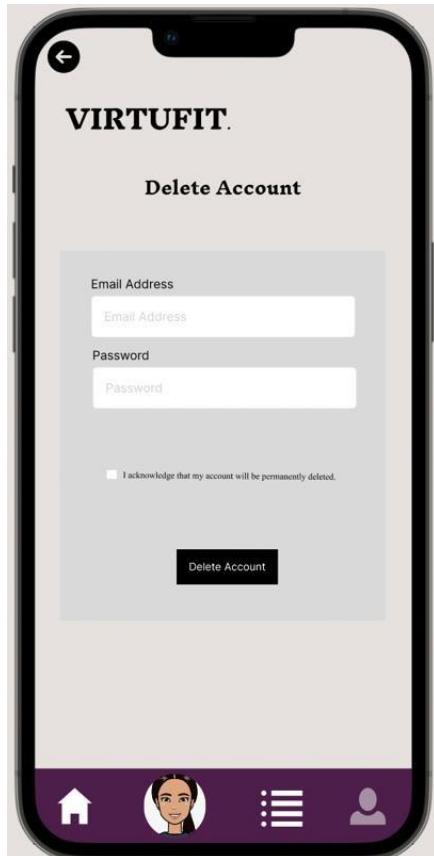
8- User Profile



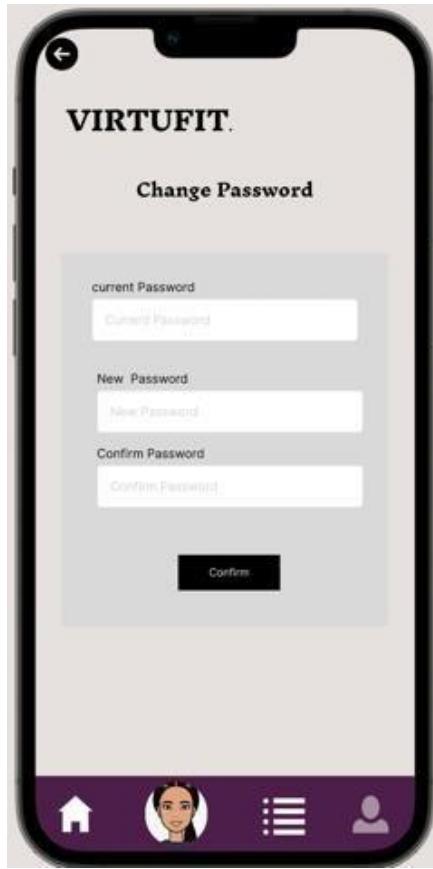
8.1- update account information



8.2- delete account



8.3- – change password



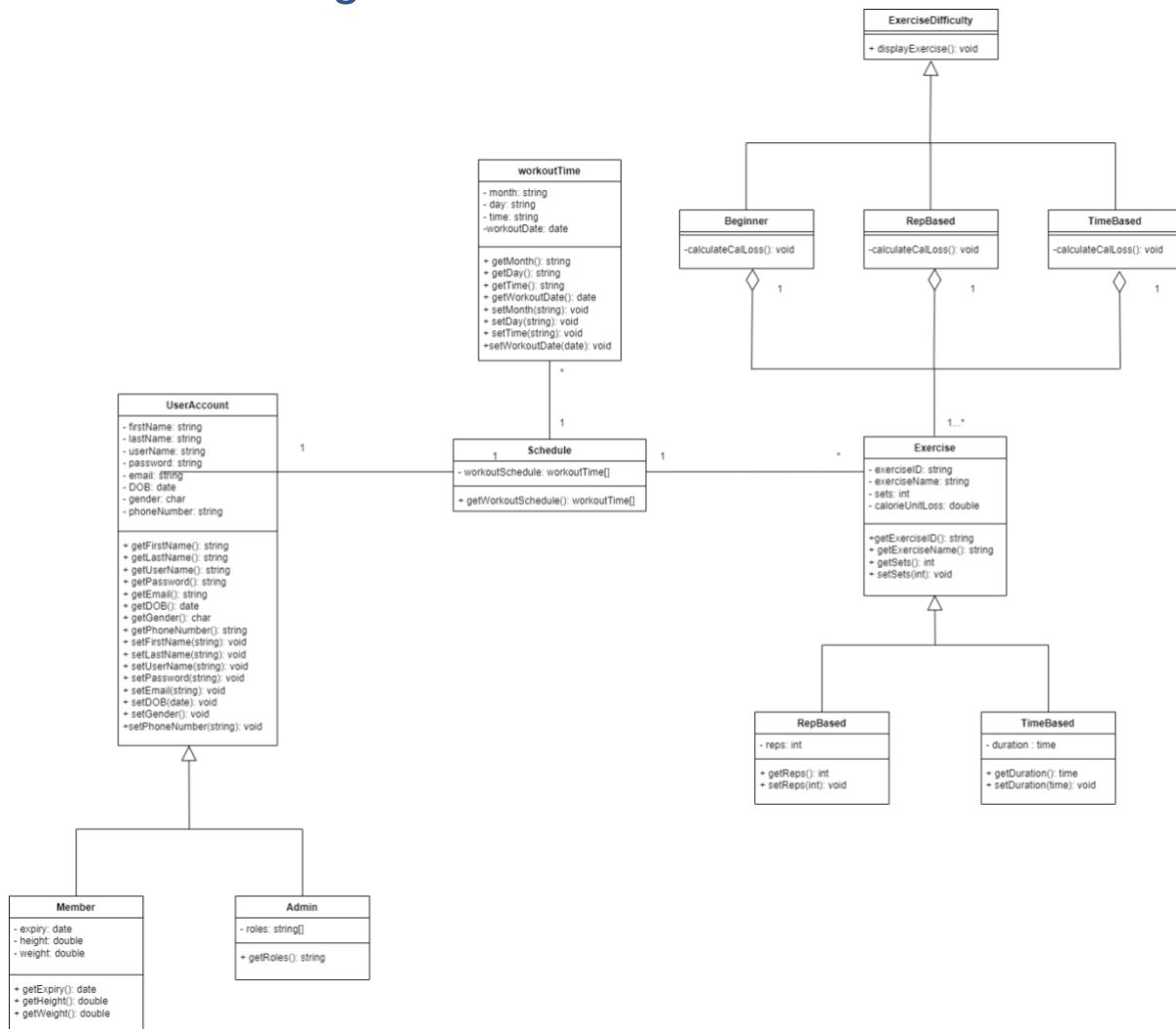
9- View all workouts



Requirement Matrix

Requirements Traceability Matrix							
Project Name:		VirtuFit	Document Author:		Team Blu		
Req #	Requirement Name	Requested/Approved By	Scope Deliverable or Feature	Scope Unique ID	Assigned To	Test Case	Current Status
1	VirtuFit Sign Up	Dr. Mai	Users need a way to give them a unique identifier on our backend database for CRUD operations.	11	Dani	Functionality	In-Progress
2	VirtuFit Log In	Dr. Mai	Users need to log in to access their unique identity and data.	12	Noor	Functionality	In-Progress
3	VirtuFit Settings	Dr. Mai	Users need to update their data and insert updated characteristics and settings.	13	Abdallah	Functionality	In-Progress
4	VirtuFit Chat	Dr. Mai	Users need to chat with the personal trainer AI assistance, VirtuFit	14	Ali	Functionality	In-Progress
5	Database Management	Dr. Mai	System needs a database to store sensitive and relevant data and content.	15	Fatima Aya	Back-End	Elaboration
6	Start Workout	Dr. Mai	Users need a way to start their workouts	16	Fatima Aya	Functionality	Elaboration
7	Real-Time Trainer	Dr. Mai	Users need a way to correct their poses and formations while working out	17	Abdallah	Functionality	Elaboration
8	Back-End Authentication	Dr. Mai	System needs an authentication mechanism to allow/block users from using the application	18	Dani	Back-End	Elaboration
9	Convenient User Experience	Dr. Mai	Users need to have a comfortable and soothing front-end experience	19	Noor	Front-End	Elaboration
10	Convenient Performance	Dr. Mai	System needs the suitable allocation of hardware resources to run smoothly	20	Abdallah	Back-End	Elaboration

UML class diagram



Test Case Document



CSCI321

Test Case Document

Team Blu

Dani Ternanni, Fatima TchanTchane, Noor
Ahmed, Ali Mela Ali, Abdullah Rafiei

Introduction:

This document provides an in-depth examination of three primary features essential to our application: Chat Bot, Form Monitor, and Scheduling. The Chat Bot, developed using Python and integrated with the OpenAI library, leverages the "Completions" endpoint to assist users in real-time, offering immediate answers to their queries and enhancing their overall platform experience. The Form Monitor, crafted using JavaScript and employing the human pose library in React Native, goes beyond mere tracking. It analyses user gestures for a more interactive and responsive user experience, with the system providing accurate feedback based on user movements. Lastly, the scheduling employs a combination of procedural programming and rule-based algorithms to provide users with a tailored fitness schedule. By understanding individual user needs and patterns, it suggests fitness routines that align with personal goals and timelines, ensuring users receive an optimized plan for their health objectives.

Test Plan:

Test plan Verification Validation (V & V):

During the development of our fitness application, it's essential to undertake various testing activities to ensure that the features meet users' fitness needs and align with their expectations. Implementing attentive checks early in the developmental phase allows us to identify any potential challenges, which is vital for the effectiveness and efficiency of the application. For validation, we have instituted unit testing measures, ensuring that when users log in, their credentials are authenticated correctly. For verification, aside from unit testing, we employ user acceptance testing. This ensures that after logging in, users can seamlessly converse with the chatbot, access their personalized workout schedule, and be guided on their fitness journey. A standout feature is the incorporation of HumanPose technology, which actively detects and analyses user movements during at-home workouts. This ensures that users maintain the correct posture and form, optimizing their exercise benefits while minimizing the risk of injury.

Test plan Schedule:

To ensure that the VIRTUFIT application meets requirements from the developer end to be delivered to users, Team BLU is determined to conduct various tasks and tests, both completed and ongoing. Each phase is designated a specific period within their respective environments to guarantee accurate and satisfactory results.

The table below exhibits the breakdown of each task and testing performed, with the agenda it tackles, and an estimation of time spent on each of them.

Task name	Task description	Estimated time
Examining requirements	evaluating the application's needs, front end and back end, for well-ordered development.	14 days
Setting test environment	assembling pertinent data about development resources to choose the tools that are ideal for our application environment	25 days
Unit testing	Testing individual modules to ensure their functionality meets the expectations of VIRTUFIT envisions.	14 days
Integration testing	Modules from different environments are integrated together with the application's front end as one complete model.	20 days
Acceptance testing	To ensure all requirement modules are completed and function up to their requirement	10 days

Test plan Requirements Traceability:

Requirement ID	Requirement Description	Test-case ID	Test-Case Description	Test Results
1	Sign up to VirtuFit and create a user account	T1, T2, T3	Testing that a user can successfully create a valid account meeting the minimum sign-up requirements	T1(Passed) T2(Passed) T3(Passed)
2	Log in to VirtuFit account	T4, T5, T6	Testing that an existing user can log in to their account successfully with the correct username and password	T4(Passed) T5(Passed) T6(Passed)
3	Converse with virtual trainer(chatbot)	T7, T8, T9	Testing that a user can successfully converse with the chatbot and receive accurate responses	T7(Passed) T8(Passed) T9(Passed)

4	Create User schedule and workout plan	T7, T8, T9, T10, T11, T12	Testing that a user can successfully provide details to chatbot to be extracted and used to create personalized schedule and workout plan	T7(Passed) T8(Passed) T9(Passed) T10(On Going) T11(On Going) T12(On Going)
5	Navigate through all available exercises	T13, T14	Testing that a user can view and navigate through all exercises and learn about them	T13(Passed) T14(Passed)
6	Start workout session	T15, T16, T17	Testing that a user can start their workout session and proceed with it to the end of the session	T15(On Going) T16(On Going) T17(On Going)
7	Monitor Form	T15, T16, T18	Testing that a user can monitor their form while working out and receive real-time feedback.	T15(Passed) T16(Passed) T18(Passed)

8	View and Edit Schedule	T19, T20	Testing that a user can view and edit their schedule	T19(On Going) T20(On Going)
---	------------------------	----------	--	------------------------------------

Test plan Test recording procedures:

After careful evaluation and examination, team Blu made the decision of using two important test recording procedures: CFG (control flow graph) diagram and def-use analysis. The inclusion of these recording procedures provides a clear overview of the code to implement crucial key features of the application, aiding in improvement of code and managing bugs and errors otherwise overlooked.

The decision of using CFG diagrams is to assist testers in locating all potential paths of the executional code and analyzing every decisional aspect of code. Moreover, the graphical representation of each node signifying the true and false scenario of the code, makes it easier to visualize the code and understand all potential cases being considered.

Secondly, the def-use analysis provides a clear view of the variables used within the code, providing an overview of the path from the point of initialization of a variable until when it's used. Furthermore, by doing so highlighting if there has been an inclusion of a variable which is not used or not initialized. The overall idea of def-use analysis has made it conveniently easier to locate errors and bugs within code which are easy to overlook with hundreds of lines of code.

Test plan Hardware and Software:

The VirtuFit mobile application is built upon various software and hardware components, combined to produce an excellent user-friendly end product. Among the various platforms available, after careful evaluation and examination of the overall structure of the application, the VirtuFit front-end is developed on the React Native platform. React Native is a developer friendly platform that deploys the UI on Android devices.

Secondly, Team BLU is making use of a SQL based database to store required user information, which will be derived to create personalized user schedules and workout plans. Moreover, to do so, the use of procedural programming and rule-based coding is integrated to achieve matchless, customized results varying from user to user.

A crucial element of the application is the monitoring form during user workouts to ensure precise user assessment. Team BLU utilized the HumanPose library for this

purpose. Furthermore, to derive user information via a humanized chatbot, the implementation of the OpenAI API can be viewed on the chatbot screen. Lastly, to further aid in the derivation of specific data from the user in the chatbot, to avoid overloading the database, NLP will be implemented.

Moreover, in terms of the necessary hardware requirements, a functional smartphone with Android and 3–4 GB of RAM is required, in addition to a camera with 12 megapixels or more; And lastly, for the phone stand, users must utilize a designed tool to access unique capabilities, including the ability to get live feedback from the form monitor.

Test plan constraints:

While the vision of VirtuFit is something Team Blu heavily works upon to bring into existence, there are some constraints and challenges that must be addressed.

1. Form monitoring:

Using technologies appropriate for the React Native environment to recognize and analyze human movements is a challenge. It is essential that the app can correctly identify and score the numerous postures and actions that are used in a variety of workouts. To train the algorithm to handle various body types, levels of fitness, and training regimens, a wide range of data is exhaustive. Therefore, the first release of the application is constrained to specific workouts only.

2. Response consistency:

while deploying the chatbot trained upon the OpenAI model, a constraint faced was consistency of conversational structure. It is not possible to present every user with the same set of questions per conversation, as the model has its own train of thought.

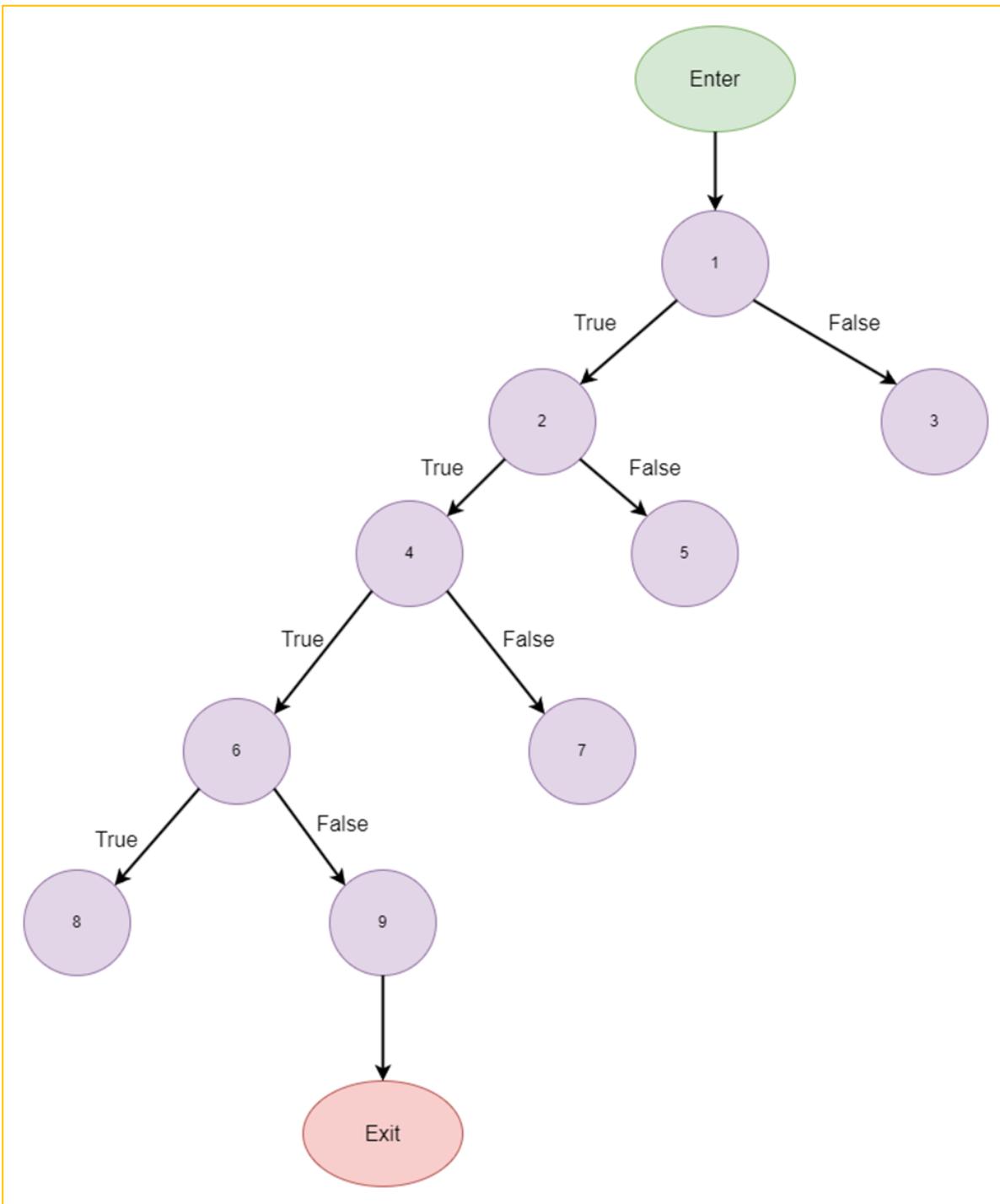
3. Device Compatibility:

Making sure the user's device complies with the application's requirements is a vital restriction.

