
Software Requirements Specification

for

WITFIT

Version 1.0 approved

Prepared by WomenInTech

WomenInTech (CZ2006 BS3)

13 Apr 2022

Table of Contents

1	Introduction	1
1.1	Purpose	1
1.2	Document Conventions	1
1.3	Intended Audience and Reading Suggestions	1
1.4	Product Scope	1
1.5	References	1
2	Overall Description	2
2.1	Product Perspective	2
2.2	Product Functions	2
2.2.1	Account Management	2
2.2.2	Workout Summary and Recommendation	2
2.2.3	Workout Creation and Management	3
2.2.4	Gym Buddy Interactivity	3
2.2.5	Diagram and Summary of System Features	3
2.3	User Classes and Characteristics	4
2.3.1	Gender	4
2.3.2	Fitness Expertise	4
2.3.3	Fitness Goals	5
2.3.4	Desired Level of Socialisation	5
2.3.5	Desired Workout Configuration	5
2.4	Operating Environment	6
2.4.1	Angular Framework	6
2.4.2	Ionic Framework	6
2.4.3	Firebase	6
2.5	Design and Implementation Constraints	6
2.6	User Documentation	7
2.6.1	Source Code ESLint Compliance	8
2.7	Assumptions and Dependencies	8
3	External Interface Requirements	8
3.1	User Interfaces	8
3.1.1	Strive for consistency	9
3.1.2	Seek universal usability.	9
3.1.3	Offer informative feedback.	10
3.1.4	Design dialogue to yield closure.	10
3.1.5	Prevent errors.	10

3.1.6 Permit easy reversal of actions.	11
3.1.7 Support internal locus of control.	11
3.1.8 Reduce short-term memory load.	12
3.2 Hardware Interfaces	12
3.3 Software Interfaces	13
3.4 Communications Interfaces	14
4 System Features	15
4.1 Create main application account	15
4.1.1 Description and Priority	15
4.1.2 Stimulus/Response Sequences	15
4.1.3 Functional Requirements	15
4.2 Enter User details	16
4.2.1 Description and Priority	16
4.2.2 Stimulus/Response Sequences	16
4.2.3 Functional Requirements	16
4.3 Login to main app account	16
4.3.1 Description and Priority	17
4.3.2 Stimulus/Response Sequences	18
4.3.3 Functional Requirements	19
4.4 Update Personal Information	19
4.4.1 Description and Priority	19
4.4.2 Stimulus/Response Sequences	19
4.4.3 Functional Requirements	19
4.5 Recommend Personalised Workout	20
4.5.1 Description and Priority	20
4.5.2 Stimulus/Response Sequences	20
4.5.3 Functional Requirements	21
4.6 Conduct Workout	23
4.6.1 Description and Priority	23
4.6.2 Stimulus/Response Sequences	23
4.6.3 Functional Requirements	24
4.7 Edit Workout Routine	25
4.7.1 Description and Priority	25
4.7.2 Stimulus/Response Sequences	25
4.7.3 Functional Requirements	25
4.8 Display Workout Summary	26
4.8.1 Description and Priority	26
4.8.2 Stimulus/Response Sequences	26

4.8.3 Functional Requirements	26
4.9 Display Weekly Statistics	27
4.9.1 Description and Priority	27
4.9.2 Stimulus/Response Sequences	27
4.9.3 Functional Requirements	28
4.10 Display created workouts	29
4.10.1 Description and Priority	29
4.10.2 Stimulus/Response Sequences	29
4.10.3 Functional Requirements	30
4.11 View completed workouts	31
4.11.1 Description and Priority	31
4.11.2 Stimulus/Response Sequences	31
4.11.3 Functional Requirements	31
4.12 Display Recommended Videos	32
4.12.1 Description and Priority	32
4.12.2 Stimulus/Response Sequences	32
4.12.3 Functional Requirements	33
4.13 Create Gym Buddy Account	33
4.13.1 Description and Priority	33
4.13.2 Stimulus/Response Sequences	33
4.13.3 Functional Requirements	34
4.14 Display Gym Buddy Home Page	35
4.14.1 Description and Priority	35
4.14.2 Stimulus/Response Sequences	35
4.14.3 Functional Requirements	36
4.15 Update Gym Buddy Account Preferences	37
4.15.1 Description and Priority	37
4.15.2 Stimulus/Response Sequences	37
4.15.3 Functional Requirements	37
4.16 View Suggested Buddy Details	38
4.16.1 Description and Priority	38
4.16.2 Stimulus/Response Sequences	38
4.16.3 Functional Requirements	39
4.17 Request to pair up with buddy	41
4.17.1 Description and Priority	41
4.17.2 Stimulus/Response Sequences	41
4.17.3 Functional Requirements	41
4.18 View all chats	41

4.18.1 Description and Priority	41
4.18.2 Stimulus/Response Sequences	41
4.18.3 Functional Requirements	42
4.19 View Chat With Buddy	42
4.19.1 Description and Priority	42
4.19.2 Stimulus/Response Sequences	42
4.19.3 Functional Requirements	43
4.20 Send Message	43
4.20.1 Description and Priority	43
4.20.2 Stimulus/Response Sequences	43
4.20.3 Functional Requirements	44
4.21 Remove Buddy	44
4.21.1 Description and Priority	44
4.21.2 Stimulus/Response Sequences	44
4.21.3 Functional Requirements	44
4.22 Share Workout	45
4.22.1 Description and Priority	45
4.22.2 Stimulus/Response Sequences	45
4.22.3 Functional Requirements	45
5 Other Nonfunctional Requirements	47
5.1 Performance Requirements	47
5.2 Safety Requirements	47
5.3 Security Requirements	47
5.4 Software Quality Attributes	48
5.4.a Reliability:	48
5.4.b Maintainability:	48
5.4.c Usability:	48
5.4.d Supportability:	48
5.4.e Business Rules	48
6 Other Requirements	49
6.e Infrastructure:	49