Unit Testing:

Login Module:

Control Flow Graph (CFG) for Register:



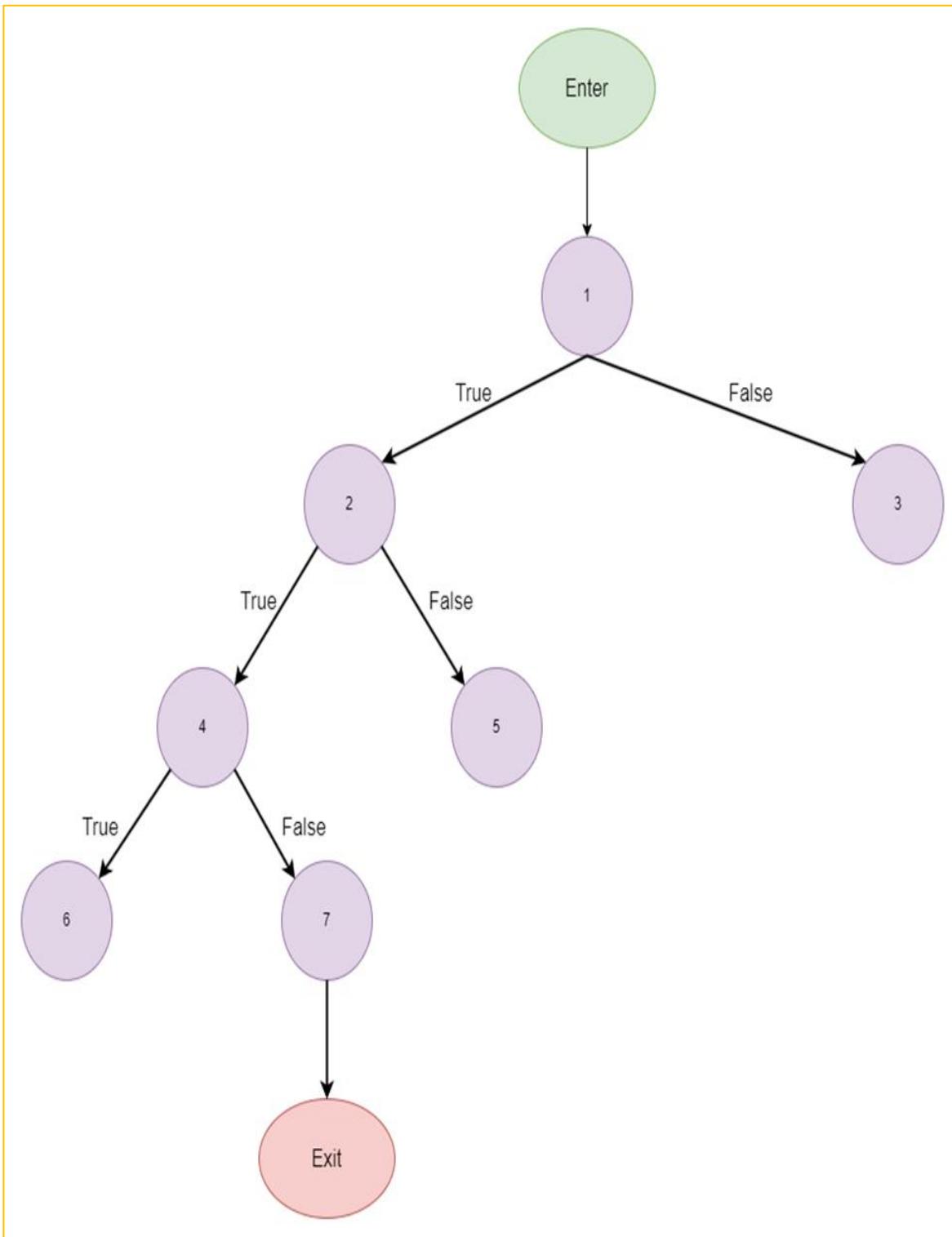
Def-Use Analysis:

Variable	Def/Use analysis	Define Node	Use Node
setData	D U	1	1
username	D UU	1	1,4
email	D UU	1	1,4
password	D UU	1	1,4
db	D U	1	4
tx	D U	4	4
results	D U	4	6
navigation	D U	1	6
error	DU DU	4,5	5,7

```
const db = SQLite.openDatabase({ name: 'VirtuFitMain.db', location: 'default' });

const setData = async () => {
  if (username.length === 0 || email.length === 0 || password.length === 0) {
    console.log("Data not complete");
  } else {
    try {
      await db.transaction(async (tx) => {
        tx.executeSql(
          "INSERT INTO User (UserName, UserEmail, UserPassword) VALUES (?, ?, ?)",
          [username, email, password],
          (tx, results) => {
            if (results.rowsAffected > 0) {
              console.log("Account created successfully");
              navigation.navigate('Home');
            } else {
              console.log("Data insertion failed");
            }
          },
          (tx, error) => {
            console.log("Error inserting data:", error);
          }
        );
      });
    } catch (error) {
      console.log(error);
    }
  }
};
```

Control Flow Graph (CFG) for Login:



Def-Use Analysis:

Variable	Def/Use analysis	Define Node	Use Node
userLogin	D U	1	1
username	D U	1	1
password	D U	1	4
db	D U	1	1
tx	D U	1	1
results	D UUU	1	1,2,4
accountExists	D U	2	2
userPasswordDB	D U	4	4
navigation	D U	1	6
error	D UU	1	1,3

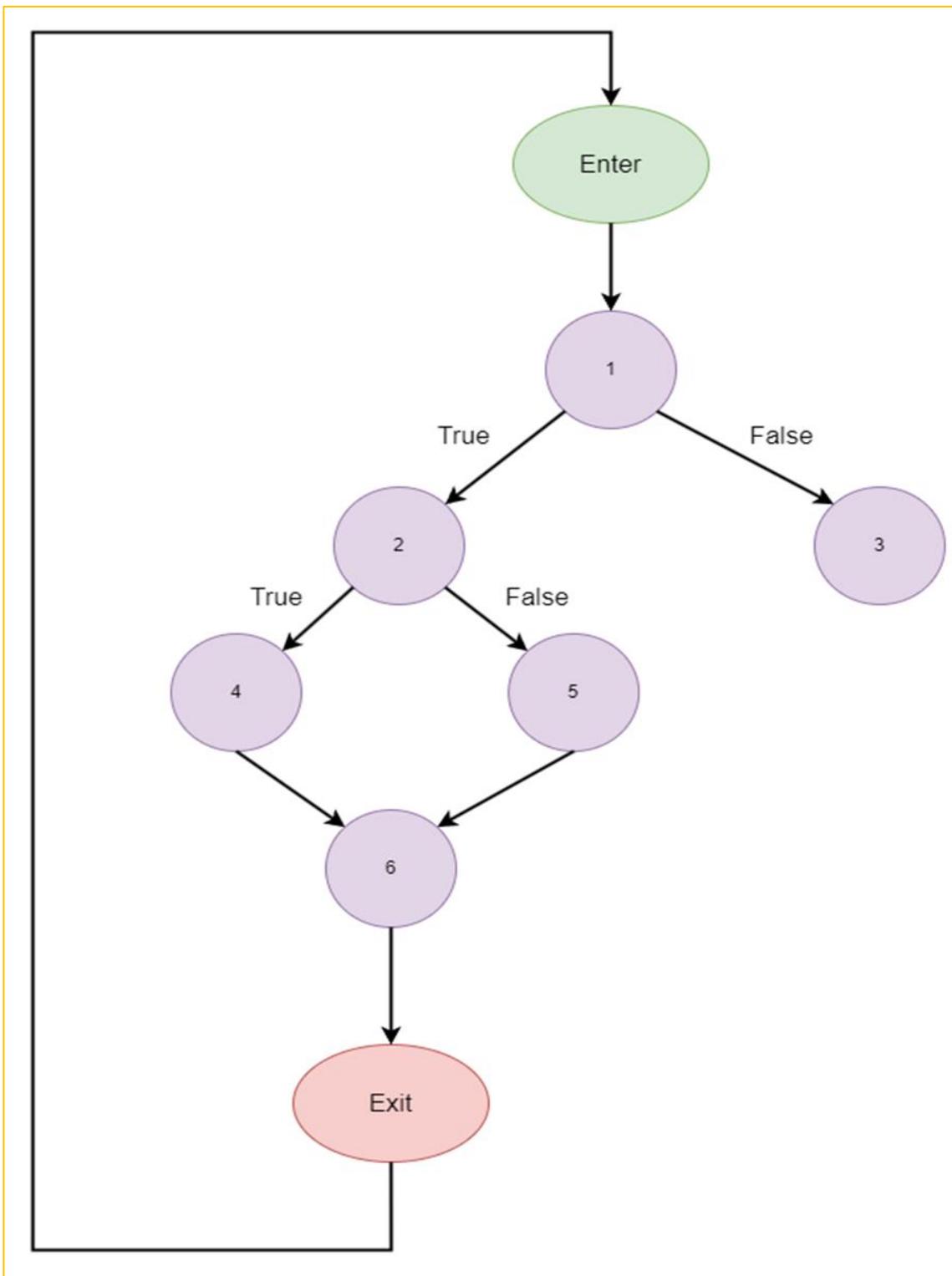
```

const db = SQLite.openDatabase({ name: 'VirtuFitMain.db', location: 'default' });

const userLogin =(username, password) =>{
  db.transaction((tx) =>{
    tx.executeSql('select UserPassword from User where UserName=(?)', [username],
    (tx, results)=>{
      const accountExists = results.rows.length > 0;
      if (accountExists) {
        const userPasswordDB = results.rows.item(0).UserPassword;
        if (userPasswordDB === password) {
          console.log('Account exists and password matched');
          navigation.navigate('Home');
        } else {
          console.log('Account exists but password does not match');
        }
      } else {
        console.log('Account does not exist');
      }
    },
    (tx, error)=>{
      console.log('error logging in: ',error)
    }
  );
});
}
  
```



Chat bot Module: Control Flow Graph (CFG):



Def-Use Analysis:

Variable	Def/Use analysis	Define Node	Use Node
prompt_list	D U	1	1
user_input	D U	1	1
get_bot_response	D U	1	1
prompt	D UU	1	1,2
bot_response	D UU	1	4,5
get_api_response	D U	1	1
choices	D U	2	2
text	D UU	1	1,2
exception e	D UU	1	1,3
create_prompt	D U	1	1
msg_history	D UU	1	1,4
pos	D U	4	4
response	DU DU	1,2	2,6

```

def get_api_response(prompt: str) -> str | None:
    text: str | None = None

    try:
        #request which creates response
        response: dict = openai.Completion.create(
            model='text-davinci-003', #type of chatbot
            prompt=prompt,
            temperature=0.9, #how random the bot is
            max_tokens=100, # response lenght
            top_p=1, #default
            frequency_penalty= 0, #reduces repetitiveness
            presence_penalty=0.9, #how often AI talks about same topic, if low chatbot changes topic more
            stop=[' Human:', ' AI:']
        )
        choices: dict=response.get('choices')[0]
        text = choices.get('text')
        #to get only the reply not whole json response

    except Exception as e:
        print('Error', e)
    return text

```



```
#message history
def msg_history(message: str, pl: list[str]):
    pl.append(message)

def create_prompt(message:str, pl: list[str]) -> str:
    p_message: str = f'\nHuman:{message}'
    msg_history(p_message, pl)
    prompt: str = ''.join(pl)
    return prompt

#only getting the message after position 5, or after the AI:
def get_bot_response(message: str, pl:list[str])-> str:
    prompt: str= create_prompt(message,pl)
    bot_response: str = get_api_response(prompt)

    if bot_response:
        msg_history(bot_response,pl)
        pos: int= bot_response.find('\nVIRTUAL TRAINER: ')
        bot_response = bot_response[pos + 5:]
    else:
        bot_response = 'I am not sure I understood. Please try again'

    return bot_response
```

```
#main function with training data
def main():#ou are a virtual fitness trainer who provides information to users about fitness related questions. begin with asking at
    prompt_list: list[str] = ['you are a virtual fitness trainer for an application called VIRTUFIT. converse with users in a natural
                                language manner']

    while True:
        user_input: str = input('You: ')
        response: str = get_bot_response(user_input, prompt_list)
        print(f'VIRTUAL TRAINER:{response}')

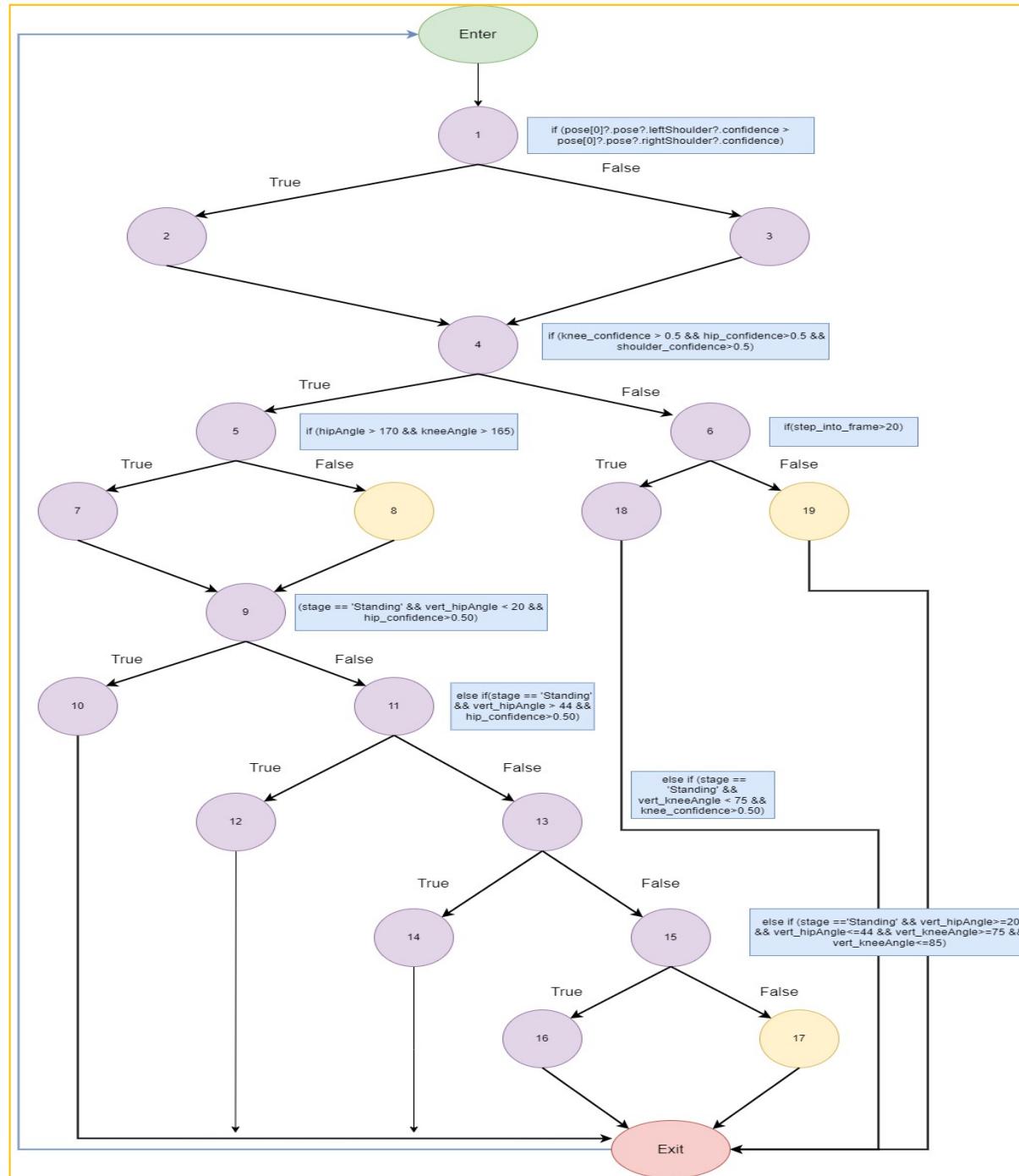
if __name__ == '__main__':
    main()
```

Form Monitoring Module:

Squats:

Control Flow Graph (CFG):

Note: Yellow Nodes are Basically just continue if it doesn't enter the condition.



Def-Use Analysis:

Variable	Def/Use analysis	Define Node	Use Node
Counters Variables for Feedback:			
lean_forward	D U,D,U D D D	1, 10, 11, 14, 18,	10, 10
lean_backward	D D U,D,U D D	1, 10, 12, 14, 18	12, 12,
lower_hip	D D D U, D, U D	1, 10, 12, 14, 18	14, 14,
step_into_frame	D U, D, U	1, 6	6, 19
const [noOfReps, setNoOfReps]	D U	1	16
Confidence Variables for checking user visibility			
const [knee_confidence, setKneeCon]	D D D U U	1, 2, 3	5, 13
const [hip_confidence, setHipCon]	D D D U U U	1, 2, 3	5, 9, 11
const [shoulder_confidence, setShoulderCon]	D D D U	1, 2, 3	4
Stage & Feedback variables			
const [feedback, setFeedback]	D D D D D	1, 10, 12, 14, 6	
const [stage, setStage]	D D U U U U D	1, 7, 16	9, 11, 13, 15
Joints Position Variables			
const [shoulder, setShoulder]	D D D UU	1,2,3	5, 5
const [hip, setHip]	D D D UUUUU	1,2,3	5,5,5,5,5
const [knee, setKnee]	D D D UUUU	1,2,3	5, 5, 5, 5
const [ankle, setAnkle]	D D D U	1,2,3	5
Vertical Position Variables for setting Thresholds			
const [vertKnee, setVertKnee]	D D U	1, 5	5
const [vertHip, setVertHip]	D D U	1, 5	5
Angle Variables			
const [hipAngle, setHipAngle]	D D U	1, 5	5
const [kneeAngle, setKneeAngle]	D D U	1, 5	5
const [vert_kneeAngle, setVertKneeAngle]	D D U } U	1, 5	13, 15
const [vert_hipAngle, setVertHipAngle]	D D U U U	1, 5	9, 11, 15

Test Cases:

Note:

Blue Cell: Represents cells that affect the output.

Yellow cells: Have value but won't affect the main output.

All Nodes:

Test Case	lean_forward	lean_backward	lower_hip	step_into_frame	const [noOfReps, setNoOfReps]	const [knee_confidence, setKneeCon]	const [hip_confidence, setHipCon]	const [shoulder_confidence, setShoulderConf]	const [feedback, setFeedback]	const [stage, setStage]	const [hipAngle, setHipAngle]	const [kneeAngle, setKneeAngle]	const [vert_kneeAngle, setVertKneeAngle]	const [vert_hipAngle, setVertHipAngle]	Path	EO	OO	Verdict
T1	0	0	0	0	0	0.7	0	0	'Empty'	'Empty'	0	0	0	0	EN, 1, (2 or 3), 4, 6, 19, EX	Step_into frame counter increases.	Step_into frame counter increases.	Pass
T2	0	0	0	0	0	0	0.7	0	'Empty'	'Empty'	0	0	0	0	EN, 1, (2 or 3), 4, 6, 19, EX	Step_into frame counter increases.	Step_into frame counter increases.	Pass
T4	0	0	0	0	0	0	0	0.7	'Empty'	'Empty'	0	0	0	0	EN, 1, (2 or 3), 4, 6, 19, EX	Step_into frame counter increases.	Step_into frame counter increases.	Pass
T4	0	0	0	0	0	0.7	0.68	0.55	'Empty'	'Empty'	0	0	0	0	EN, 1, (2 or 3), 4, 5, 8, 9, 11, 13, 15, 17, EX	Sets Values for (Vert Hip, Vert Knee, Hip Angle, Knee Angle, Vert Hip Angle, Vert Knee angle)	Sets Values for (Vert Hip, Vert Knee, Hip Angle, Knee Angle, Vert Hip Angle, Vert Knee angle)	Pass
T5	0	0	0	21	0	0	0	0	'Empty'	'Empty'	0	0	0	0	EN, 1, (2 or 3), 4, 6, 18, EX	Text to speech is used to say the phrase('Step into Frame')	Text to speech is used to say the phrase('Step into Frame')	Pass
T6	0	0	0	0	0	0.88	0.7	0.78	'Empty'	'Empty'	171	170	0	0	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 13, 15, 17, EX	SetStage to Standing	SetStage to Standing	Pass
T7	21	0	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	17	EN, 1, (2 or 3), 4, 5, 7, 9, 10, EX	Text to speech is used to say the phrase('Lean Forward')	Text to speech is used to say the phrase('Lean Forward')	Pass
T8	0	21	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	47	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 12, EX	Text to speech is used to say the phrase('Lean Backward')	Text to speech is used to say the phrase('Lean Backward')	Pass
T9	0	0	21	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	65	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 13, 14, EX	Text to speech is used to say the phrase('Lower your Hip')	Text to speech is used to say the phrase('Lower your Hip')	Pass
T10	17	0	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	17	EN, 1, (2 or 3), 4, 5, 7, 9, 10, EX	Text to speech is used to say the phrase('Lean Forward')	Text to speech is used to say the phrase('Lean Forward')	Pass
T11	0	17	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	47	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 12, EX	Text to speech is used to say the phrase('Lean Backward')	Text to speech is used to say the phrase('Lean Backward')	Pass
T12	0	0	17	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	65	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 13, 14, EX	Text to speech is used to say the phrase('Lower your Hip')	Text to speech is used to say the phrase('Lower your Hip')	Pass
T13	0	0	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	79	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 13, 15, 16, EX	Increment Squat rep counter and TTS should say 'Correct Squat'	Increment Squat rep counter and TTS should say 'Correct Squat'	Pass

Conditional Nodes:

Test Case	lean_forward	lean_backward	lower_hip	step_into_frame	const [noDFRep_s, setNoDFR setKneeC eps]	const [knee_cnfidence, setKneeC on]	const [hip_confidence, setHipConf]	const [shoulder_confidence, setShoulderConf]	const [feedback, selfFeedback]	const [stage, setStage]	const [hipAngle, setHipAngle]	const [kneeAngle, setKneeAngle]	const [vert_kneeAngle, setVertKneeAngle]	const [vert_hipAngle, setVertHipAngle]	Conditional				Path	EO	OO	Verdict
T1	0	0	0	0	0	0	0	0	'Empty'	'Empty'	0	0	0	0	if [pose] > 0? pose ? leftShoulder? confidence > pose[0]? pose ? rightShoulder? confidence = True	EN, 1, 2, 4, 6, 13, EX	The values and confidence of the left joints of the user are stored		Pass			
T2	0	0	0	0	0	0	0	0	'Empty'	'Empty'	0	0	0	0	if [pose] > 0? pose ? leftShoulder? confidence > pose[0]? pose ? rightShoulder? confidence = False Else = True	EN, 1, 3, 4, 6, 13, EX	The values and confidence of the right joints of the user are stored		Pass			
T4	0	0	0	0	0.68	0.85	0.79	'Empty'	'Empty'	0	0	0	0	if [linee_confidence] > 0.5 & [hp_confidence] > 0.5 & [shoulder_confidence] / 0.5 = True	EN, 1, 2 (or 3), 4, 5, 6, 9, 11, 13, 15, 17, EX	will set values of angles between joints and thresholds		Pass				
T4	0	0	0	0	0	0	0.3	0	'Empty'	'Empty'	0	0	0	0	if [linee_confidence] > 0.5 & [hp_confidence] > 0.5 & [shoulder_confidence] / 0.5 = False Else = True	EN, 1, 2 (or 3), 4, 6, 13, EX	Step into frame counter increases. Once it reaches above 20 it will activate a text to speech to warn user		Pass			
T5	0	0	0	0	0	0.7	0.6	0.9	'Empty'	'Empty'	173	169	0	0	if [hipAngle] > 170 & [kneeAngle] > 165 = True	EN, 1, 2 (or 3), 4, 5, 6, 9, 11, 13, 15, 17, EX	SetStage to standing so we know the user currently is standing		Pass			
T6	0	0	0	0	0	0.7	0.6	0.9	'Empty'	'Empty'	165	169	0	0	if [hipAngle] > 170 & [kneeAngle] > 165 = False	EN, 1, 2 (or 3), 4, 5, 6, 9, 11, 13, 15, 17, EX	Stage is not set and no other condition is run		Pass			
T7	0	0	0	0	0	0.7	0.6	0.9	'Empty'	'Standing'	0	0	0	17	if [stage] == Standing & [vert_hipAngle] < 20 & [hp_confidence] > 0.5 = True	EN, 1, 2 (or 3), 4, 5, 6, 9, 10, EX	Lean forward counter increases. Once it goes above 20 it will activate text to speech warning		Pass			
T8	0	0	0	0	0	0.7	0.6	0.9	'Empty'	'Standing'	0	0	0	47	if [stage] == Standing & [vert_hipAngle] < 20 & [hp_confidence] > 0.5 = False Else = EN, 1, 2 (or 3), 4, 5, 6, 9, 11, 12, EX	Lean back counter increases. Once it goes above 20 it will activate text to speech warning		Pass				
T9	0	0	0	0	0	0.7	0.6	0.9	'Empty'	'Standing'	0	0	70	17	if [stage] == Standing & [vert_hipAngle] < 20 & [hp_confidence] > 0.5 = False Else = EN, 1, 2 (or 3), 4, 5, 6, 9, 11, 13, 14, EX	Lower your hip counter increases. Once it goes above 20 it will activate text to speech warning		Pass				
T10	0	0	0	0	0	0.7	0.6	0.9	'Empty'	'Standing'	173	170	77	17	if [stage] == Standing & [vert_hipAngle] < 20 & [hp_confidence] > 0.5 = False Else = EN, 1, 2 (or 3), 4, 5, 6, 9, 11, 13, 15, 16, EX	Lean back counter increases. Once it goes above 20 it will activate text to speech warning		Pass				

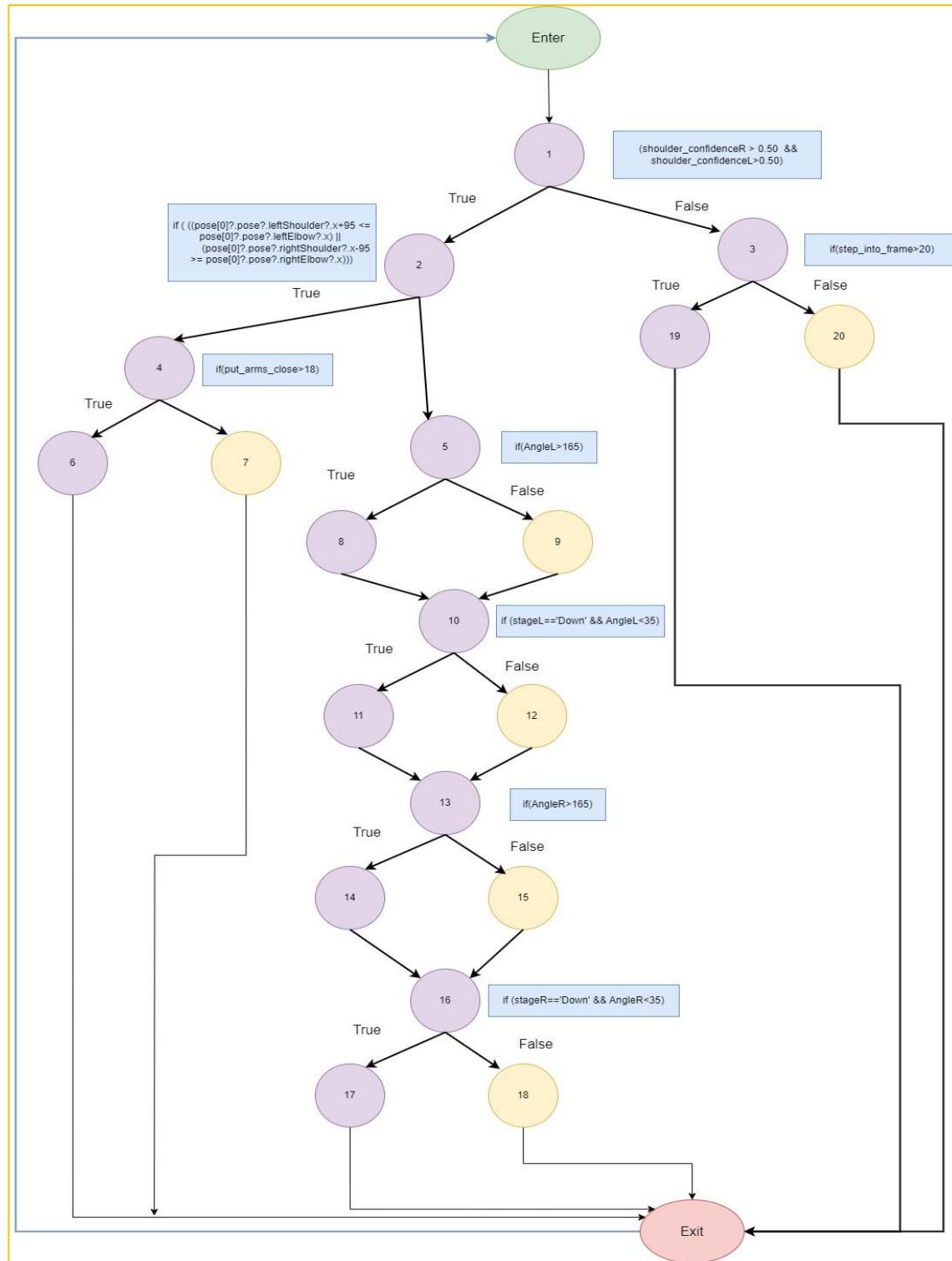
All Independent paths:

Test Case	lean_forward	lean_backward	lower_hip	step_into_frame	const [noDFRep_s, setNoDFR setKneeC eps]	const [knee_cnfidence, setKneeC on]	const [hip_confidence, setHipConf]	const [shoulder_confidence, setShoulderConf]	const [feedback, selfFeedback]	const [stage, setStage]	const [hipAngle, setHipAngle]	const [kneeAngle, setKneeAngle]	const [vert_kneeAngle, setVertKneeAngle]	const [vert_hipAngle, setVertHipAngle]	Path				Path	EO	OO	Verdict
T1	0	0	0	0	0	0.7	0	0	'Empty'	'Empty'	0	0	0	0	EN, 1, 2 (or 3), 4, 6, 13, EX	Step_into frame counter increases.	Step_into frame counter increases.		Pass			
T2	0	0	0	0	0	0.7	0.68	0.55	'Empty'	'Empty'	0	0	0	0	EN, 1, 2 (or 3), 4, 6, 13, EX	Sets Values for [Vert Hip, Vert Knee, Hip Angle, Knee Angle, Vert Hip Angle, Vert Knee angle]	Sets Values for [Vert Hip, Vert Knee, Hip Angle, Knee Angle, Vert Hip Angle, Vert Knee angle]		Pass			
T3	0	0	0	21	0	0	0	0	'Empty'	'Empty'	0	0	0	0	EN, 1, 2 (or 3), 4, 6, 13, EX	Text to speech is used to say the phrase['Step into Frame']	Text to speech is used to say the phrase['Step into Frame']		Pass			
T4	0	0	0	0	0	0.88	0.7	0.78	'Empty'	'Empty'	171	170	0	0	EN, 1, 2 (or 3), 4, 5, 6, 9, 11, 13, 15, 17, EX	SetStage to Standing	SetStage to Standing		Pass			
T5	21	0	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	17	EN, 1, 2 (or 3), 4, 6, 13, EX	Text to speech is used to say the phrase['Lean Forward']	Text to speech is used to say the phrase['Lean Forward']		Pass			
T6	0	21	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	47	EN, 1, 2 (or 3), 4, 5, 7, 9, 11, 13, 15, 17, EX	Text to speech is used to say the phrase['Lean Backward']	Text to speech is used to say the phrase['Lean Backward']		Pass			
T7	0	0	21	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	65	34	EN, 1, 2 (or 3), 4, 5, 7, 9, 10, EX	Text to speech is used to say the phrase['Lower your Hip']	Text to speech is used to say the phrase['Lower your Hip']		Pass			

T8	17	0	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	17	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 12, EX	increment lean forward	Text to speech is used to say the phrase("Lean Forward")	Pass
T9	0	17	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	0	47	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 13, 14, EX	increment lean backward	Text to speech is used to say the phrase("Lean Backward")	Pass
T10	0	0	17	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	65	34	EN, 1, (2 or 3), 4, 5, 7, 9, 10, EX	increment lower your hips	Text to speech is used to say the phrase("Lower your Hip")	Pass
T11	0	0	0	0	0	0.88	0.7	0.78	'Empty'	'Standing'	0	0	38	79	EN, 1, (2 or 3), 4, 5, 7, 9, 11, 12, EX	Increment Squat rep counter and TTS should say 'Correct Squat'	Increment Squat rep counter and TTS should say 'Correct Squat'	Pass

Bicep Curl:

Control Flow Graph (CFG):



Def-Use Analysis:

Variable	Def/Use analysis	Define Node	Use Node
Counters Variables for Feedback:			
put_arms_close	D U,D,U	1, 6	4, 7
step_into_frame	D U,D,U	1, 19	3, 20
const [noOfRepsL, setNoOfRepsL]	D U	1	11
const [noOfRepsR, setNoOfRepsR]	D U	1	17
Confidence Variables for checking user visibility			
const [shoulder_confidenceR, setShoulderConR]	D U	1	1
const [shoulder_confidenceL, setShoulderConL]	D U	1	1
Stage & Feedback variables			
const [feedback, setFeedback]	D D D	1, 6, 19	
const [stageR, setStageR]	D D U D	1, 14, 17	16
const [stageL, setStageL]	D D U D	1, 8, 11	10
Joints Position Variables			
const [shoulderR, setShoulderR]	D D U	1,5	5
const [elbowR, setElbowR]	D D U	1,5	5
const [wristR, setWristR]	D D U	1,5	5
const [shoulderL, setShoulderL]	D D U	1,5	5
const [elbowL, setElbowL]	D D U	1,5	5
const [wristL, setWristL]	D D U	1,5	5
Angle Variables			
const [AngleR, setAngleR]	D D U U	1, 5	5,10
const [AngleL, setAngleL]	D D U U	1, 5	13, 16

Test Cases:

All Nodes:

Test Case	put_arms_close	step_into_frame	const [noOfRepsL, setNoOfRepsL]	const [noOfRepsR, setNoOfRepsR]	const [shoulder_confidenceL, setShoulderConfidenceL]	const [shoulder_confidenceR, setShoulderConfidenceR]	const [feedback, setFeedback]	const [stageR, setStageR]	const [stageL, setStageL]	const [AngleR, setAngleR]	const [AngleL, setAngleL]	Path	EO	OO	Verdict	
T1	0	0	0	0	0.6	0.4	Empty'	Empty'	Empty'	0	0	EN, 1, 3, 20, EX	Increments step into frame by 1	Increments step into frame by 1	Pass	
T2	0	21	0	0	0.6	0.4	Empty'	Empty'	Empty'	0	0	EN, 1, 3, 19, EX	Text to speech is used to warn user to step into frame	Text to speech is used to warn user to step into frame	Pass	
T3	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	0	0	EN, 1, 2, 5, 9, 10, 12, 13, 15, 16, 18, EX	Set Values to Left and Right (Shoulder, Elbow/Wrist)	Set Values to Left and Right (Shoulder, Elbow/Wrist)	Pass	
T4	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	175	160	EN, 1, 2, 5, 8, 10, 12, 13, 15, 16, 18, EX	Only stage of Left is set to down	Only stage of Left is set to down	Pass	
T5	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	160	172	EN, 1, 2, 5, 9, 10, 12, 13, 14, 16, 18, EX	Only stage of Right is set to down	Only stage of Right is set to down	Pass	
T6	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	170	172	EN, 1, 2, 5, 8, 10, 12, 13, 15, 16, 18, EX	Both Stages left and Right set to Down	Both Stages left and Right set to Down	Pass	
T7	0	0	0	0	0.7	0.78	Empty'	Down'	Empty'	34	172	EN, 1, 2, 5, 8, 10, 12, 13, 15, 16, 17, EX	Increments Rep counter of Right arm and sets stage to UP	Increments Rep counter and sets stage to UP	Pass	
T8	0	0	0	0	0.7	0.78	Empty'	Empty'	Down'	170	30	EN, 1, 2, 5, 9, 10, 11, 13, 14, 16, 18, EX	Increments Rep counter of Left arm and sets stage to UP	Increments Rep counter of Left arm and sets stage to UP	Pass	
T9	0	0	0	0	0.7	0.78	Empty'	Down'	Down'	30	30	EN, 1, 2, 5, 8, 10, 11, 13, 14, 16, 17, EX	Both left and right rep counter will increment	Both left and right rep counter will increment	Pass	
T10	0	0	0	0	0	0.7	0.86	Empty'	Empty'	Empty'	0	0	EN, 1, 2, 4, 8, EX	Put arms close variable incremented if reaches above 20 it will activate Text to speech warning	Put arms close variable incremented if reaches above 20 it will activate Text to speech warning	Pass

Conditional Nodes:

Test Case	put_arms_close	step_into_frame	const [noOfRepsL, setNoOfRepsL]	const [noOfRepsR, setNoOfRepsR]	const [shoulder_confidenceL, setShoulderConfidenceL]	const [shoulder_confidenceR, setShoulderConfidenceR]	const [feedback, setFeedback]	const [stageR, setStageR]	const [stageL, setStageL]	const [AngleR, setAngleR]	const [AngleL, setAngleL]	Conditional	Path	EO	OO	Verdict
T1	0	0	0	0	0.7	0.86	Empty'	Empty'	Empty'	0	0	(shoulder_confidenceL > 50 & shoulder_confidenceR > 50) = True	EN, 1, 2, 5, 9, 10, 12, 13, 15, 16, 18, EX	goes to next if condition in node 2	goes to next if condition in node 2	Pass
T2	0	0	0	0	0.6	0.4	Empty'	Empty'	Empty'	0	0	(shoulder_confidenceL > 50 & shoulder_confidenceR > 50) = False Else = True	EN, 1, 3, 20, EX	Step into Frame counter increases	Step into Frame counter increases	Pass
T3	0	0	0	0	0.7	0.86	Empty'	Empty'	Empty'	0	0	((pose[0]? pose.leftShoulder.x <= pose[0]? pose.leftElbow.x) (pose[0]? pose.rightShoulder.x <= pose[0]? pose.rightElbow.x)) = True	EN, 1, 2, 4, 8, EX	Put arms close variable incremented if reaches above 20 it will activate Text to speech warning	Put arms close variable incremented if reaches above 20 it will activate Text to speech warning	Pass
T4	0	0	0	0	0.7	0.86	Empty'	Empty'	Empty'	0	0	((pose[0]? pose.leftShoulder.x <= pose[0]? pose.leftElbow.x) (pose[0]? pose.rightShoulder.x <= pose[0]? pose.rightElbow.x)) = True Else = true	EN, 1, 2, 5, 9, 10, 12, 13, 15, 16, 18, EX	Will set values for joint variables Left and Right (Elbow, shoulder, wrist)	Will set values for joint variables Left and Right (Elbow, shoulder, wrist)	Pass
T5	0	0	0	0	0.7	0.86	Empty'	Empty'	Empty'	175	174	If(AngleL>165) = True If(stageL==Down & AngleL<CS5) = False If(AngleR>165) = True If(stageR==Down & AngleR<CS5) = False	EN, 1, 2, 5, 8, 10, 12, 13, 14, 16, 18, EX	Both Stages left and Right set to Down	Both Stages left and Right set to Down	Pass
T6	0	0	0	0	0.7	0.86	Empty'	Empty'	Empty'	33	27	If(AngleL>165) = False If(stageL==Down & AngleL<CS5) = True If(AngleR>165) = False If(stageR==Down & AngleR<CS5) = True	EN, 1, 2, 5, 8, 10, 11, 13, 14, 16, 17, EX	Both left and right rep counter will increment	Both left and right rep counter will increment	Pass

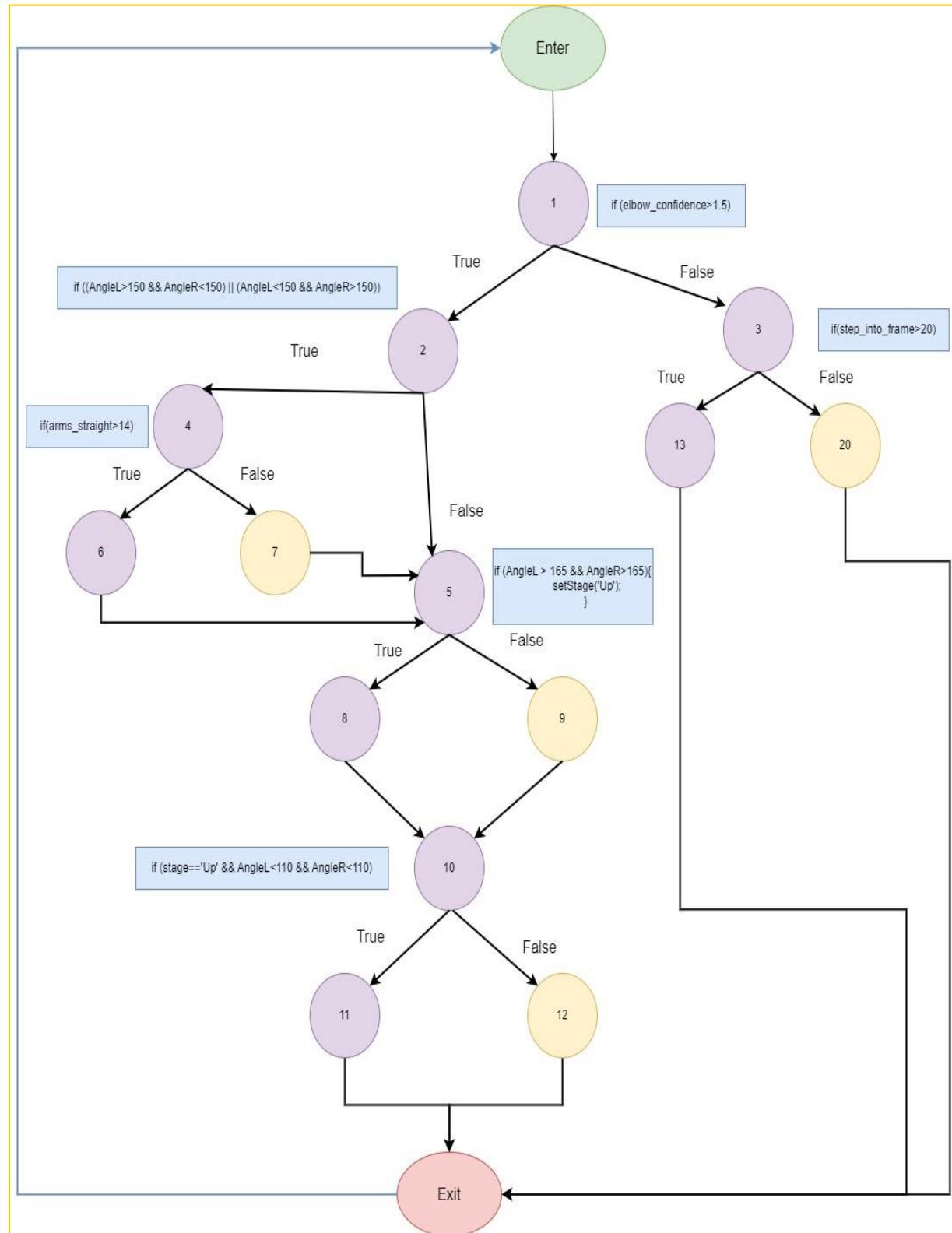
All Independent paths:

Test Case	put_arms_cloose	step_into_frame	const [noOfRepsL]	const [noOfRepsR]	const [shoulder_cconfidenceR]	const [shoulder_cconfidenceL]	const [feedbackL, selfFeedbackR]	const [stageR, setStageR]	const [stageL, setStageL]	const [angleR, setAngleR]	const [angleL, setAngleL]	Path	EO	OO	Verdict
T1	0	0	0	0	0.6	0.4	Empty'	Empty'	Empty'	0	0	EN, 1, 3, 20, EX	Increments step into frame by 1	Increments step into frame by 1	Pass
T2	0	21	0	0	0.6	0.4	Empty'	Empty'	Empty'	0	0	EN, 1, 3, 19, EX	Text to speech is used to want user to step into frame	Text to speech is used to want user to step into frame	Pass
T3	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	0	0	EN, 1, 2, 5, 3, 10, 12, 13, 15, 16, 18, EX	Set Values to Left and Right (Shoulder, Elbow/Wrist)	Set Values to Left and Right (Shoulder, Elbow/Wrist)	Pass
T4	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	175	160	EN, 1, 2, 5, 8, 10, 12, 13, 15, 16, 18, EX	Only stage of Left is set to down	Only stage of Left is set to down	Pass
T5	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	160	172	EN, 1, 2, 5, 8, 10, 12, 13, 14, 16, 18, EX	Only stage of Right is set to down	Only stage of Right is set to down	Pass
T6	0	0	0	0	0.7	0.78	Empty'	Empty'	Empty'	170	172	EN, 1, 2, 5, 8, 10, 12, 13, 15, 16, 18, EX	Both Stages left and Right set to Down	Both Stages left and Right set to Down	Pass
T7	0	0	0	0	0.7	0.78	Empty'	Down'	Empty'	34	172	EN, 1, 2, 5, 8, 10, 12, 13, 15, 16, 17, EX	Increments Rep counter of Right arm and sets stage to UP	Increments Rep counter of Right arm and sets stage to UP	Pass
T8	0	0	0	0	0.7	0.78	Empty'	Empty'	Down'	170	30	EN, 1, 2, 5, 8, 10, 11, 13, 14, 16, 18, EX	Increments Rep counter of Left arm and sets stage to UP	Increments Rep counter of Left arm and sets stage to UP	Pass
T9	0	0	0	0	0.7	0.78	Empty'	Down'	Down'	30	30	EN, 1, 2, 5, 8, 10, 11, 13, 14, 16, 17, EX	Both left and right rep counter will increment	Both left and right rep counter will increment	Pass