Revision History

Name	Date	Reason For Changes	Version
WomenInTech	12/4/2022	Initialise document	0.1

WomenInTech	16/4/2022	Completed document for release	1.0
-------------	-----------	--------------------------------	-----

1. Introduction

1.1 Purpose

This document presents a detailed description of the first release of the WITFIT application. This document describes an overall description of the application which includes its operating environment along with certain design and implementation constraints. It also defines the different external interface requirements and communications with external APIs. Last but not least, it describes the different system features present as well as other non-functional requirements.

1.2 Document Conventions

Each requirement is a collection of items that were gathered from our weekly roleplay sessions as well as our initial research about the gaps in the existing market. Each requirement has been agreed upon by the entire group. Each requirement is a clear one or two sentence description of the intention of the requirement. Priorities for higher-level requirements are inherited by detailed requirements. The use of *italics* represent captions. The use of underlines represents a reference to a feature in the product.

1.3 Intended Audience and Reading Suggestions

This document aims to provide an excellent mix between technical information as well as the business context behind our application. This allows developers, project managers, users and testers to understand our application at a deeper level regardless of their technical experience. This document will first provide the rationale for why the application was created and then bring you through the technical intricacies that make up the application. By the end of the document, the user will have a clear understanding of the features present and how it brings value to the end user. This document can be read in sequential order.

1.4 Product Scope

“Witfit” is an cross-platform application that is designed to fill in the niche that is currently not being addressed by conventional workout application. It is designed from the ground up to provide an inclusive experience regardless of the end-user’s experience with working out. We aim to build a platform and a community where anyone is able to simply hop on the application and be provided with the tools necessary to start working out.

1.5 References

The design of our code base closely followed 10 key basic programming principles as shown here: <https://www.makeuseof.com/tag/basic-programming-principles/>

The user interface design followed several best practices that is outlined in this document: <https://xd.adobe.com/ideas/process/ui-design/4-golden-rules-ui-design/>

The UI components of this application was created by closely following the Ionic documentation: <https://ionicframework.com/docs>

In order to make the most use of the Ionic framework, we also referenced some best practices that was released by Ionic: <https://ionic.io/enterprise-guide/angular>

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained product addressing the burgeoning need to keep Singapore healthy. The Singapore government has adopted a three-pronged approach to address this need¹, but the last factor hinges upon the willpower of the people and their conscious, deliberate effort to exercise regularly. With only 26% of Singaporeans being reported to be physically active 3 days per week², more needs to be done to inculcate a lifestyle of fitness.

In line with the nation's drive towards becoming a technologically-savvy Smart Nation, our team proposes WITFIT, a cross-platform mobile application designed to build an active and healthy community through nurturing a health-conscious lifestyle.

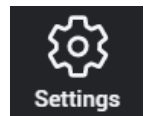
We note the efforts of previous teams and the fitness applications they have developed. However, few of them actually cater to the needs of both men and women, and there is currently no application that caters to the needs of women in their fitness journey. Concerns surrounding natural female physiological processes (such as menstruation) and female-specific recommendations and exercise advice are highly nuanced subjects, and we hope that our team will manage to address this in our application.

2.2 Product Functions

We have 22 functions in total. These can generally be broken down into four (4) interacting and complementary facets:

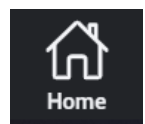
2.2.1 Account Management

This concerns functions surrounding the user, such as account registration and management, and login.



2.2.2 Workout Summary and Recommendation

This powers the Homepage of the application, home to a dashboard informing users of



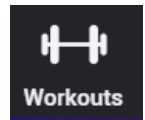
¹ www.straitstimes.com/singapore/health/incentives-work-to-keep-people-healthy-and-are-cheaper-than-treatment

² www.activehealth.sg/move-better

their weekly and overall videos, and personalised video recommendations to guide users at every step of their fitness journey.

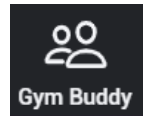
2.2.3 Workout Creation and Management

The core functionality of our app, this allows users to automatically generate a personalised workout routine and conduct it for themselves. They also have the ability to save and edit their previously generated workouts.