Push Ups:

Control Flow Graph (CFG):



Def-Use Analysis:

Variable	Def/Use analysis	Define Node	Use Node
Counters Variables for Feedback:			
arms_straight	D U, D, U	1,6	4, 7
step_into_frame	D U, D, U	1, 13	3,14
const [noOfReps, setNoOfReps]	D U	1	11
Confidence Variables for checking user visibility			
const [elbow_confidence, setElbowCon]	D, U	1	1
Stage & Feedback variables			
const [feedback, setFeedback]	D D	6, 13	
const [stage, setStage]	D D U, D	1, 8,11	10
Joints Position Variables			
const [shoulderR, setShoulderR]	D D, U	1, 2,	2
const [elbowR, setElbowR]	D D, U	1, 2,	2
const [wristR, setWristR]	D D, U	1, 2,	2
const [shoulderL, setShoulderL]	D D, U	1, 2,	2
const [elbowL, setElbowL]	D D, U	1, 2,	2
const [wristL, setWristL]	D D, U	1, 2,	2
Angle Variables			
const [AngleL, setAngleL]	D D, U U U	1, 2	2, 5, 10
const [AngleR, setAngleR]	D D, U U U	1, 2	2, 5, 10

Test Cases:

All Nodes:

Test Case	arms_straig ht	step_into_f rame	const [noOfReps, setNoOfRe ps]	const [elbow_c onfide nce, setElbow Con]	const [feedback, setFeedbac k]	const [stage, setStage]	const [AngleL, setAngle L]	const [AngleR, setAngle R]	Path	EO	OO	Verdict
T1	0	0	0	0.8	'Empty'	'Empty'	0	0	EN, 1,3,20,EX	Step into Frame counter is incremented	Step into Frame counter is incremented	Pass
T2	0	0	0	1.75	'Empty'	'Empty'	0	0	EN, 1,2,5,9,10,12,EX	Set Values to Left and Right (Shoulder, Elbow, Wrist)	Set Values to Left and Right (Shoulder, Elbow, Wrist)	Pass
T3	0	0	0	1.79	'Empty'	'Empty'	165	137	EN, 1,2,4,7,5,9,10,12,EX	Arms straight counter is incremented	Arms straight counter is incremented	Pass
T4	0	21	0	0.8	'Empty'	'Empty'	0	0	EN, 1,3,13,EX	Text to speech is activated to warn user step into frame	Text to speech is activated to warn user step into frame	Pass
T5	21	0	0	1.79	'Empty'	'Empty'	165	137	EN, 1,2,4,6,5,9,10,12,EX	Text to speech is activated to warn users arms straight	Text to speech is activated to warn users arms straight	Pass
T6	0	0	0	1.88	'Empty'	'Empty'	170	169	EN, 1,2,5,9,10,12,EX	Set Stage to UP	Set Stage to UP	Pass
T7	0	0	0	1.88	'Empty'	'Up'	105	90	EN, 1,2,5,,9,10,11,EX	Correct push increments rep counter by 1	Correct push increments rep counter by 1	Pass

Conditional Nodes:

Test Case	arms_straight	step_into_frame	const [noOfReps, setNoOfReps]	const [elbow_confidence, setElbowConf]	const [feedback, setFeedback]	const [stage, setStage]	const [AngleL, setAngleL]	const [AngleR, setAngleR]	Conditional	Path	EO	OO	Verdict
T1	0	0	0	0.8	'Empty'	'Empty'	0	0	if (elbow_confidence>1.5) = False Else = True	EN, 1, 3,20, EX	Step into Frame counter is incremented	Step into Frame counter is incremented	Pass
T2	0	0	0	1.75	'Empty'	'Empty'	0	0	if (elbow_confidence>1.5) = True	EN, 1, 2,5, 9, 10, 12, EX	Set Values to Left and Right (Shoulder, Elbow, Wrist)	Set Values to Left and Right (Shoulder, Elbow, Wrist)	Pass
T3	0	0	0	1.79	'Empty'	'Empty'	165	137	if ((AngleL>150 & AngleR<150) (AngleL<150 & AngleR>150)) = True	EN, 1, 2, 4, 7, 5, 9, 10, 12, EX	Arms straight counter is incremented	Arms straight counter is incremented	Pass
T4	0	21	0	0.8	'Empty'	'Empty'	0	0	if (step_into_frame>20) = True	EN, 1, 3,13, EX	Text to speech is activated to warn user step into frame	Text to speech is activated to warn user step into frame	Pass
T5	21	0	0	1.79	'Empty'	'Empty'	165	137	if (arms_straight>14) = True	EN, 1, 2, 4, 6, 5, 9, 10, 12, EX	Text to speech is activated to warn users arms staright	Text to speech is activated to warn users arms staright	Pass
T6	0	0	0	1.88	'Empty'	'Empty'	170	169	if (AngleL > 165 & AngleR > 165) = True	EN, 1, 2, 5, 9, 10, 12, EX	Set Stage to UP	Set Stage to UP	Pass
T7	0	0	0	1.88	'Empty'	'Up'	105	90	if (stage=='Up' && AngleL<110 & AngleR<110) = True	EN, 1, 2, 5,, 9, 10, 11, EX	Correct push increments rep counter by 1	Correct push increments rep counter by 1	Pass

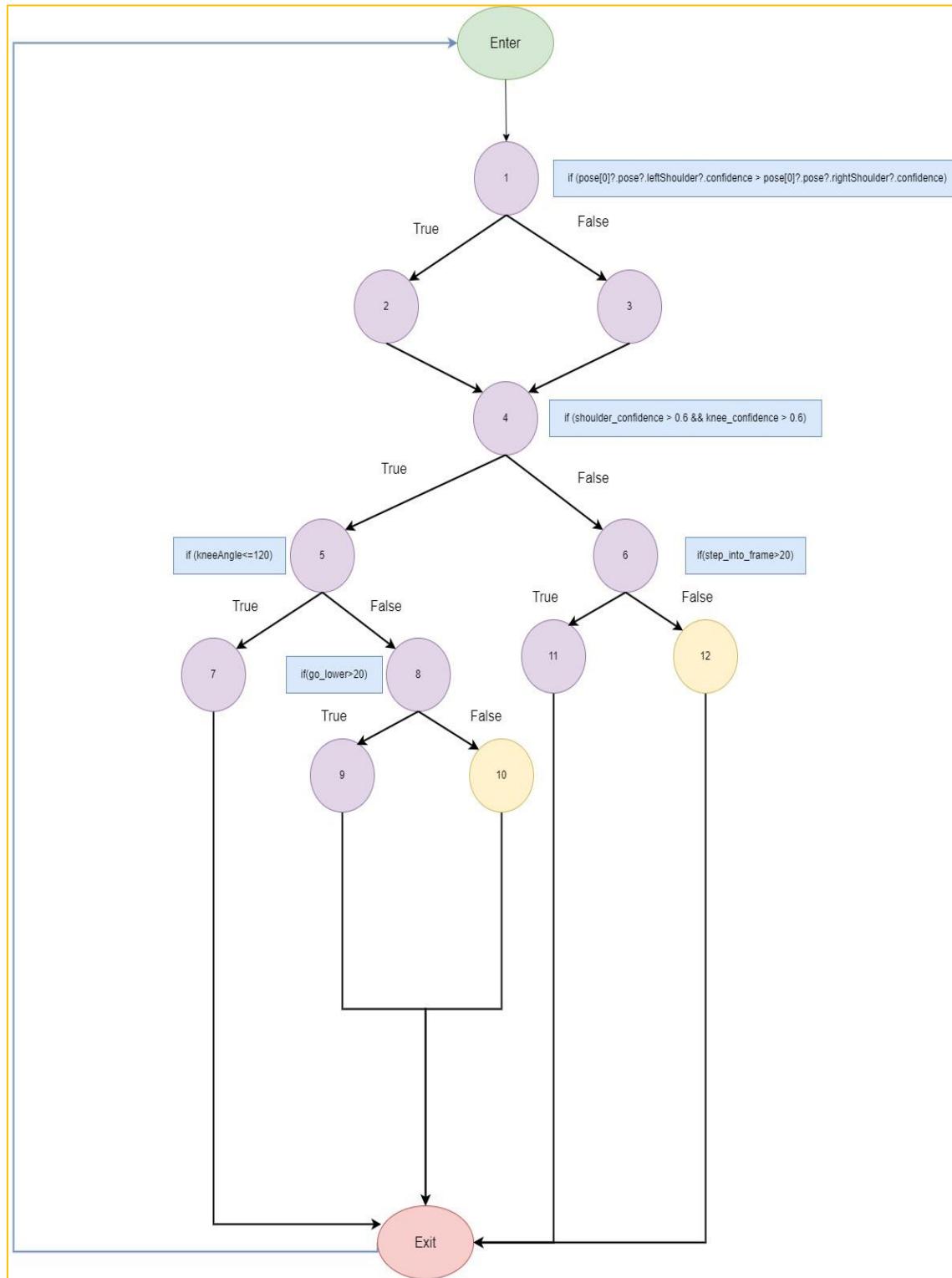
All Independent paths:

Test Case	arms_straight	step_into_frame	const [noOfReps, setNoOfReps]	const [elbow_confidence, setElbowConf]	const [feedback, setFeedback]	const [stage, setStage]	const [AngleL, setAngleL]	const [AngleR, setAngleR]	Path	EO	OO	Verdict
T1	0	0	0	0.8	'Empty'	'Empty'	0	0	EN, 1, 3,20, EX	Step into Frame counter is incremented	Step into Frame counter is incremented	Pass
T2	0	0	0	1.75	'Empty'	'Empty'	0	0	EN, 1, 2,5, 9, 10, 12, EX	Set Values to Left and Right (Shoulder, Elbow, Wrist)	Set Values to Left and Right (Shoulder, Elbow, Wrist)	Pass
T3	0	0	0	1.79	'Empty'	'Empty'	165	137	EN, 1, 2, 4, 7, 5, 9, 10, 12, EX	Arms straight counter is incremented	Arms straight counter is incremented	Pass
T4	0	21	0	0.8	'Empty'	'Empty'	0	0	EN, 1, 3,13, EX	Text to speech is activated to warn user step into frame	Text to speech is activated to warn user step into frame	Pass
T5	21	0	0	1.79	'Empty'	'Empty'	165	137	EN, 1, 2, 4, 6, 5, 9, 10, 12, EX	Text to speech is activated to warn users arms staright	Text to speech is activated to warn users arms staright	Pass
T6	0	0	0	1.88	'Empty'	'Empty'	170	169	EN, 1, 2, 5, 9, 10, 12, EX	Set Stage to UP	Set Stage to UP	Pass
T7	0	0	0	1.88	'Empty'	'Up'	105	90	EN, 1, 2, 5,, 9, 10, 11, EX	Correct push increments rep counter by 1	Correct push increments rep counter by 1	Pass



Wall sit :

Control Flow Graph (CFG):



Def-Use Analysis:

Variable	Def/Use analysis	Define Node	Use Node
Counters Variables for Feedback:			
step_into_frame	D U, D, U	1, 11	6, 12
go_lower	D U, D, U	1, 9	8, 10
const countRef	D , U, U	1	1, 1
const [user_visibility, setUserVisibility]	D U U U U	1	1, 3, 7, 9
Confidence Variables for checking user visibility			
const [shoulder_confidence, setShoulderCon]	D D D U	1, 2, 3	4
const [knee_confidence, setKneeCon]	D D D U	1, 2, 3	4
Joints Position Variables			
const [knee, setKnee]	D D, U	1, 2, 3	4
const [hip, setHip]	D D, U	1,2, 3	4
const [ankle, setAnkle]	D D, U	1,2 ,3	4
Angle Variables			
const [kneeAngle, setKneeAngle]	D D U	1, 4,	7

Test Cases:

All Nodes:

Test Case	step_into_frame	go_lower	const countRef	const [user_visibility, setUserVisibility]	const [shoulder_confidence, setShoulderCon]	const [knee_confidence, setKneeCon]	const [kneeAngle, setKneeAngle]	Path	E0	OO	Verdict
T1	0	0	0	FALSE	0.43	0.57	0	EN, 1, (2 or 3), 4, 6, 12, EX	Increment step into frame counter	Increment step into frame counter	Pass
T2	21	0	0	FALSE		0.57	0	EN, 1, (2 or 3), 4, 6, 11, EX	Activates Text to speech to warn user to step into frame	Activates Text to speech to warn user to step into frame	Pass
T3	0	0	0	FALSE	0.78	0.95	105	EN, 1, (2 or 3), 4, 5, 7, EX	Sets Visibility to True starts timer module	Sets Visibility to True starts timer module	Pass
T4	0	0	0	TRUE	0.78	0.95	105	EN, 1, (2 or 3), 4, 5, 7, EX	Timer module is incremented time is moving	Sets Visibility to True starts timer module	Pass
T5	0	0	0	TRUE	0.78	0.95	127	EN, 1, (2 or 3), 4, 5, 8, 10, EX	User visibility set to false and Incremented go lower counter	User visibility set to false and Incremented go lower counter	Pass
T6	0	21	0	TRUE	0.78	0.95	127	EN, 1, (2 or 3), 4, 5, 8, 9, EX	User visibility set to False and Text to speech is activates saying go lower	User visibility set to False and Text to speech is activates saying go lower	Pass

Conditional Nodes:

Test Case	step_into_frame	go_lower	const countRef	const [user_visibility, setUserVisibility]	const [shoulder_confidence, setShoulderCon]	const [knee_confidence, setKneeConf]	const [kneeAngle, setKneeAngle]	Conditional	Path	EO	OO	Verdict
T1	0	0	0	FALSE	0.6	0.7	0	if (pose[0].pose?.leftShoulder?.confidence > pose[0].pose?.rightShoulder?.confidence) = True	EN, 1, 2, 4, 6, 12, EX	Set Knee, Hip, Ankle, KneeCon, ShoulderCon to Left	Set Knee, Hip, Ankle, KneeCon, ShoulderCon to Left	Pass
T2	0	0	0	FALSE	0.6	0.7	0	if (pose[0].pose?.leftShoulder?.confidence > pose[0].pose?.rightShoulder?.confidence) = False Else = True	EN, 1, 3, 4, 6, 12, EX	Set Knee, Hip, Ankle, KneeCon, ShoulderCon to Right	Set Knee, Hip, Ankle, KneeCon, ShoulderCon to Right	Pass
T3	0	0	0	TRUE	0.78	0.95	127	(shoulder_confidence > 0.6 & knee_confidence > 0.6) = True	EN, 1, (2 or 3), 4, 5, 8, 10, EX	User visibility set to false and Incremented go lower counter	User visibility set to false and Incremented go lower counter	Pass
T4	0	0	0	FALSE	0.78	0.95	105	(shoulder_confidence > 0.6 & knee_confidence > 0.6) = True	EN, 1, (2 or 3), 4, 5, 7, EX	Sets Visibility to True starts timer module	Sets Visibility to True starts timer module	Pass
T5	21	0	0	FALSE	0.43	0.57	0	(shoulder_confidence > 0.6 & knee_confidence > 0.6) = False Else = True	EN, 1, (2 or 3), 4, 6, 11, EX	Activates Text to speech to warn user to step into frame	Activates Text to speech to warn user to step into frame	Pass
T6	0	0	0	TRUE	0.78	0.95	105		EN, 1, (2 or 3), 4, 5, 7, EX	Timer module is incremented time is moving	Timer module is incremented time is moving	Pass
T7	0	21	0	TRUE	0.78	0.95	127	if (kneeAngle <= 120) = False else = True	EN, 1, (2 or 3), 4, 5, 8, 9, EX	User visibility set to False and Text to speech is activates saying go lower	User visibility set to False and Text to speech is activates saying go lower	Pass

All Independent paths:

Test Case	step_into_frame	go_lower	const countRef	const [user_visibility, setUserVisibility]	const [shoulder_confidence, setShoulderCon]	const [knee_confidence, setKneeConf]	const [kneeAngle, setKneeAngle]	Path	EO	OO	Verdict
T1	0	0	0	FALSE	0.43	0.57	0	EN, 1, (2 or 3), 4, 6, 12, EX	Increment step into frame counter	Increment step into frame counter	Pass
T2	21	0	0	FALSE	0.43	0.57	0	EN, 1, (2 or 3), 4, 6, 11, EX	Activates Text to speech to warn user to step into frame	Activates Text to speech to warn user to step into frame	Pass
T3	0	0	0	FALSE	0.78	0.95	105	EN, 1, (2 or 3), 4, 5, 7, EX	Sets Visibility to True starts timer module	Sets Visibility to True starts timer module	Pass
T4	0	0	0	TRUE	0.78	0.95	105	EN, 1, (2 or 3), 4, 5, 7, EX	Timer module is incremented time is moving	Sets Visibility to True starts timer module	Pass
T5	0	0	0	TRUE	0.78	0.95	127	EN, 1, (2 or 3), 4, 5, 8, 10, EX	User visibility set to false and Incremented go lower counter	User visibility set to false and Incremented go lower counter	Pass
T6	0	21	0	TRUE	0.78	0.95	127	EN, 1, (2 or 3), 4, 5, 8, 9, EX	User visibility set to False and Text to speech is activates saying go lower	User visibility set to False and Text to speech is activates saying go lower	Pass

Acceptance Testing:

Sign up for VirtuFit

Tested Requirement	Create a Virtufit account (Positive)
Quick Description	The user is able to successfully create an account on VirtuFit
Prerequisite	Email and username must not be associating with an existing account
Inputs	Full Name, email, password
Expected Outputs	Account is successfully created
Observed Outputs	-
Verdict	-

Tested Requirement	Create a Virtufit account (Negative 1)
Quick Description	The user is unable to create an account on VirtuFit due to missing details
Prerequisite	Email and username must not be associated with existing account
Inputs	Full Name, email, password (One or more missing)
Expected Outputs	Error: Fill in all your details
Observed Outputs	-
Verdict	-

Tested Requirement	Create a Virtufit account (Negative 2)
Quick Description	The user is unable to create an account on VirtuFit due to not meeting password requirements
Prerequisite	Email and username must not be associated with existing account
Inputs	Full Name, Phone number, email, password (Invalid)
Expected Outputs	Password does not meet minimum requirements
Observed Outputs	-
Verdict	-

Log in to VirtuFit

Tested Requirement	Log in to VirtuFit (Positive)
Prerequisite	User must have an existing account
Quick Description	The user can log in to VirtuFit with valid email and password
Inputs	Username / email, password
Expected Outputs	"You have been successfully logged in"
Observed Outputs	-
Verdict	-

Tested Requirement	Log in to VirtuFit (Negative 1)
--------------------	---------------------------------

Prerequisite	User must have an existing account
Quick Description	The user is unable to log in to VirtuFit due to invalid email / username
Inputs	Username / email, password
Expected Outputs	"This username / email does not exist"
Observed Outputs	-
Verdict	-

Tested Requirement	Log in to VirtuFit (Negative 2)
Prerequisite	User must have an existing account
Quick Description	The user is unable to log in to VirtuFit due to incorrect password
Inputs	Username / email, password
Expected Outputs	"This password is incorrect. Please try again."
Observed Outputs	-
Verdict	-

Converse with virtual trainer (chatbot)

Tested Requirement	Converse with chatbot (positive)
Prerequisite	User must be logged in

Quick Description	The user can converse with the chatbot regarding fitness related topics and receive satisfactory responses
Inputs	Text (user input on chat box)
Expected Outputs	Chatbot responses to user input
Observed Outputs	-
Verdict	-

Tested Requirement	Converse with chatbot (negative)
Prerequisite	User must be logged in
Quick Description	The user can converse with the chatbot regarding fitness related topics but does not receive satisfactory response
Inputs	Text (user input on chat box)
Expected Outputs	Chatbot responds to user input with "Sorry, i did not understand what you meant"
Observed Outputs	-
Verdict	-

Create schedule and workout plan

Tested Requirement	Create schedule and workout plan (positive)
Prerequisite	User must be logged in
Quick Description	The user provides the chatbot with details needed to build the schedule and workout plan
Inputs	Availability, goal, medical condition, height, and weight
Expected Outputs	Schedule and workout plan
Observed Outputs	-
Verdict	-

Tested Requirement	Create schedule and workout plan (negative)
Prerequisite	User must be logged in
Quick Description	The user provides the chatbot with details needed to build the schedule and workout plan however a missing detail is needed
Inputs	Availability, goal, medical condition, height, and weight (one at least is missing)
Expected Outputs	Chatbot asks user to provide the details

Observed Outputs	-
Verdict	-

Start Workout Session

Tested Requirement	Start Workout session
Prerequisite	User must be logged in
Quick Description	The user selects start workout and begins exercising
Inputs	User selection
Expected Outputs	Start workout page, camera, timer, and exercises
Observed Outputs	-
Verdict	-

View and Edit Schedule

Tested Requirement	View and edit schedule
Prerequisite	User must be logged in
Quick Description	The user views their schedule and can edit it by changing the days they will workout
Inputs	User selection

Expected Outputs	Schedule is updated
Observed Outputs	-
Verdict	-

Monitor Form

Squat:

Tested Requirement	Monitor Squat (Correct Squat)
Quick Description	if the user can do a proper squat it should increase the rep counter. It should match the Vert_HipAngle and Vert_KneeAngle conditions
Inputs	Stage, Vert_HipAngle, Vert_KneeAngle
Expected Outputs	NoOfReps Incremented
Observed Outputs	NoOfReps is Incremented
Verdict	Passed

Tested Requirement	Monitor Squat (Feedback: Lean Forward)
Quick Description	If User is standing straight, it should tell the user to lean forward. And there should be an interval before it repeats the feedback again
Inputs	Stage, Vert_HipAngle, Hip_Confidence
Expected Outputs	Text to speech is activated users are told to lean forward

Observed Outputs	Text to speech is activated users are told to lean forward
Verdict	Passed

Tested Requirement	Monitor Squat (Feedback: Lean Backward)
Quick Description	If the user is leaning too forward more than 44 degrees then it should give a feedback and tell users to lean backward
Inputs	Stage, Vert_HipAngle, Hip_Confidence
Expected Outputs	Text to speech is activated. Users are told to lean Backward
Observed Outputs	Text to speech is activated. Users are told to lean Backward
Verdict	Passed

Tested Requirement	Monitor Squat (Feedback: Lower your hips)
Quick Description	If the Users knee is not bent low for it to be considered a squat it should give feedback to the user to lower their hips.
Inputs	Stage, Vert_KneeAngle, Knee_Confidence
Expected Outputs	Text to speech is activated. Users are told to lower your hips
Observed Outputs	Text to speech is activated. Users are told to lower your hips
Verdict	Passed

Bicep Curl:

Tested Requirement	Monitor Squat (Correct Curl)
Quick Description	If the User curls with his left or right arm it should increment the rep counter for each arm separately
Inputs	StageL, StageR, AngleL, AngleR
Expected Outputs	NoOfReps increment based on whether the left or right hand moves up and down
Observed Outputs	NoOfReps increment based on whether the left or right hand moves up and down
Verdict	Passed

Tested Requirement	Monitor Squat (Feedback: Arms Close)
Quick Description	If the Users elbow is far from the chest, then it should give a feedback to users to keep their arm close to their chests
Inputs	Put_Arms_CClose
Expected Outputs	Warn users to keep their arm close to their chests using Text to speech
Observed Outputs	Warn users to keep their arm close to their chests using Text to speech
Verdict	Passed

Push up:

Tested Requirement	Monitor Squat (Correct Push up)
Quick Description	If the Users does a correct it should increment push up counter and Tech to speech to inform users, they have done a correct push.
Inputs	Stage, AngleL, AngleR
Expected Outputs	Increment pushes up counter and activates text to speech to inform users they have done a correct push up
Observed Outputs	Increment pushes up counter and activates text to speech to inform users they have done a correct push up
Verdict	Passed

Tested Requirement	Monitor Squat (Feedback: Arms straight)
Quick Description	If the users both arms are not straight it will provide feedback to user to keep both of his arms straight when doing a push up.
Inputs	Arms_Staright , AngleL, AngleR
Expected Outputs	ArmsStraight Counter is incremented and Text to speech is activated warning users to keep their arms straight
Observed Outputs	ArmsStraight Counter is incremented and Text to speech is activated warning users to keep their arms straight
Verdict	Passed

Wall Sit:

Tested Requirement	Monitor Squat (Correct Wall Sit)
Quick Description	If the Users does a correct wall sit position and holds that position it will increment the timer module every second accurately. While the user is holding that position. When user stops holding the wall sit position it should stop the timer module automatically
Inputs	Knee Angle
Expected Outputs	Increment the timer while the user is holding the wall sit position.
Observed Outputs	Increment the timer while the user is holding the wall sit position.
Verdict	Passed

Tested Requirement	Monitor Squat (Feedback: Go lower)
Quick Description	If the user is in the and he is not low enough to start the timer module. The text to speech is activating informing users to go lower
Inputs	Go_lower, Knee_Angle
Expected Outputs	Inform users to go lower if they are no in the proper wall sit position
Observed Outputs	Inform users to go lower if they are no in the proper wall sit position
Verdict	Passed

Dumbbell Deadlifts:

Tested Requirement	Monitor (Correct Dumbbell Deadlift)
Quick Description	If the User Follows the feedback and performs a correct rep it will increment the rep counter.
Inputs	Stage, LeftWrist, LeftKnee, RightWrist, RightKnee
Expected Outputs	Increment the rep counter by 1 when user arms are straight and proceeds to lean forward
Observed Outputs	Rep counter is incremented.
Verdict	Passed

Tested Requirement	Monitor (Feedback:Arms Straight)
Quick Description	If user arm is not straight, it will increment the arms straight Value which will inform the user to straighten their arm using text to speech.
Inputs	AngleL, AngleR, arms straight
Expected Outputs	Increment the ArmsStraight value

Observed Outputs	Arms straight value is incremented
Verdict	Passed

Tested Requirement	Monitor (Feedback: Put Arms Close)
Quick Description	If user arm is far apart it will increment the put_arms_close value which will warn the user with text to speech to put their arms together
Inputs	LeftShoulder, LeftElbow, RightShoulder, RightElbow, put arms close
Expected Outputs	Increment put arms close value
Observed Outputs	Put arms close value is incremented
Verdict	Passed
Tested Requirement	Monitor (Feedback: Go lower)
Quick Description	If the user is standing still will provide feedback to go lower and increment the go lower value
Inputs	LeftWrist, LeftKnee, RightWrist, RightKnee, Go lower
Expected Outputs	Increment the go lower value

Observed Outputs	Go lower value
Verdict	Passed

Jumping Jack:

Tested Requirement	Monitor (Correct Jumping Jack)
Quick Description	If the user performs a proper jumping jack the wrist is above the shoulder and legs are far apart then it will increment the rep counter.
Inputs	Stage, leftWrist, LeftShoulder, Right wrist, Right shoulder, difference
Expected Outputs	Increment the rep counter value
Observed Outputs	Rep counter value is incremented
Verdict	Passed

Tested Requirement	Monitor (Feedback: Align Legs)
--------------------	--------------------------------

Quick Description	If the users legs are not in the same y axis then it will warn user to align their legs.
Inputs	ankleL, ankleR, Align Leg
Expected Outputs	Increment Align leg value
Observed Outputs	Align leg value is incremented
Verdict	Passed

Lateral Raises:

Tested Requirement	Monitor (Correct Lateral Raises)
Quick Description	The user is asked to perform the lateral raise exercise. If done correctly will increment the rep counter
Inputs	Stage
Expected Outputs	Increment the rep counter value

Observed Outputs	Rep counter value is incremented
Verdict	Passed

Tested Requirement	Monitor (Feedback: Arms straight)
Quick Description	If the users both arms are not straight it will provide feedback to user to keep both of his arms straight when doing a lateral raise.
Inputs	Arms_Staright , AngleL, AngleR
Expected Outputs	ArmsStraight Counter is incremented
Observed Outputs	ArmsStraight Counter is incremented and Text to speech is activated warning users to keep their arms straight
Verdict	Passed

Tested Requirement	Monitor (Feedback: Raise Arm)
Quick Description	If the user's arms are down and stationary it will provide feedback to raise arms.
Inputs	Elbow L, Shoulder L, Elbow R, Shoulder R

Expected Outputs	Raise arm feedback should be incremented
Observed Outputs	Raise arm feedback is incremented and Text to speech is used to provide feedback to user.
Verdict	Passed

Shoulder press:

Tested Requirement	Monitor (Correct Shoulder Press)
Quick Description	Once a correct shoulder is completed the rep counter is incremented
Inputs	Stage, AngleL, AngleR
Expected Outputs	Increment the rep counter value
Observed Outputs	Rep counter value is incremented
Verdict	Passed

Tested Requirement	Monitor (Feedback: Adjust Arm)
Quick Description	If user's arm is down and is not holding the correct form then it will increment the adjust arm value. Which will inform the user to adjust elbow in a shoulder press position.
Inputs	rightWrist, rightShoulder, leftwrist, leftshoulder, adjust arms
Expected Outputs	Adjust arm should be incremented and text to speech should inform the user.
Observed Outputs	Adjust arm is incremented and text to speech is activated to inform user and provide feedback
Verdict	Passed

Side Plank:

Tested Requirement	Monitor (Correct Side Plank)
Quick Description	If the user is leaning sideways in a plank position where the left shoulder is on a higher y axis than the right shoulder or vice versa it will increment the timer.
Inputs	ShoulderL, ShoulderR, User Visibility

Expected Outputs	Timer should increment and counting is started
Observed Outputs	Timer is started
Verdict	Passed

Time Based General Exercise:

This section includes all time-based cardio exercise where the time starts when the user is inside the frame and stops if user leaves the frame. Some of these exercise include burpees, Side heel squat, Butt Kicks, and High Knees.

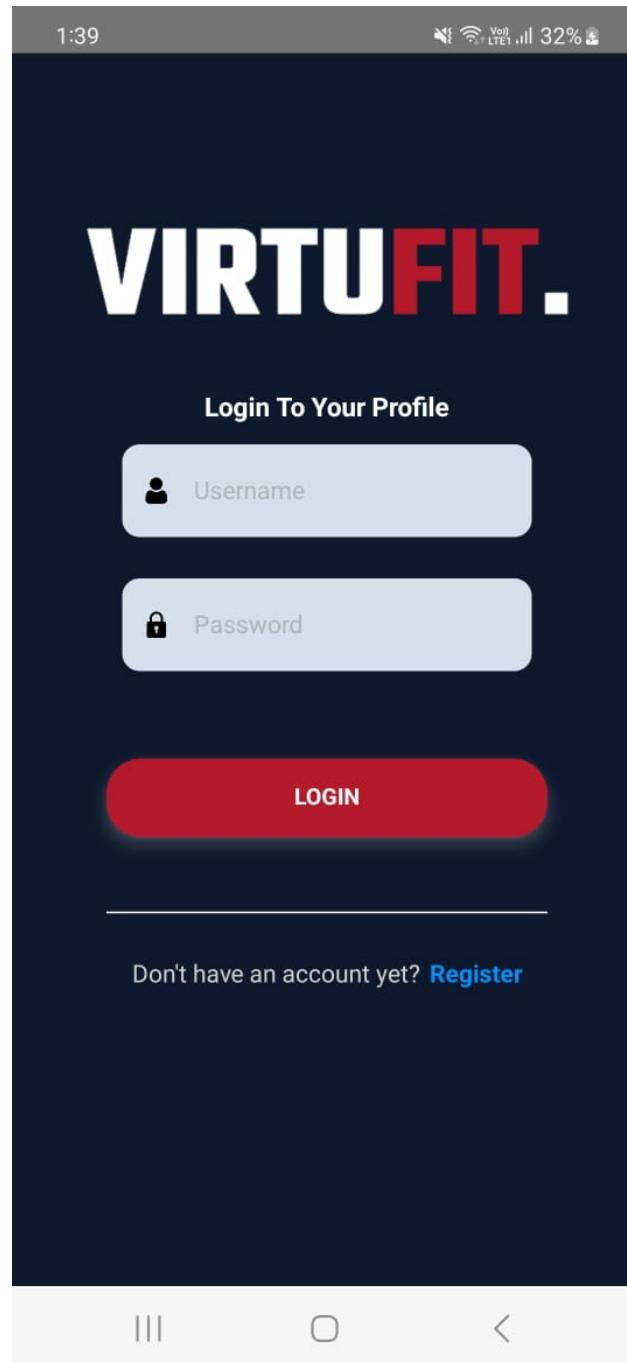
Tested Requirement	Timed Exercise (Start Button)
Quick Description	The user has to remain in the frame and press the start button then the timer will begin, and user can start exercise. At any time, the user can leave and the timer will stop automatically.
Inputs	buttonName, Shoulder_Confidence, user_start, user_visibility
Expected Outputs	Timer should increment and counting is started if user in frame and if the user is not in frame the timer will stop.
Observed Outputs	Timer is started

Verdict	Passed
---------	--------

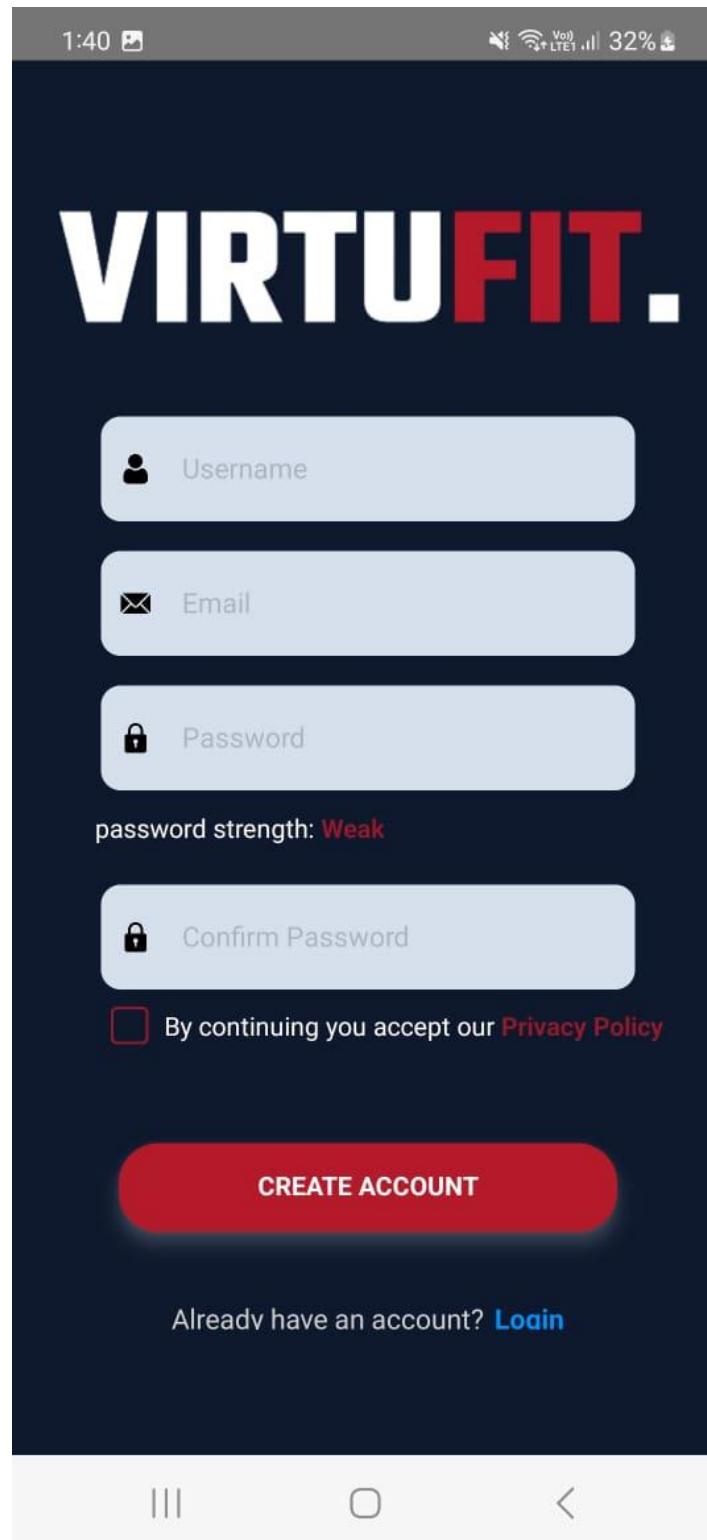
Tested Requirement	Timed Exercise (Pause button)
Quick Description	When pause button is pressed it does not matter if the user is in or out of the frame the timer counter will not increment.
Inputs	buttonName, Shoulder_Confidence, user_start, user_visibility
Expected Outputs	Timer should not increment regarding of whether user is in or out of the frame.
Observed Outputs	Timer is stopped
Verdict	Passed