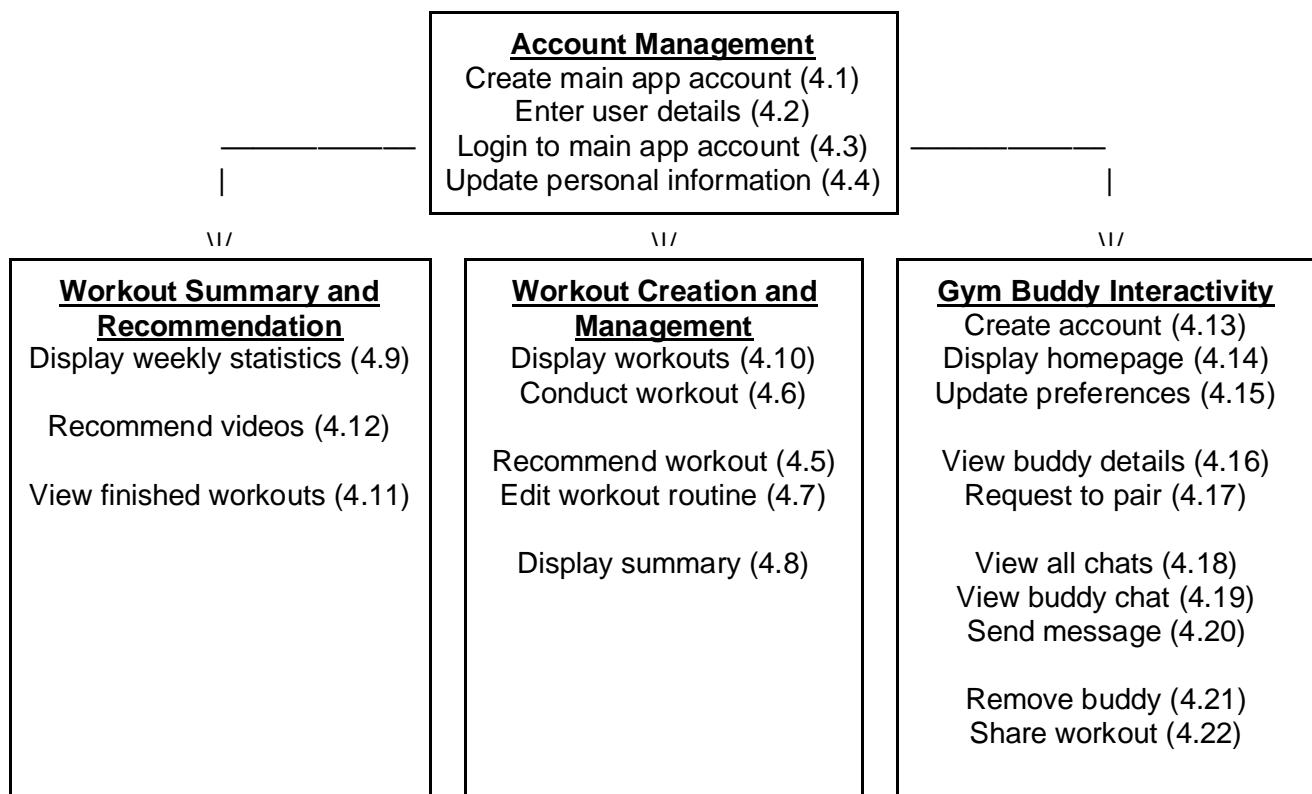
2.2.4 Gym Buddy Interactivity

This is an entire subsection within our application that aims to underscore the social, communal aspect of fitness, offering users a way to connect, interact, and form meaningful bonds with like-minded people based on their match preferences.



2.2.5 Diagram and Summary of System Features

Details of each of the 22 functions can be seen in Section 4. A diagram mapping each function to one of the 4 facets can also be found below:



2.3 User Classes and Characteristics

In general, our application is targeted towards users looking for an application to accompany them in their fitness journey. We can further segment this target audience based on the following five (5) user classes:

<u>Gender</u>	<u>Fitness Expertise</u>	<u>Fitness Goals</u>	<u>Desired Level of Socialisation</u>	<u>Desired Workout Configuration</u>
Male Female	Beginner Advanced	Increase muscularity Build strength Body conditioning Raise athleticism Lose weight	Introverted Extroverted	<u>Intensity</u> - Low, Medium, High <u>Duration</u> - Low, Medium, Long <u>Location</u> - Gym, Home, Outdoors <u>Equipment</u> - None, Minimal

In each subsequent subsection, we will explore the various subclasses in each user class, analysing their pertinent characteristics and corresponding requirements. This influences our decision on app functionality.

This is then followed by a business decision distinguishing the more important user classes from the rest, where we will also provide our rationale and justification if any.

2.3.1 Gender

Male	General fitness advice on the Internet is unfairly biased towards the male demographic.
Female	Fitness advice catered for the female physiology are comparatively rarer. Most fitness apps also do not account for menstruation cycles.

For females, it is hence important for us to account for their gender in the Account Management facet to allow them to store information about their menstrual cycle in 4.2 Enter user details, as well as in the Workout Summary and Recommendation facet to 4.12 Recommend personalised videos specific for females.

As expressed in our Product Perspective in Section 2.1, a fitness app catered to females has a more pressing business need and aligns better with our team WomenInTech's goals. Thus, we will distinguish our female user class as more important to satisfy than that of the male user class.

2.3.2 Fitness Expertise

Beginner	Requires more advice on general fitness, form checks, etc
Advanced	Requires less advice on general fitness, but might still need form checks sometimes

Beginners may require more guidance when starting out, as they may be inexperienced with the general knowledge around exercising. Thus, we account for this in the Workout Summary and Recommendation facet, where we 4.12 Recommend personalised videos suitable for beginners and specific to their general fitness goal. When they become an Advanced user, we will then 4.12 Recommend personalised videos based on their most recent workout, ensuring they will still receive some guidance on their form from time to time.

As expressed in our Product Perspective in Section 2.1, our fitness app aims to inculcate a healthier lifestyle in its users. Thus, we will distinguish our beginner user class as more important to satisfy.

2.3.3 Fitness Goals

To optimise for a specific fitness goal, different approaches to training, dieting, and workout regimentation are required. Thus, it is important to make this distinction for users, especially for the requirements under the facet of Gym Buddy Interactivity and users' ability to 4.15 Update gym buddy preferences.