Application Screens: Login



Register

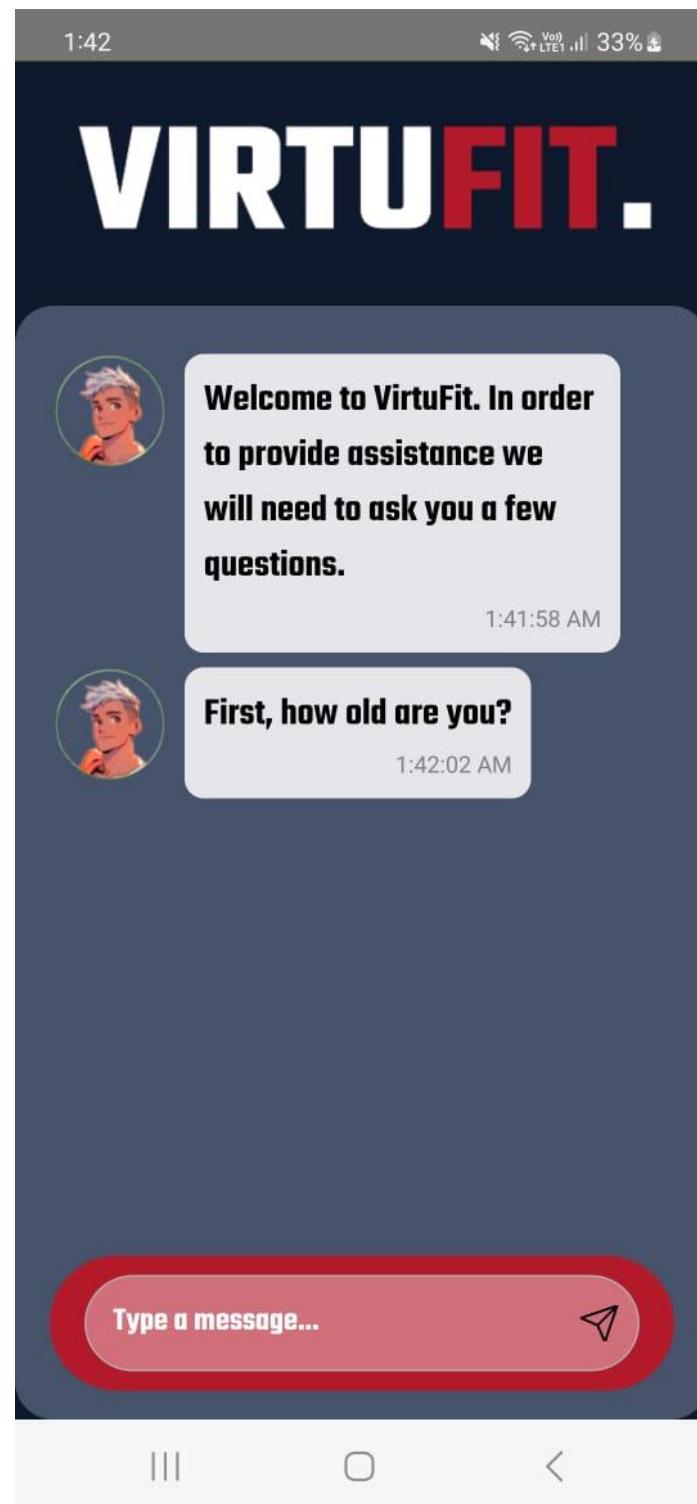


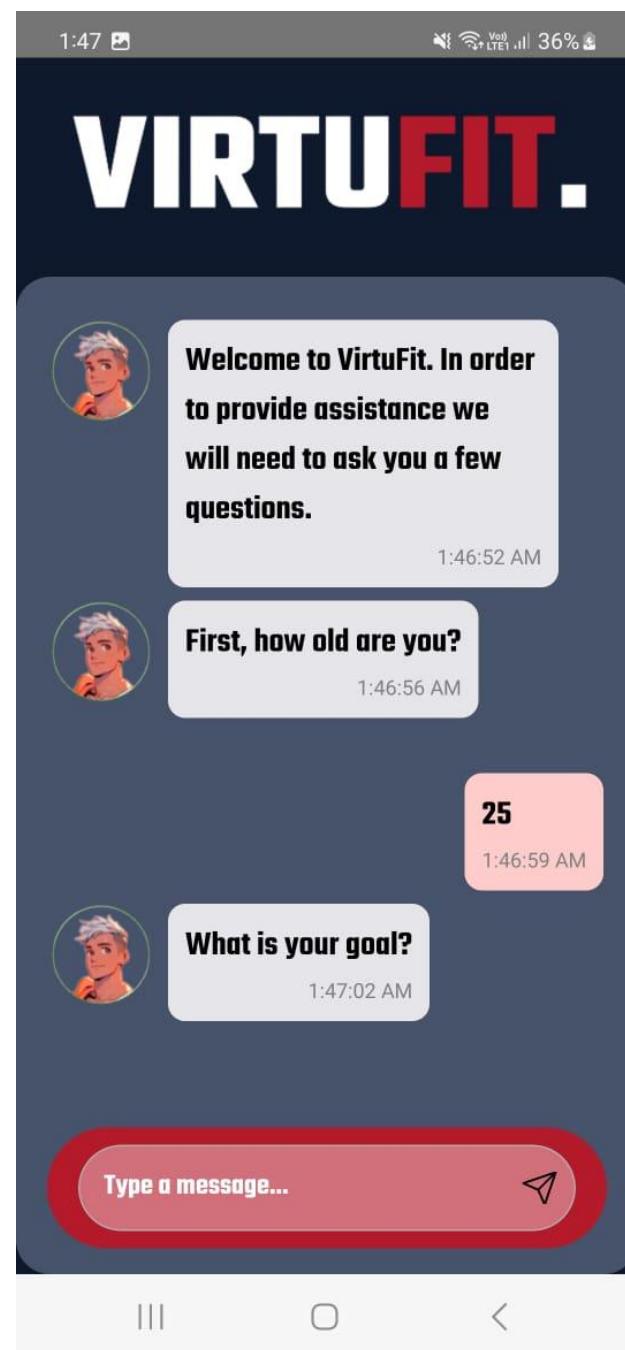
Choose Trainer

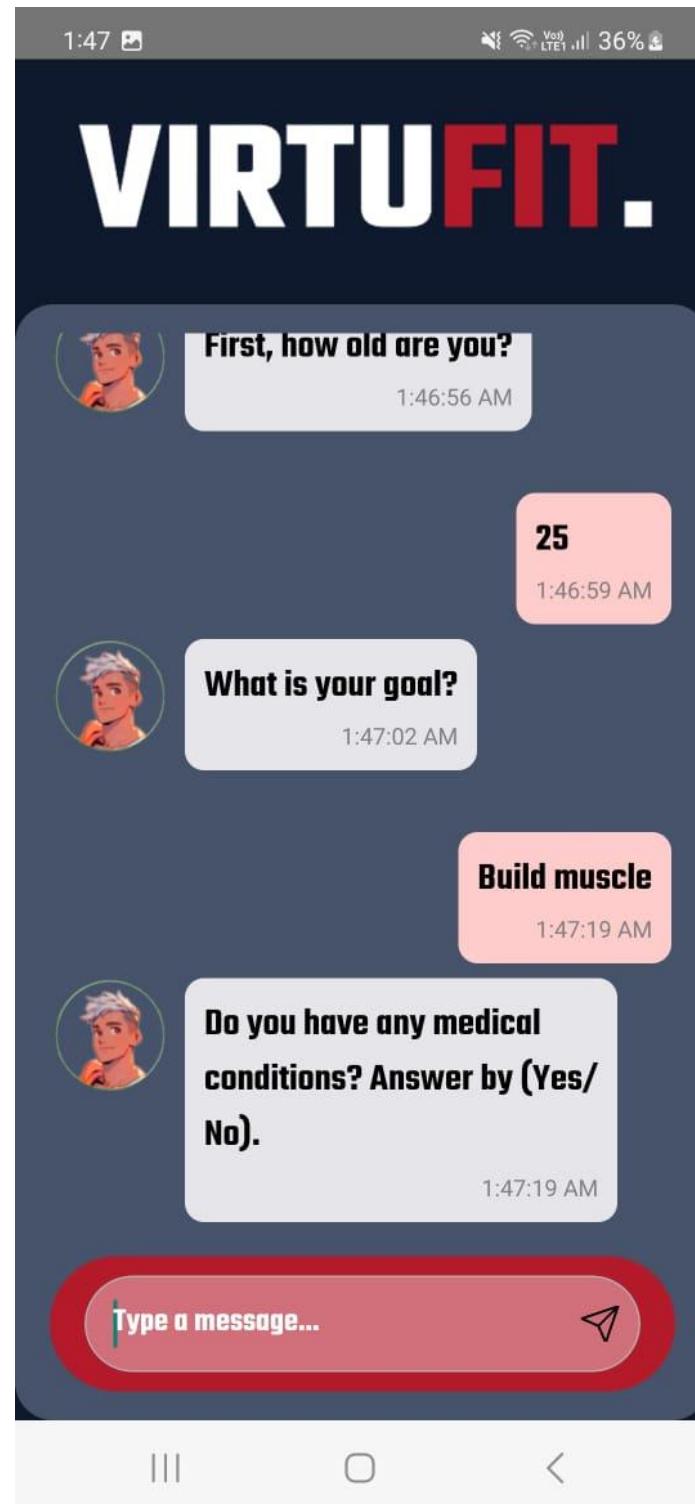


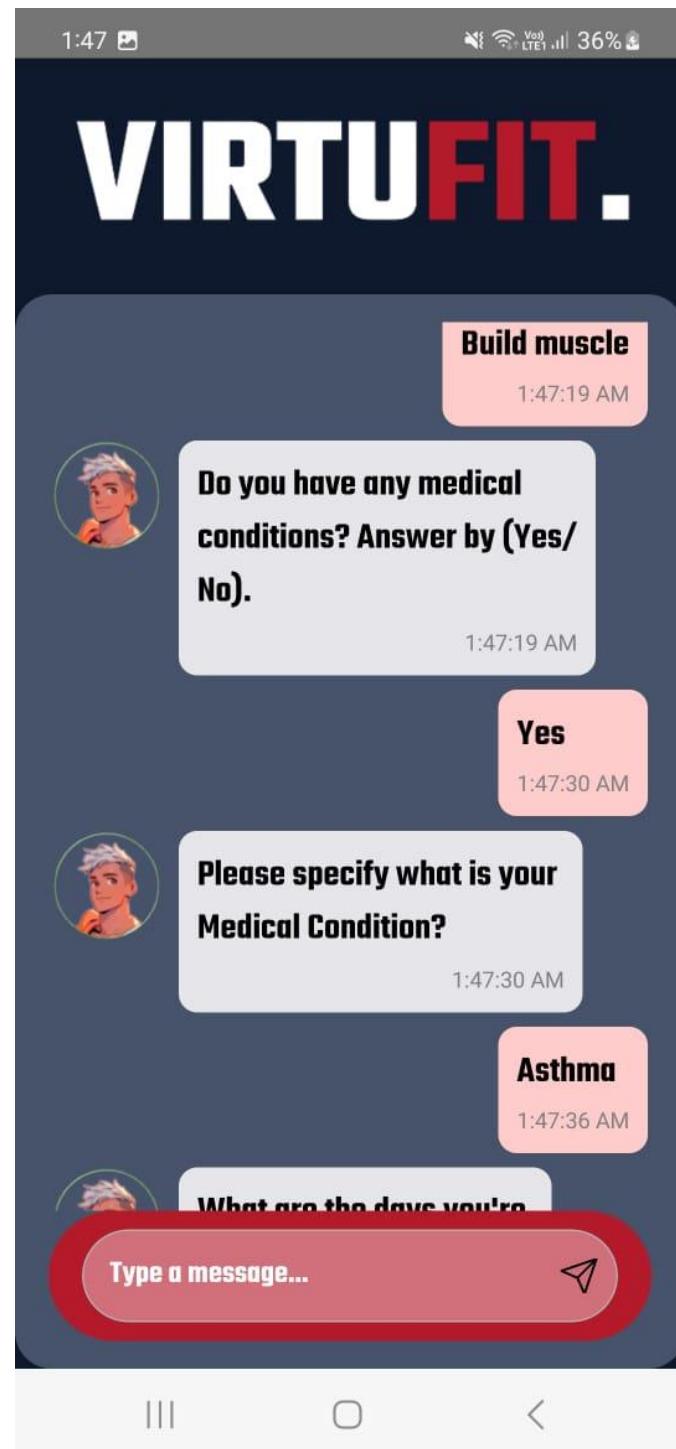


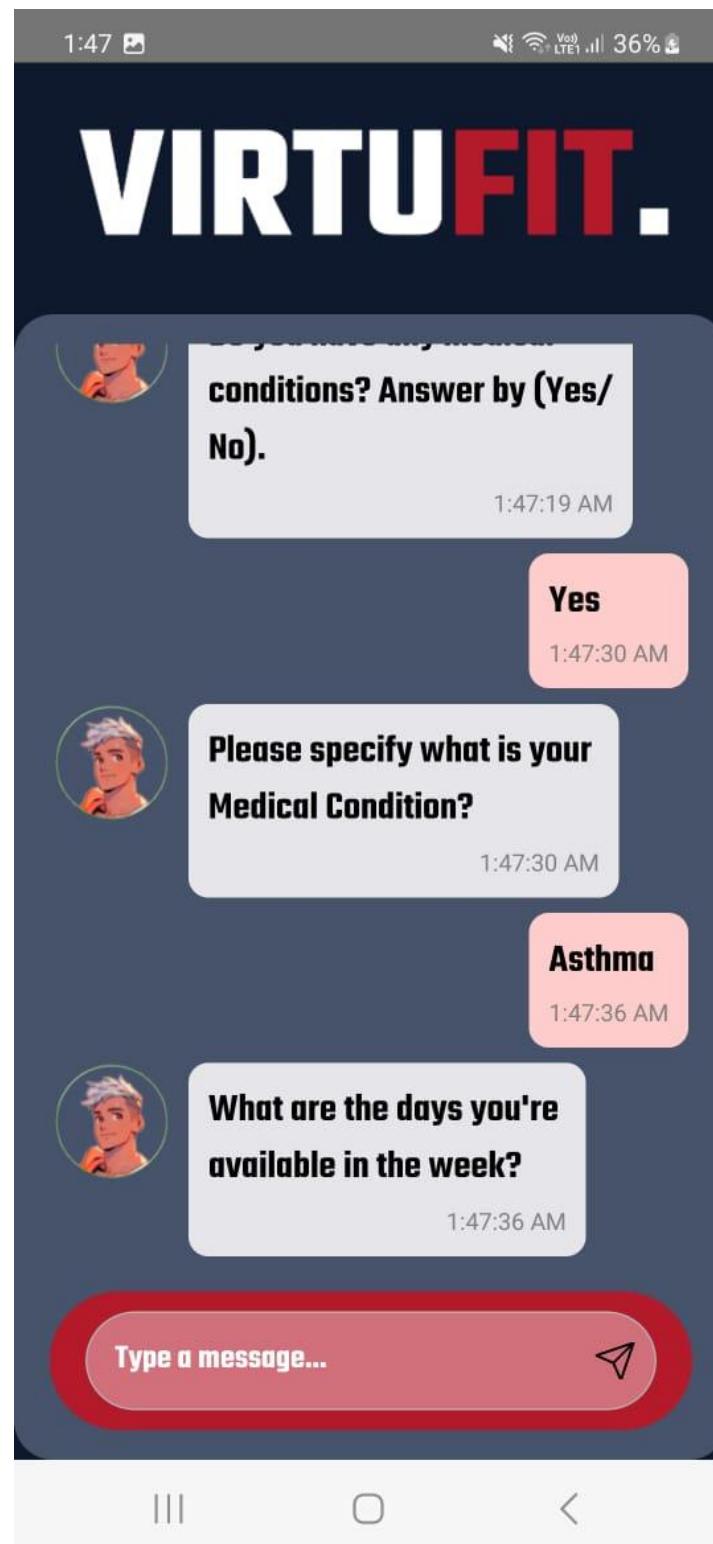
Beginning Chatbot





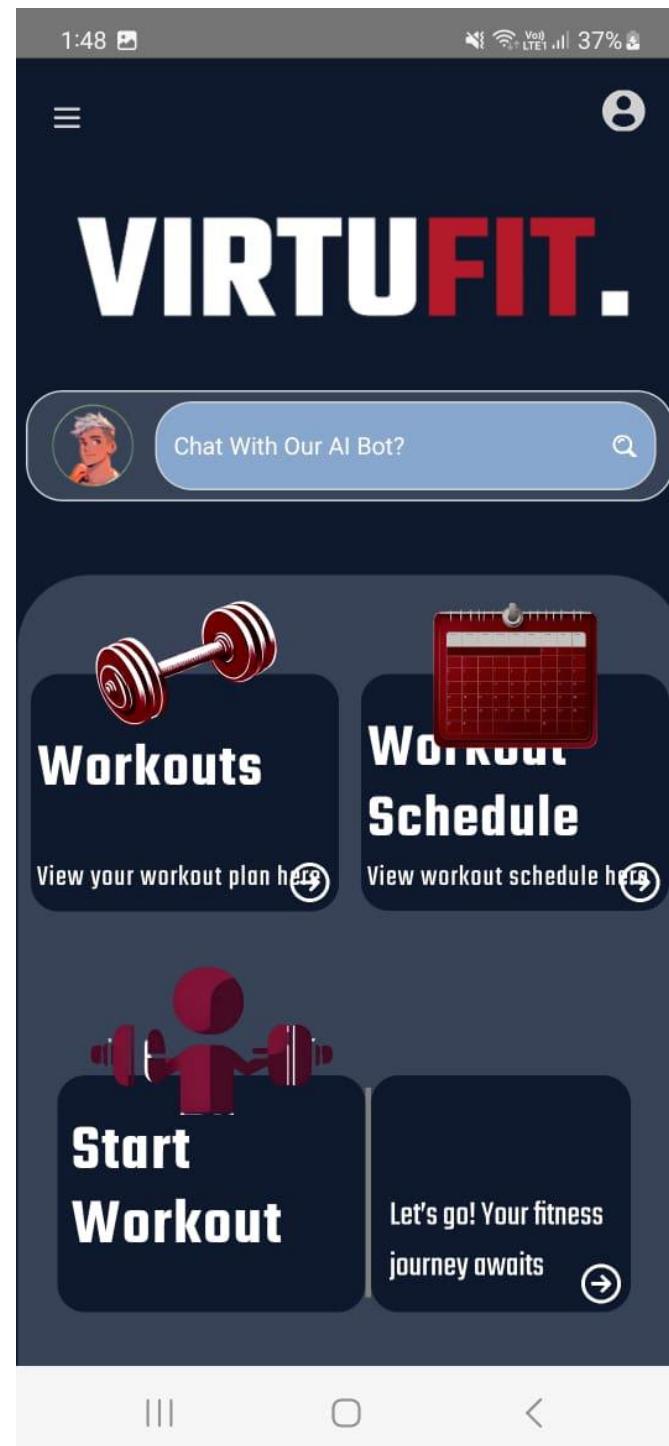




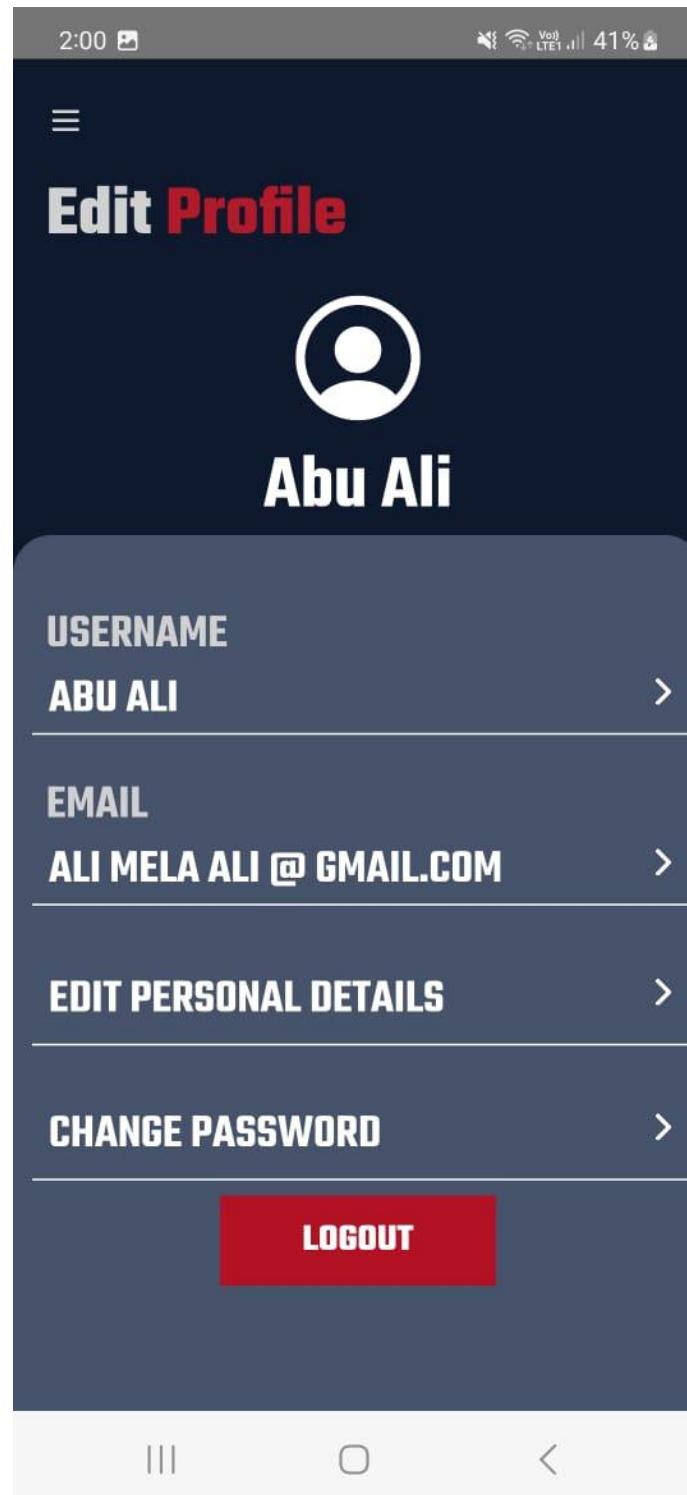




Home Page

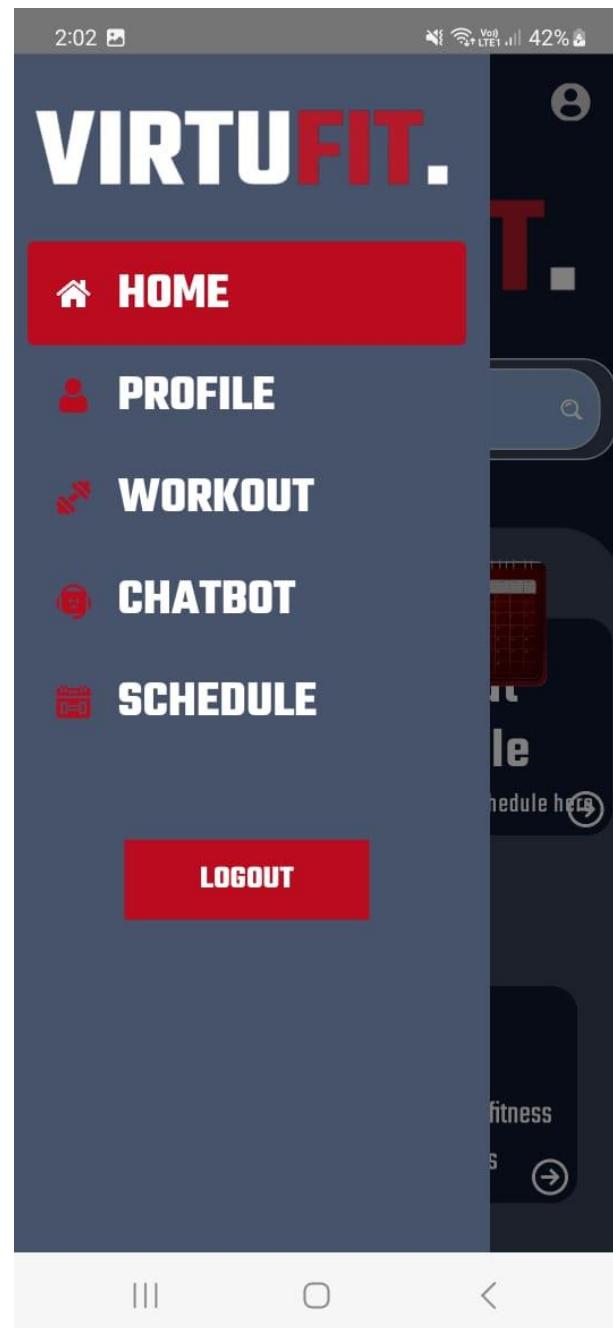


Profile Page



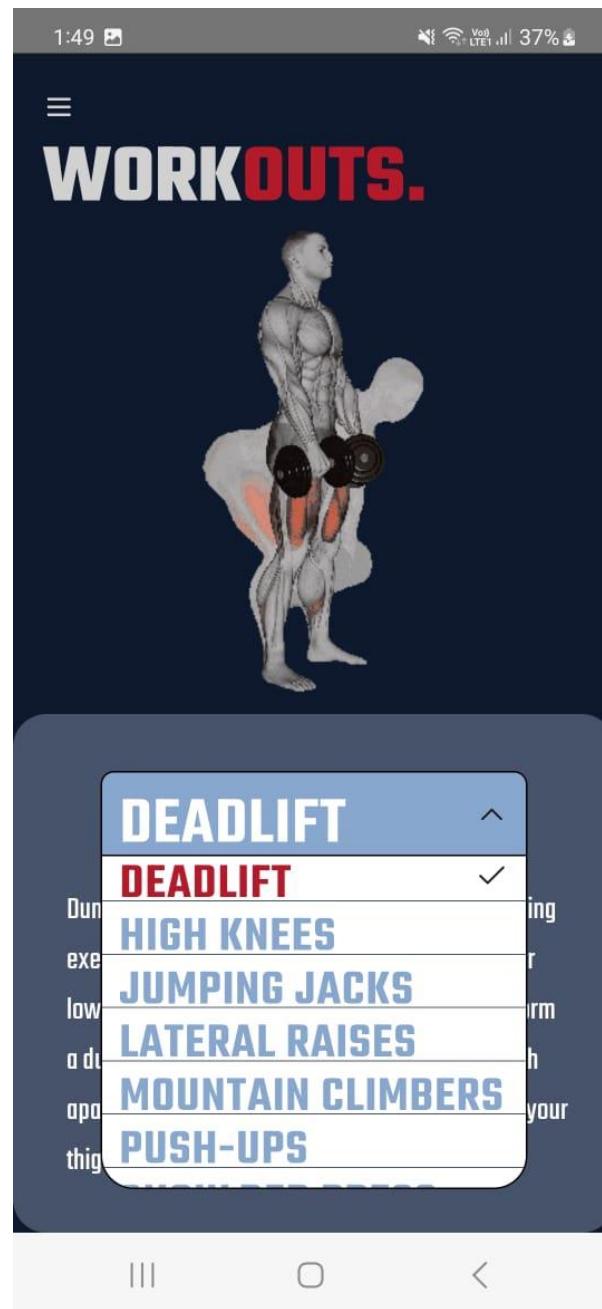


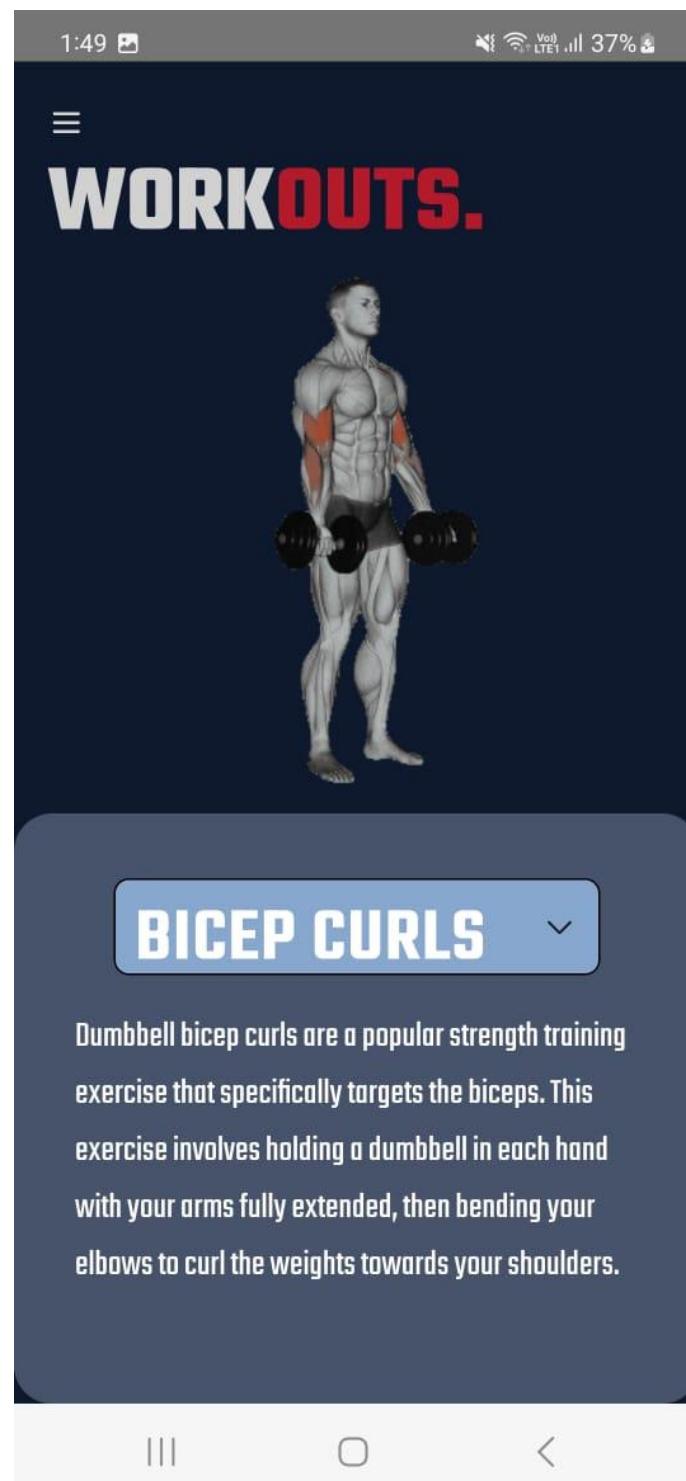
Side Bar



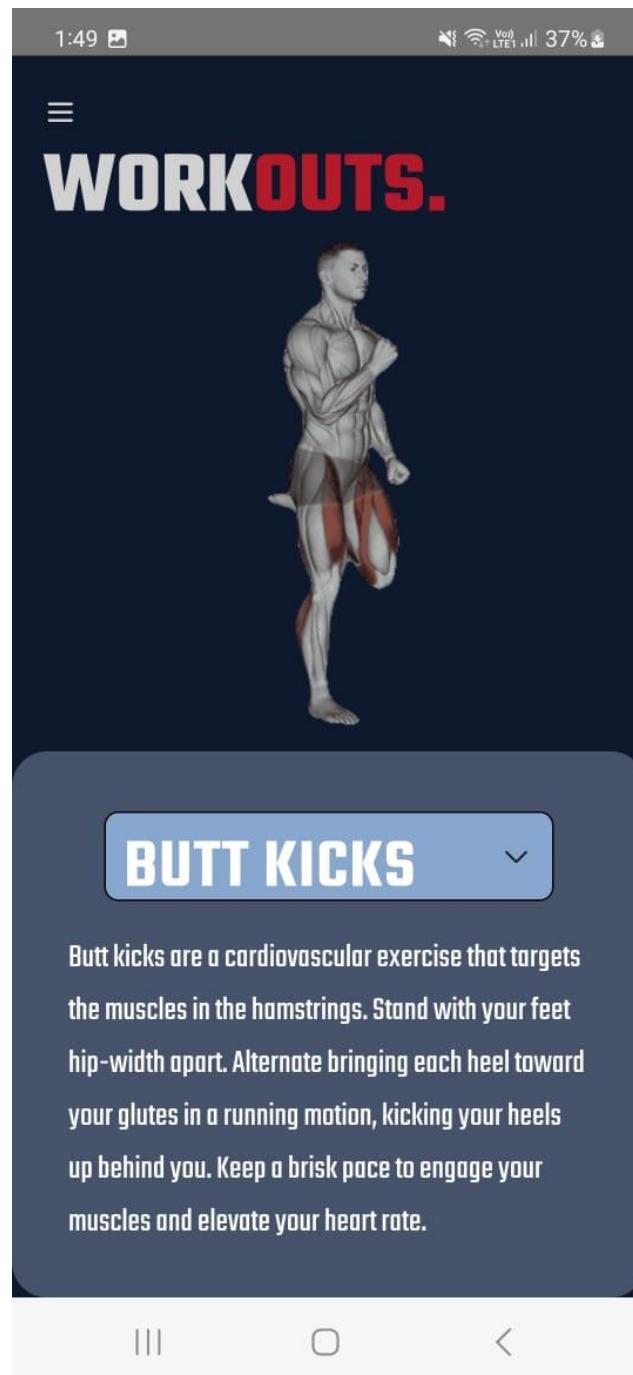


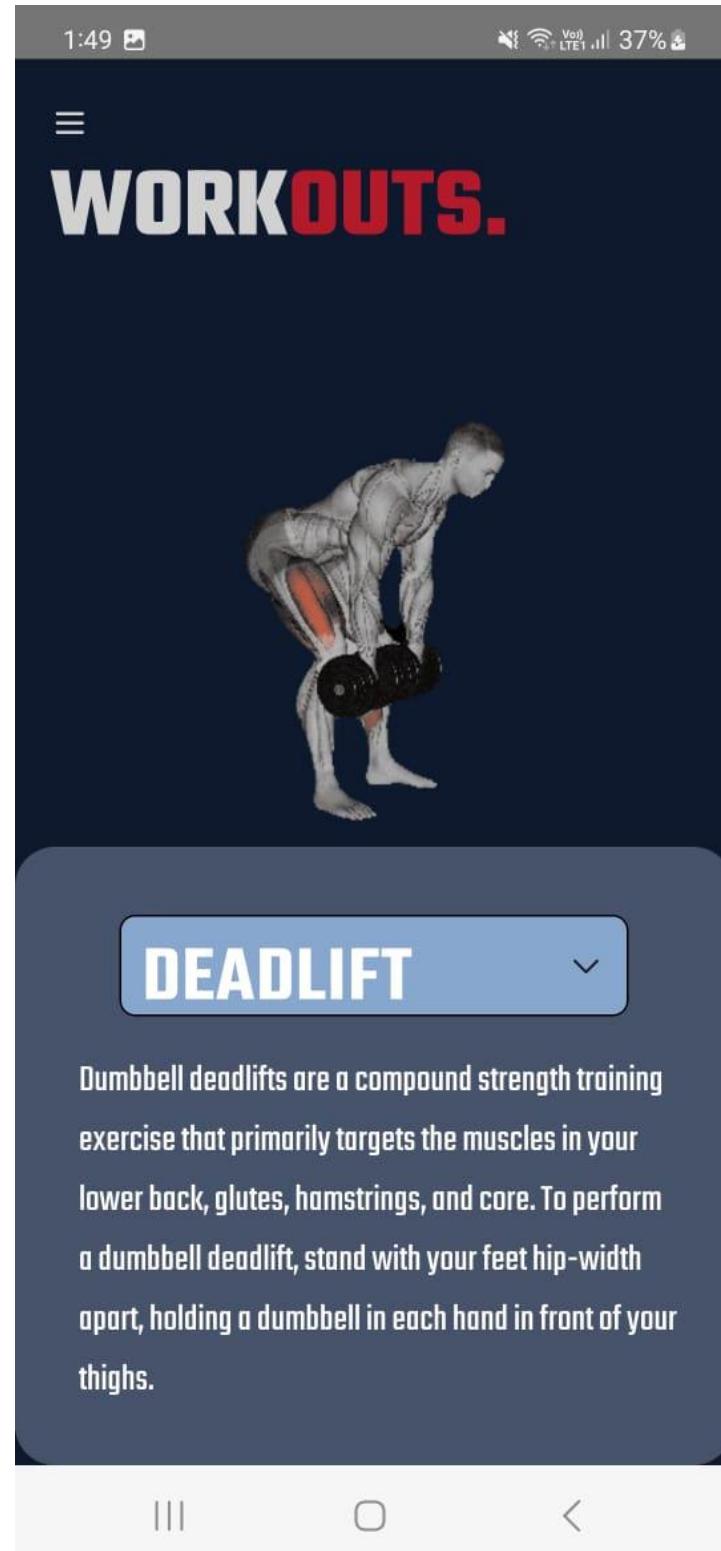
Workouts Description

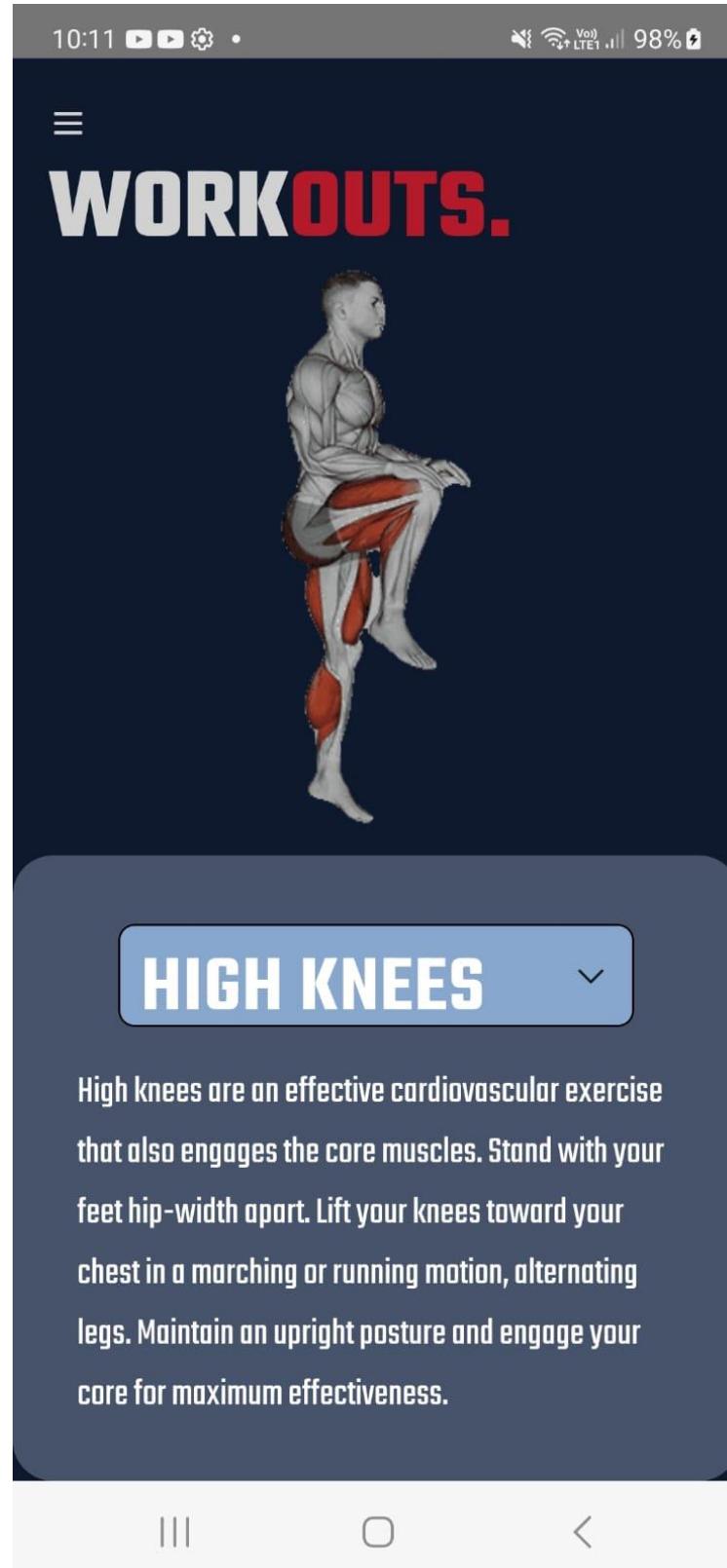












The image shows a smartphone screen displaying a fitness application. At the top, there is a navigation bar with icons for time (10:11), notifications, battery level (98%), and signal strength. Below the navigation bar is a menu icon (three horizontal lines) and a large, bold title "WORKOUTS." where "WORKOUTS." is split into "WORKOUTS" in white and "WORKOUTS." in red. The main content area features a 3D anatomical illustration of a male torso and legs in profile, with the quadriceps muscle highlighted in red. Below the illustration is a blue rounded rectangle containing the text "HIGH KNEES" in white capital letters. To the right of the text is a small downward-pointing arrow. The text below the title provides instructions for the "HIGH KNEES" exercise: "High knees are an effective cardiovascular exercise that also engages the core muscles. Stand with your feet hip-width apart. Lift your knees toward your chest in a marching or running motion, alternating legs. Maintain an upright posture and engage your core for maximum effectiveness." At the bottom of the screen are three control icons: three vertical lines, a square, and a left-pointing arrow.

10:11

98%

≡

WORKOUTS.



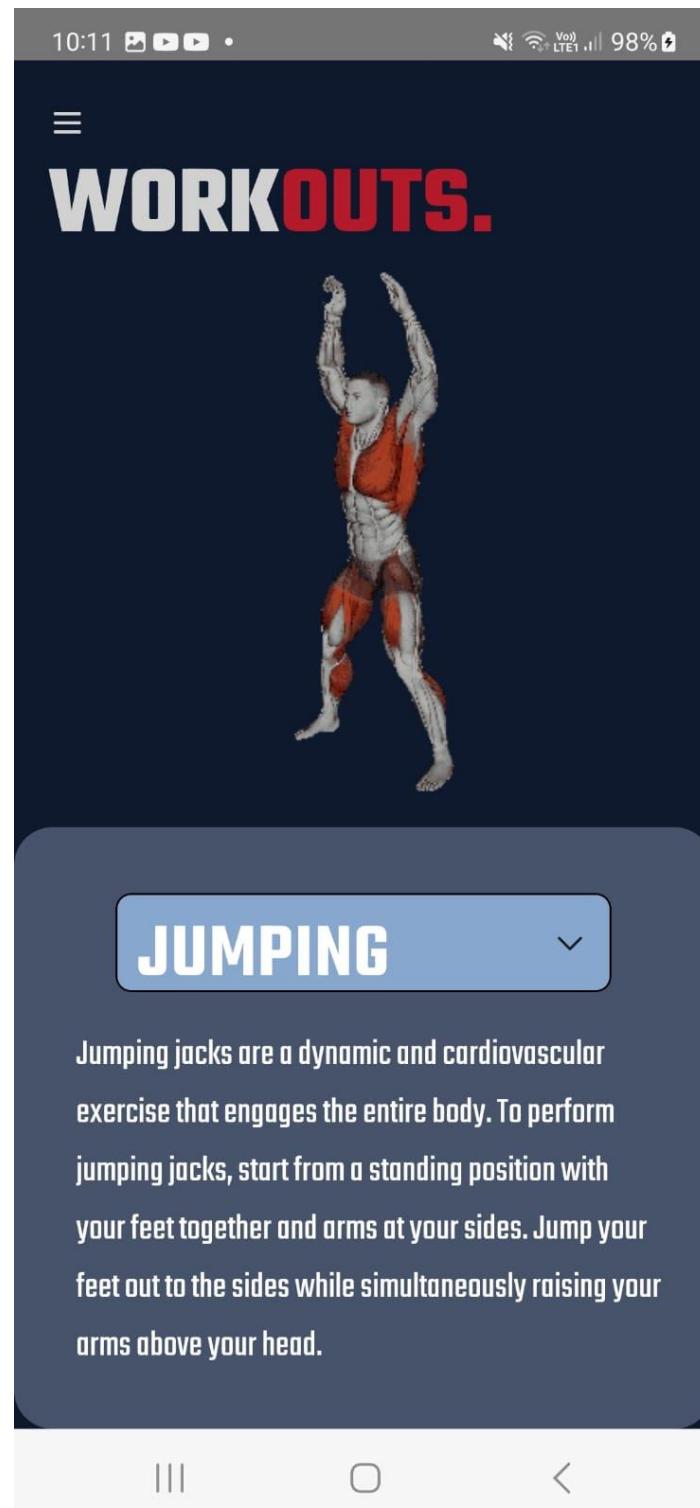
HIGH KNEES

High knees are an effective cardiovascular exercise that also engages the core muscles. Stand with your feet hip-width apart. Lift your knees toward your chest in a marching or running motion, alternating legs. Maintain an upright posture and engage your core for maximum effectiveness.