The purpose of our application is to promote a healthy lifestyle, irrespective of specific fitness goal, hence, we have decided to place equal importance on each user subclass in Fitness Goal. Distinguishing based on Fitness Goal will also skew our userbase towards a certain Fitness Goals, limiting diversity and possibly making it harder for users of our Gym Buddy Interactivity facet to find a match if their preferences belong to a user subclass we have neglected.

2.3.4 Desired Level of Socialisation

Introverted	Prefers a low level of socialisation, likely to exercise alone
Extroverted	Prefers a high level of socialisation, likely to exercise with others

Users' preference for introversion/extraversion will also influence their workout preferences, whether it be alone or with others. This distinction is especially important for the Gym Buddy Interactivity facet, where introverted users are less likely to make use of this facet, or are less likely to make the first move 4.16 Request to pair compared to extroverts.

Be that as it may, our userbase requires a healthy mix of both introverts and extroverts for a positive user experience in the Gym Buddy Interactivity facet. As such, we will place equal importance on both subclasses. To cater to the introverted user class, we will make the Gym Buddy Interactivity facet operate on an opt-in basis via 4.13 Create gym buddy account.

2.3.5 Desired Workout Configuration

There are many permutations of Intensity, Duration, Location, and Equipment one can choose in their preferred workout configuration as listed above. Thus, it is important to make this distinction for users, especially for the requirements under the facet of Workout Creation and Management and our app's ability to 4.5 Recommend personalised workout.

Similar to Section 2.3.3, the purpose of our application is to promote a healthy lifestyle agnostic of how it is achieved. Hence equal importance will be placed on each user subclass in Desired Workout Configuration.

2.4 Operating Environment

Our technology stack will utilise the following tools:

2.4.1 Angular Framework



A free and open-source web application framework. Its key feature is the ability to build robust, cross-platform web applications for both mobile and desktop. This allows us to design our application for the following operating environments in mind:

Mobile	Desktop
iOS: 2 most recent major versions Android: 2 most recent major versions	Chrome: latest version Firefox: latest and extended support release Edge: 2 most recent major versions

2.4.2 Ionic Framework



More specifically, we utilise the Ionic framework, which is a free and open-source Software Development Kit (SDK) that encompasses Angular (see 2.4.1) and packages it with many front-end tools to allow us to develop beautiful applications. Similar to Angular, it has the ability to develop cross-platform applications.



On top of its own framework features, Angular Ionic also makes use of the following languages: HTML5, Sassy CSS (SCSS), and TypeScript (TS). We will use these in our application.

2.4.3 Firebase



Lastly, we will make use of Firebase, a Backend-as-a-Service for our data storage and organisation. It stores and retrieves information in JSON format, and integrates well with the Angular framework.

2.5 Design and Implementation Constraints

As mentioned in our Product Perspective in Section 2.1, this product is a new, self-contained product developed from scratch. Thus, no real design and implementation constraints exist on the application yet. The frameworks chosen in Section 2.4 Operating Environment are suggestions, and represent a single possibility on which this codebase can be developed.

However, once we start development on this framework (as we have) and continue working in this environment, it will become increasingly expensive to migrate frameworks. When this cost of refactoring becomes prohibitively expensive, the above-mentioned framework will become a Design and Implementation Constraint and should be migrated to this Section instead.

2.6 User Documentation

The following documents are provided along with this SRS:

Deliverables	Delivery Format	Standards Compliances
Functional Requirements Elicitation	.docx	Page 126-130, Fox ³
Non-Functional Requirements Elicitation	.docx	
Use Case Diagram	.pdf	Developed in Visual Paradigm
Use Case Description	.docx	-
UI Mockup	.pdf	Done in Figma
Class Diagram	.pdf	Page 341-345, Fox ³
Sequence Diagram	.pdf	Developed in Visual Paradigm
Dialog Map	.pdf	Page 160, Fox ³
System Architecture	.pdf	Developed in Visual Paradigm
Source Code	SVN Repository	JSDocs-complete, ESLint-compliant
Test Cases	.docx	White-box and black-box testing
Product Demonstration Video	.mp4	-

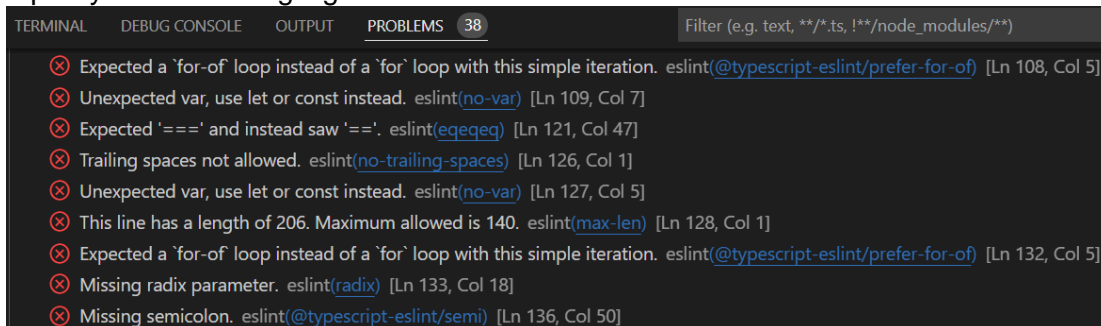
³ Introduction to Software Engineering Design: Processes, Principles and Patterns with UML2 – C. Fox

2.6.1 Source Code ESLint Compliance



Our source code is compliant with version 8.13.0 of ESLint, a static code analysis tool for enforcing high code quality and proper coding styles within the JavaScript language. With many developers on the WomenInTech team, having a consistent ruleset to follow was important in ensuring our codebase remains consistent and easily readable by everyone in the team.

Borrowing from agile development principles, we applied the refactoring strategy in our software development by iteratively improving upon code syntax with guidance from ESLint. Some examples of code quality issues are highlighted below.



ESLint can only serve as a guide, as certain coding styles were intentionally designed to be that way despite being flagged out by ESLint. Despite this, we were able to spot and correct over two hundred correctly identified code quality issues using ESLint, greatly improving the general maintainability and readability of our codebase.

2.7 Assumptions and Dependencies

The following assumptions are crucial in the operational capabilities of our application:

- The user has a stable and consistent internet connection throughout their usage of the app.
- The Firestore database will continue to have no rate limits or total call limits.
- The Workout API we use to retrieve exercises will continue to remain free and operational with no rate limits or total call limits.

3. External Interface Requirements

3.1 User Interfaces

Ben Shneiderman is an expert in the field of human-machine interaction, and has formulated eight golden rules with regards to designing “user-friendly” UI/UX. We applied each of these 8 golden rules⁴⁵ in the design of our app, and we discuss each rule in the following individual subsections.

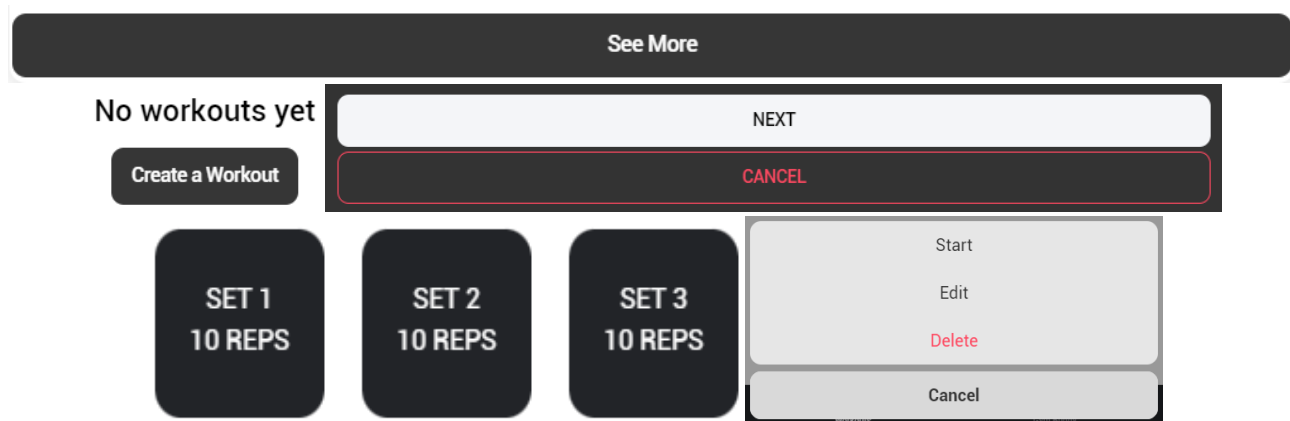
⁴ www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces

⁵ <https://www.cs.umd.edu/~ben/goldenrules.html>

3.1.1 Strive for consistency

Standardising the way information is conveyed ensures users are able to apply knowledge from one click to another without the need to learn new representations for the same actions. Consistent colour, layout, capitalisation, fonts, etc should be employed throughout the application.

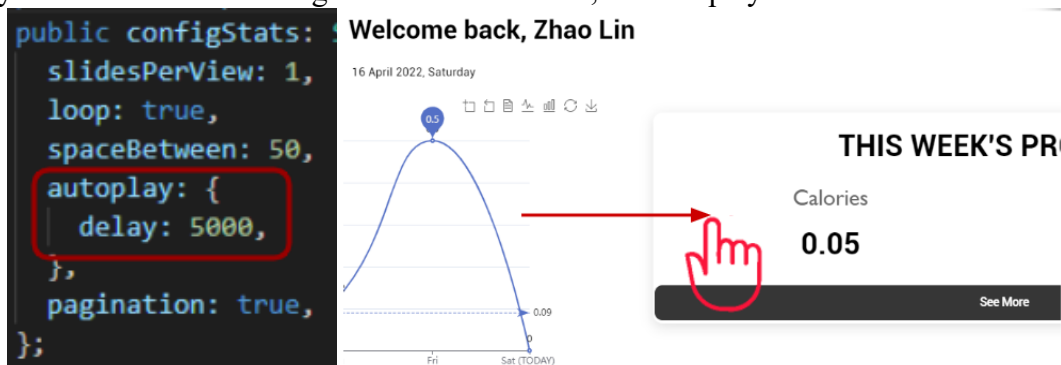
We applied this throughout the entire design of our application, adhering closely to the Ionic framework's guidelines when designing our front-end. Some examples of our UI buttons are shown below.



3.1.2 Seek universal usability.

The needs of diverse users have to be recognised, and hence designed around, facilitating transformation of content. Users of different levels of technical capability, age range, disability etc require different kinds of features that cater to their skill level.

We applied this principle in the Workout Summary and Recommendation facet in our 4.9 Display(ing of) weekly statistics. Using an intuitive slider interface, we first set the sliders to slide automatically every 5 seconds as seen below on the left. This will be the main way beginner users will access our information. This autoplay also acts as an invisible 'tutorial' for users to understand how to navigate between the two panes, and more advanced users will hence be able to swipe between the two slides manually as seen above on the right. When this occurs, the autoplay will be disabled.