|||

□

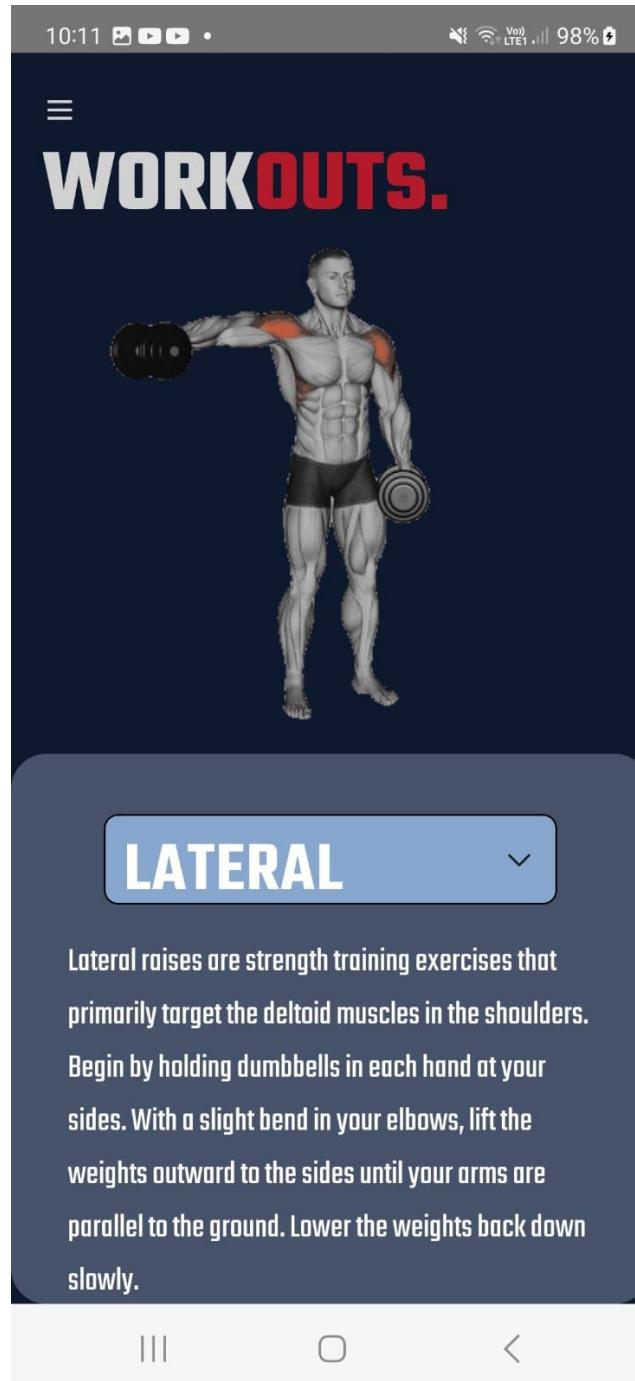
<

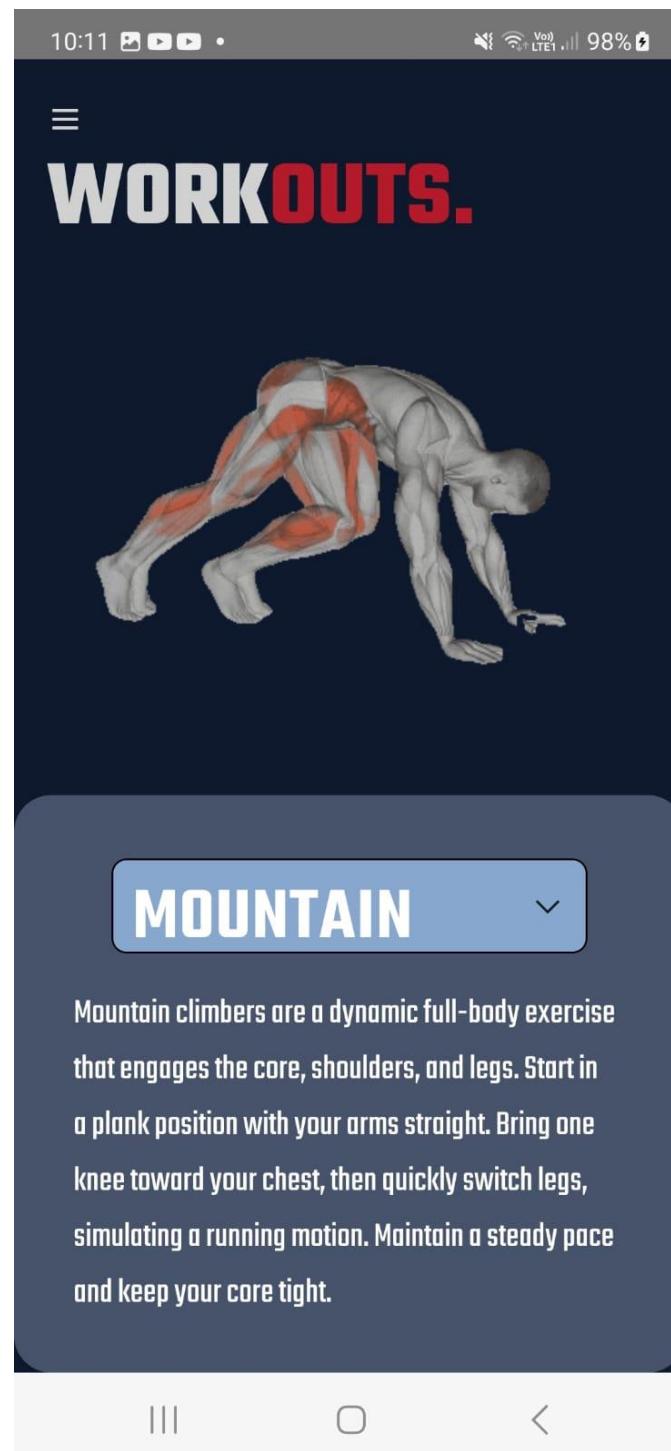


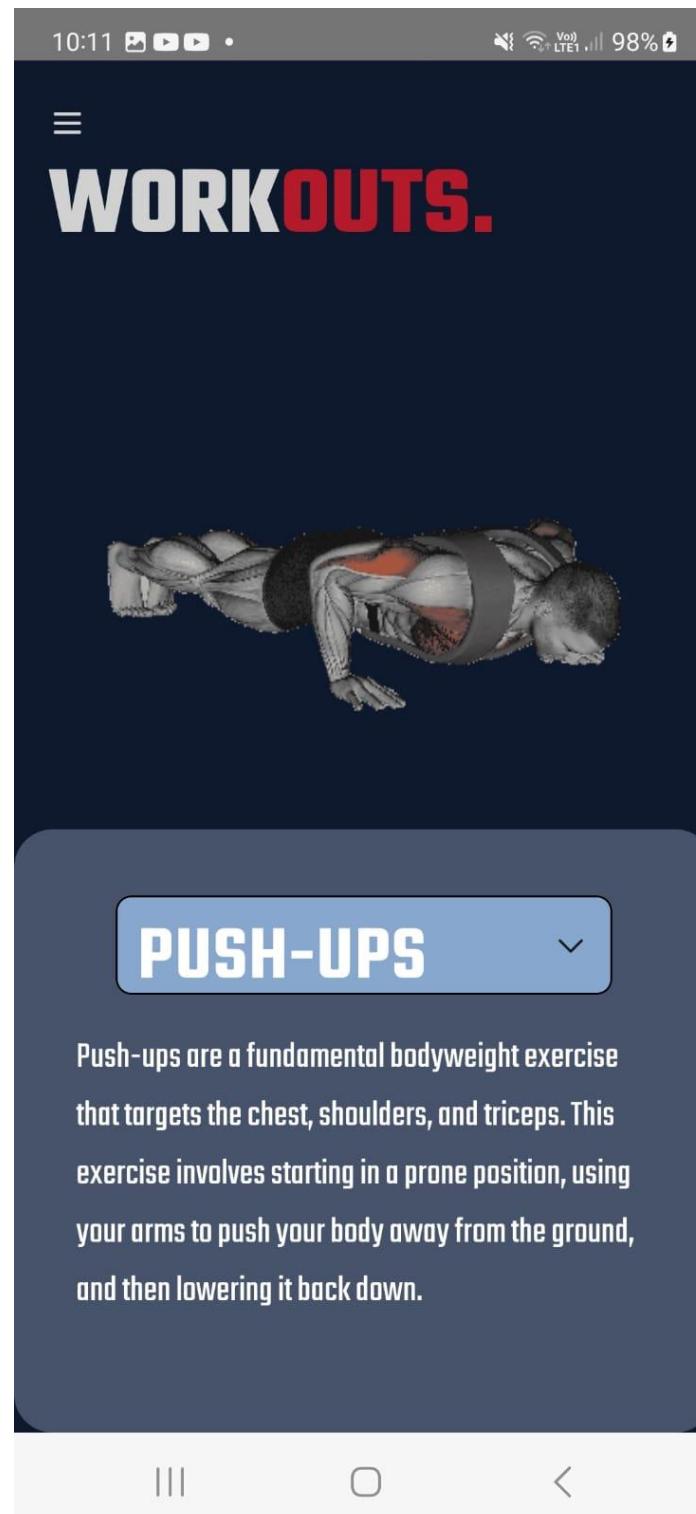
The image shows a smartphone screen displaying a fitness application. The top status bar indicates the time as 10:11, battery level at 98%, and signal strength. Below the status bar, there is a menu icon (three horizontal lines) and a large red header that reads "WORKOUTS.". In the center of the screen is a 3D anatomical model of a male torso and legs, with muscle groups highlighted in red. Below this, a blue rounded rectangle contains the word "JUMPING" in white capital letters. To the right of "JUMPING" is a small downward-pointing arrow. A dark blue callout box contains the following text:

Jumping jacks are a dynamic and cardiovascular exercise that engages the entire body. To perform jumping jacks, start from a standing position with your feet together and arms at your sides. Jump your feet out to the sides while simultaneously raising your arms above your head.

At the bottom of the screen are three navigation icons: three vertical lines, a square, and a left-pointing arrow.







10:11  98% 

≡

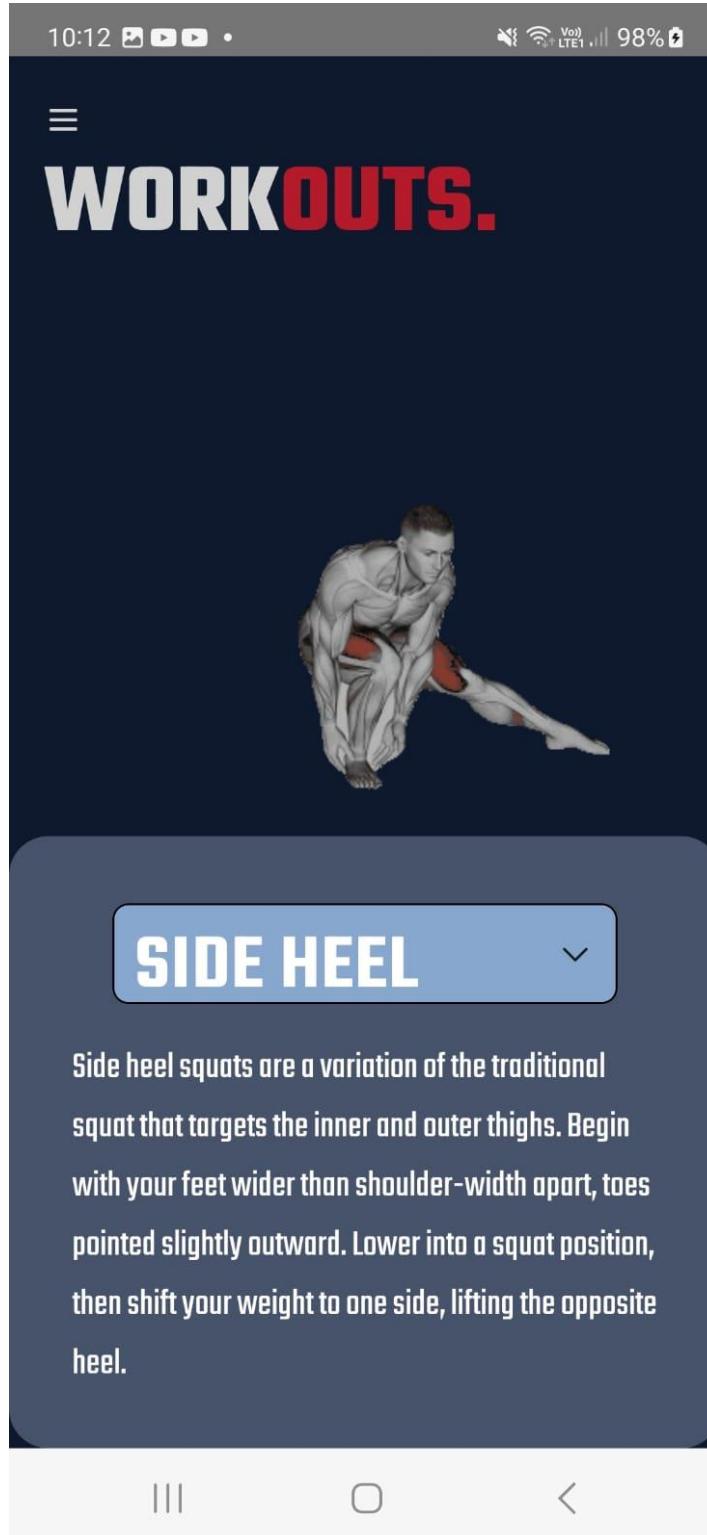
WORKOUTS.



SHOULDER 

The shoulder press, also known as the overhead press, is a fundamental strength training exercise that targets the muscles in the shoulders and upper arms. To perform a shoulder press, start by holding a dumbbell in each hand at shoulder height, palms facing forward. Stand with your feet hip-width apart.

|||  <



The image shows a smartphone screen displaying a workout application. At the top, the status bar shows the time as 10:12, signal strength, battery level at 98%, and other icons. Below the status bar, there is a dark header with a menu icon (three horizontal lines) and the word "WORKOUTS." in large, bold, white and red capital letters. In the center of the screen, there is a 3D rendering of a muscular male figure performing a side heel squat. Below this image, a blue rounded rectangle contains the text "SIDE HEEL" in white capital letters. To the right of the text is a small downward-pointing arrow. A dark blue rectangular box contains a detailed description of the exercise: "Side heel squats are a variation of the traditional squat that targets the inner and outer thighs. Begin with your feet wider than shoulder-width apart, toes pointed slightly outward. Lower into a squat position, then shift your weight to one side, lifting the opposite heel." At the bottom of the screen, there are three white navigation icons: three vertical lines, a square, and a left arrow.

SIDE HEEL

Side heel squats are a variation of the traditional squat that targets the inner and outer thighs. Begin with your feet wider than shoulder-width apart, toes pointed slightly outward. Lower into a squat position, then shift your weight to one side, lifting the opposite heel.

10:12 VoIP 98%

≡

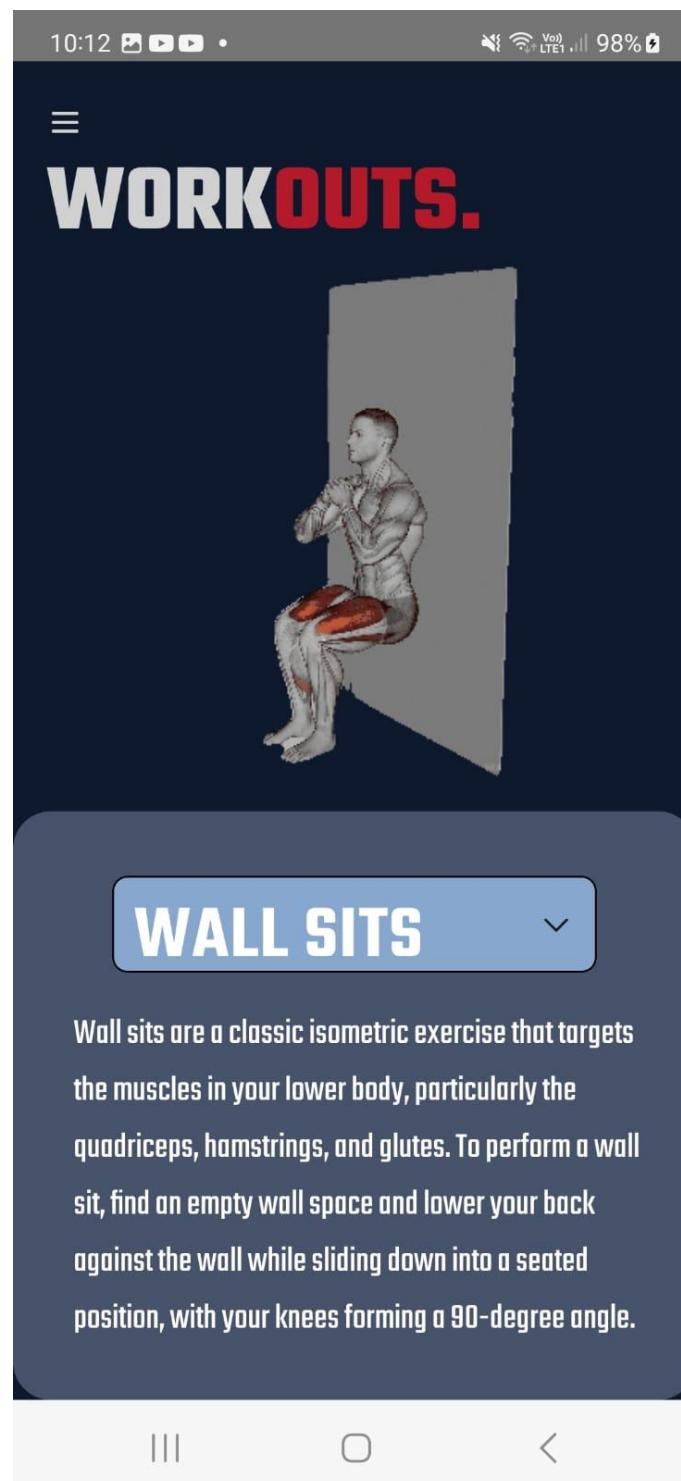
WORKOUTS.



SIDE LUNGES 

Side lunges are functional exercises that work the muscles in the thighs and hips. Start with your feet together, then take a wide step to the side, bending one knee while keeping the other leg straight. Push off the bent leg and return to the starting position.
Alternate sides.

||| <





Workout Schedule Page

The image shows a smartphone screen displaying a workout schedule application. The top status bar indicates the time as 1:49, signal strength, battery level at 37%, and a pen icon. Below the status bar, there is a navigation bar with three horizontal lines on the left and a pen icon on the right. The main title "WORKOUT SCHEDULE" is displayed in large, bold, white and red letters. The days of the week are listed below it, each with a right-pointing arrow indicating further details: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. At the bottom of the screen, there is a light gray footer bar with three icons: three vertical lines (likely a menu or list icon), a square (likely a home or recent apps icon), and a less than sign (likely a previous/next page or search icon).

WORKOUT SCHEDULE

Sunday >

Monday >

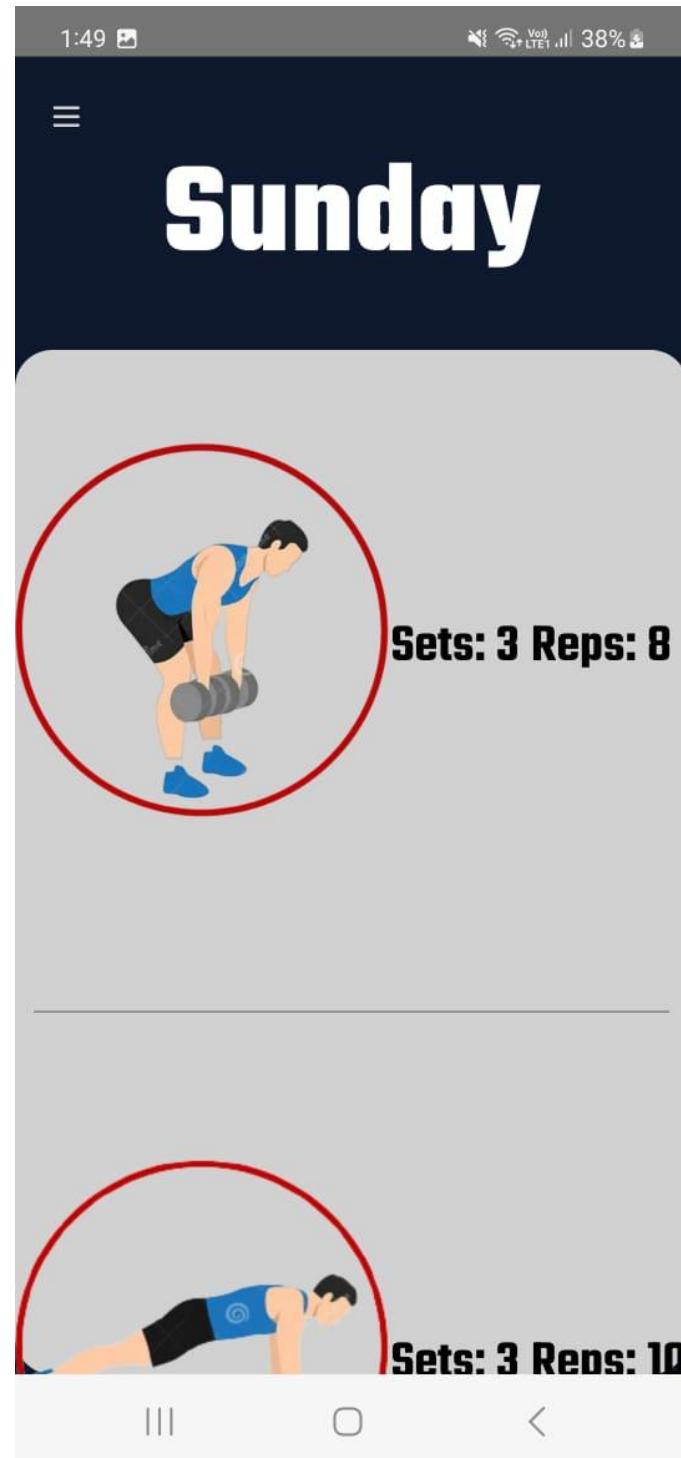
Tuesday >

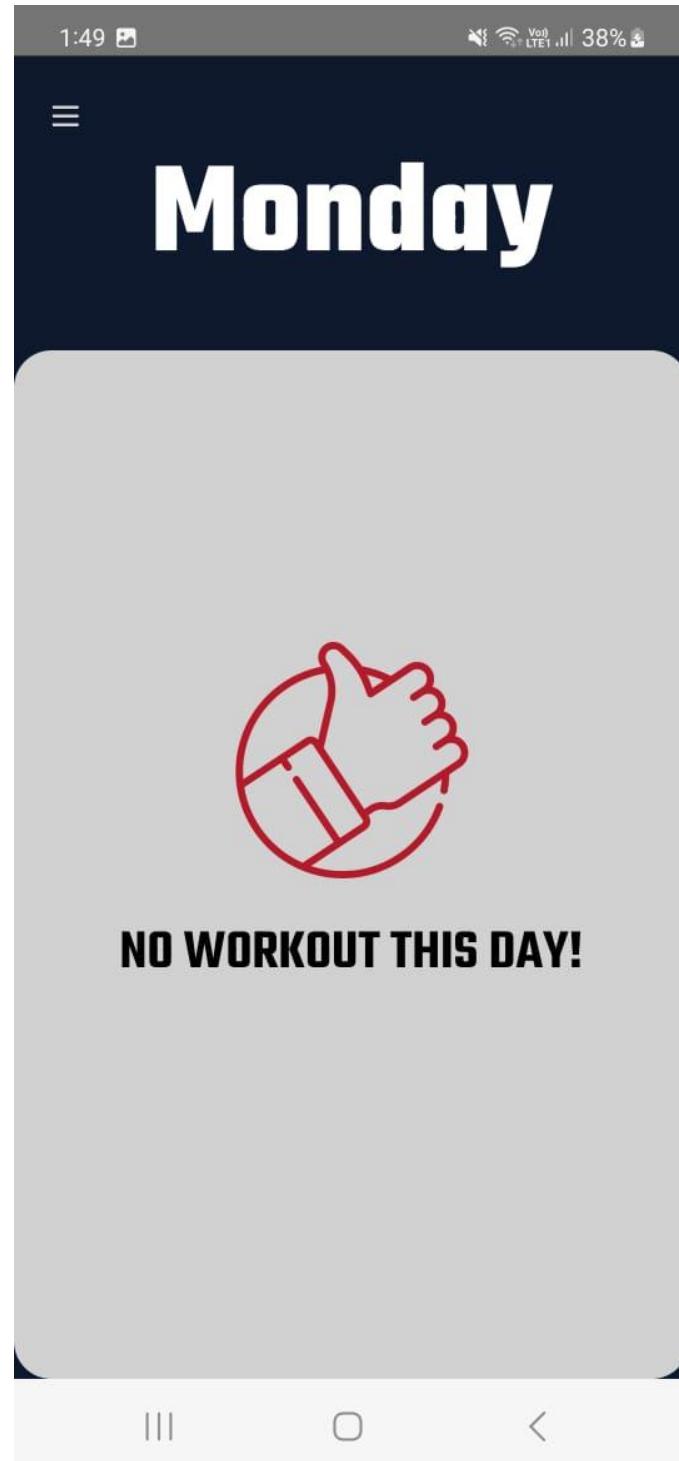
Wednesday >

Thursday >

Friday >

Saturday >







Start Workout Page