3.1.3 Offer informative feedback.

Every user action should have an interface feedback, allowing them to know where they are at and what is going on at all times.

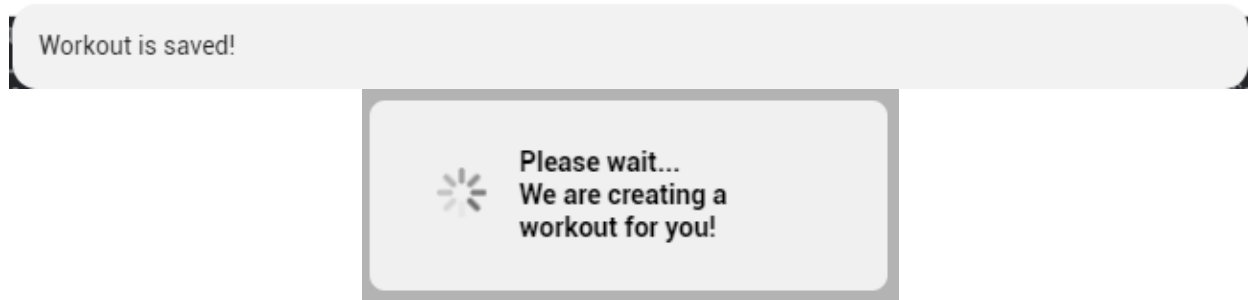
We applied this principle by employing an app-wide toolbar at the bottom, to clearly indicate which facet of the app the user is currently in.



3.1.4 Design dialogue to yield closure.

Sequences of actions should result in feedback displayed on the app

We applied this principle using frequent toast messages that appear when a user confirms an action, such as in [4.5 Recommend workout](#) under the [Workout Creation and Management](#) facet:



3.1.5 Prevent errors.

Interfaces should be designed such that users cannot make serious errors. If users do make an error, interfaces should offer simple and intuitive step-by-step instructions to solve the problem as quickly and painlessly as possible.

To prevent users from making errors, we ensure that options that cannot be chosen are greyed out, so that users do not even have the option of selecting it in the first place. For example, in [4.13 Create gym buddy account](#) of the [Gym Buddy Interactivity](#) facet, users are allowed to select up to 2 locations they would like to train. Once 2 locations are selected, we grey out the other options, as seen in the figure below. Additionally, if the user has not filled out all the required forms, the buttons to navigate forward are also greyed out, as seen in the 'NEXT' button in the above figure.

Where would you like to train?
Select up to 2 locations

☐ Outdoor

☐ Gym

☒ Studio

☐ Park

☐ Home

☐ Stadium

NEXT

Where would you like to train?
Select up to 2 locations

☐ Outdoor

☐ Gym

☒ Studio

☒ Park

☐ Home

☐ Stadium

NEXT

As for error resolution, we strive for rapid error correction by the user by flagging our requirements in red, as seen in [4.1 Create main app account](#) of the [Account Management](#) facet. Here, our requirements for the Username and Password fields are stated clearly in red to contrast with the black text, letting users know exactly what is required for these fields. We similarly grey out the forward navigation button (SIGN UP).

Username*

name

Username needs to be 5 characters

Password*

Password

Password needs to be 6 characters

Confirm Password*

.

Password does not match

SIGN UP

3.1.6 Permit easy reversal of actions.

Users should have obvious ways to reverse their actions at any point of the app to relieve anxiety.

We applied this principle by ensuring each page has a 'Back' button that users can use to navigate backward to the page they came from. One example exists in [4.11 View finished workouts](#), where clicking the Back button brings them back to [4.9 Display weekly statistics](#) all within the [Workout Summary and Recommendation](#) facet.

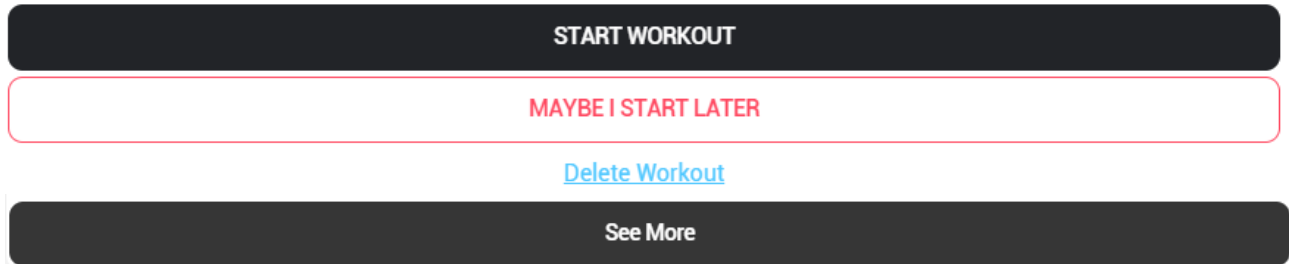


Your Profile

3.1.7 Support internal locus of control.

Allow your users to be the initiators of actions, giving them the sense that they are in full control of events occurring in the digital space.

We apply this by offering users the choice to initiate many actions, such as on when to initiate the workout on 4.6 Conduct workout in the Workout Creation and Management facet, and when they wish to 4.11 View finished workouts in the Workout Summary and Recommendation facet.



START WORKOUT

MAYBE I START LATER

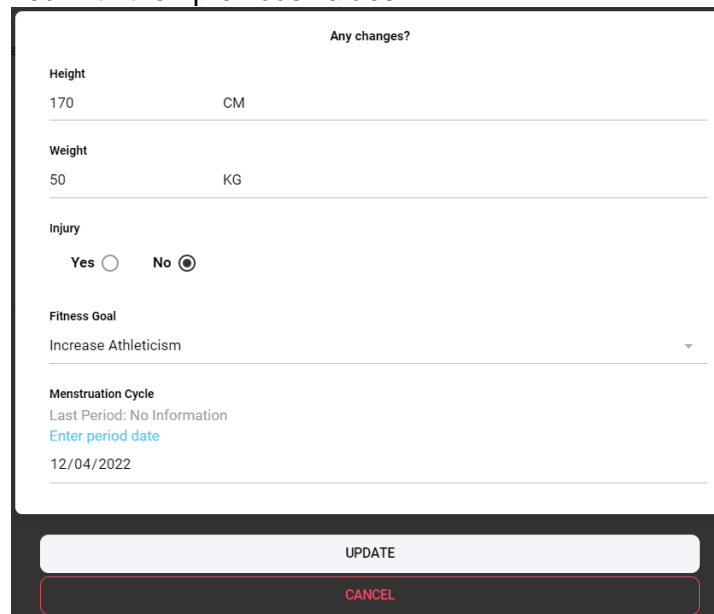
[Delete Workout](#)

See More

3.1.8 Reduce short-term memory load.

Human attention is limited and we are only capable of maintaining around five items in our short-term memory at one time. Therefore, interfaces should be as simple as possible with proper information hierarchy, and choosing recognition over recall.

We applied this principle in 4.4 Updating personal information under the Account Management facet, where forms are pre-filled with their previous values.



Any changes?

Height
170 CM

Weight
50 KG

Injury
Yes ☐ No ☒

Fitness Goal
Increase Athleticism

Menstruation Cycle
Last Period: No Information
[Enter period date](#)
12/04/2022

UPDATE

CANCEL

3.2 Hardware Interfaces

As this application is cross-platform, we are able to support input from both PCI-compliant peripheral devices such as mouse, keyboard, etc in the case of web applications, as well as from capacitive

touch screens in the case of mobile applications. Emulators also allow us to simulate a capacitive touch screen using a peripheral, and vice versa.

3.3 Software Interfaces

The following API endpoints are used by the application:

Firestore Database API v7.2.1	To store and fetch all data related to users, workouts, completed workouts, and <u>Gym Buddy Interactivity</u> -related artifacts.
Youtube Data API v3	To fetch recommended Youtube videos for our user in <u>4.12 Recommend videos</u> under the <u>Workout Summary and Recommendation</u> facet, and in <u>4.6 Conduct workout</u> under the <u>Workout Creation and Management</u> facet.
Exercise API	To fetch recommended exercises for our user's created workout, for <u>4.5 Recommend workout</u> under the <u>Workout Creation and Management</u> facet.

Additionally, these are all the identified dependencies for our application based on package.json:

Angular	
@angular/common": "~13.0.0", @angular/core": "~13.0.0", @angular/cli": "13.0.4", @angular/compiler": "~13.0.0", @angular-devkit": "~13.0.1", @angular/language-service": "~13.0.0", @ionic/angular-toolkit": "^4.0.0", @ionic-native/core": "^5.36.0", @ionic/angular": "^6.0.10", rxjs": "~6.6.0", tslib": "^2.2.0", zone.js": "~0.11.4", @types/jquery": "^3.5.14", @types/node": "^12.11.1", ts-node": "~8.3.0", typescript": "~4.4.4"	Core application functionality powered by core Angular, Ionic, and TypeScript libraries
firebase": "^9.6.7", @angular/fire": "^7.2.1", rxfire": "^6.0.3",	Allow the application to connect to Firestore using core Angular functionality
@angular/platform-browser": "~13.0.0", @angular/platform-browser-dynamic": "~13.0.0",	Allow the application to run on the web using core Angular functionality
cordova-android": "10.1.1", cordova-ios": "^5.1.0",	Allow the application to run on mobile (Android and iOS) using Cordova and Capacitor plugin

cordova-plugin-device": "2.0.2", cordova-plugin-ionic-keyboard": "^2.0.5", cordova-plugin-ionic-webview": "^5.0.0", cordova-plugin-splashscreen": "5.0.2", cordova-plugin-statusbar": "^2.4.2", @capacitor/app": "1.1.0", @capacitor/core": "^3.4.3", @capacitor/cli": "^3.4.3", @capacitor/filesystem": "1.1.0", @capacitor/keyboard": "1.2.2", @capacitor/status-bar": "1.0.8", @capacitor/storage": "1.2.4",	
@angular/forms": "~13.0.0", @angular/router": "~13.0.0", echarts": "^5.3.1", swiper": "^8.0.7",	Allow the application to have various functionality such as user forms, routing, charts, and swiper slides using various Angular and Ionic libraries
eslint": "^8.13.0", eslint-plugin-import": "2.22.1", eslint-plugin-jsdoc": "30.7.6", eslint-plugin-prefer-arrow": "1.2.2", @angular-eslint/template-parser": "~13.0.1", @typescript-eslint/parser": "5.3.0",	Allow the application to work with ESLint (see Section 2.6.1)
@angular/youtube-player": "13.3.1", @ionic-native/youtube-video-player": "5.36.0",	Allow Angular to play Youtube videos

3.4 Communications Interfaces

We use the HTTP web protocol to communicate with APIs. Since we are receiving Youtube video URLs from the Youtube API, extra precaution is required in sanitising the URLs we receive from the Youtube API endpoint. This is to prevent Cross-site Scripting (XSS) attacks, and is recommended as part of the OWASP Top 10 Secure Coding Principles.⁶

```
@Pipe({
  name: 'safe'
})
/**
 * SafePipe class to sanitise URLs to prevent cross-site scripting attacks. Used to sanitise the youtube API urls.
 */
export class SafePipe implements PipeTransform {
  constructor(private sanitizer: DomSanitizer) { }
  transform(url) {
    return this.sanitizer.bypassSecurityTrustResourceUrl(this.sanitizer.sanitize(SecurityContext.URL, url));
  }
}
```

⁶ <https://owasp.org/www-project-top-ten/>

4. System Features

4.1 Create main application account

4.1.1 Description and Priority

First time users are to create a WITFIT account to use the application. This feature is of high priority as users without an account will not be able to login to access any of the application functionalities.

4.1.2 Stimulus/Response Sequences

1. Unregistered user selects “I don’t have an account”
2. App brings user to sign up page
3. User fills up sign up form (First name, last name, email, birthday, gender, username, password, confirm password) and clicks sign up button.
4. User is prompted to enter user details ([4.2 Enter User Details](#))
5. If user’s credential exists, app returns to log in page

4.1.3 Functional Requirements

1. The app must allow the new users to sign up for an account
 - a. If user exists, the app must prompt users to log in.

WITFIT

SIGN UP

First Name*

First Name

Last Name*

Last Name

Email*

Email

Birthday*

dd/mm/yyyy

Gender*

Select One

Username*

@

Password*

Password

Confirm Password*

Confirm Password

SIGN UP

[I have an account](#)

4.1 Sign Up page that will be navigated to if user selects “I don’t have an account”

4.2 Enter User details

4.2.1 Description and Priority

First time users are to enter their credentials as part of account creation. This feature is of high priority as users will not be able to login to access any of the application functionalities without an account

4.2.2 Stimulus/Response Sequences

User presses continue account creation (Continuous from 4.1 Create main application account)

1. App will prompt user for their height and weight
2. App validates height and weight and prompt user for any physical injuries or pre-existing health conditions
 - a. If height or weight entered is invalid, app displays error message "Please enter a valid height and weight"
 - b. App returns to account creation
3. If user answers "No" to both questions, app will prompt user to enter their fitness goal
 - a. If user answers "Yes" to physical injuries, app will prompt user for their physical injuries
 - b. If user answers "Yes" to pre-existing medical conditions, app will prompt user for their medical condition
 - c. If user answers "Yes" to both questions, step *a* and *b* takes place
4. App sends sign up details to database
 - a. If user is female, app will prompt user for menstruation details
 - b. App will prompt user to enter date of their last period

User selects "Not sure" / "I am pregnant" / Enter date of last period

5. Welcome message is displayed

4.2.3 Functional Requirements

1. The app must prompt first time users for their login details.
 - a. The app must prompt users for their username.
 - b. The app must prompt users for their password.
 - c. The app must prompt users to confirm their password.
 - d. The app must prompt first time users for their personal details.
 - e. The app must prompt users for their first name.
 - f. The app must prompt users for their last name.
 - g. The app must prompt users for their email.
 - h. The app must prompt users for their birthday.
 - i. The app must prompt users for their gender.
 - j. The app must prompt users for their height and weight.
 - k. The app must prompt users about the presence of any physical injuries.
 - l. The app must prompt users about the type of physical injuries.
 - m. The app must prompt users about the area of injuries.
 - n. The app must prompt users about the presence of any pre-existing health conditions.
 - o. The app must prompt users about their fitness goals.

- p. If the user is female, the app must prompt users about the date of their last period.
 - i. The app shall provide the option to skip this step if they are unsure of the date of their last period.
 - ii. The app shall provide the option to skip this step if the user is pregnant.

We would like to know more about you

A screenshot of a mobile application interface. At the top, the text "We would like to know more about you" is displayed above a horizontal line. Below this, a white rounded rectangle contains the following elements: the text "What is your height?" followed by a text input field and the label "CM"; the text "What is your weight?" followed by a text input field and the label "KG"; and a wide, dark grey button with the text "Next" in white.

4.2 App prompting for user for their height and weight

We would like to know more about you

A screenshot of a mobile application interface. At the top, the text "We would like to know more about you" is displayed above a horizontal line. Below this, a white rounded rectangle contains the following elements: the text "Any physical injuries?" followed by two radio buttons labeled "Yes" and "No"; the text "Do you have pre-existing health conditions?" followed by two radio buttons labeled "Yes" and "No"; a wide, dark grey button with the text "Next" in white; and a text link "Go back" in blue.

4.2 App prompting for user about the presence of injuries and pre-existing health conditions

What is your fitness goal?

GAIN MUSCLES

LOSE WEIGHT

BUILD STRENGTH

BODY CONDITIONING

INCREASE ATHLETICISM

Finish

[Go back](#)

4.2 App prompting for users for their fitness goals

We would like to know more about you

When was your last period?

mm / dd / yyyy

Not Sure

I'm pregnant

Finish

[Go back](#)

App prompting for female user for the date of their last period

4.3 Login to main app account

4.3.1 Description and Priority

User logs into app using their WITFIT account. This feature is of high priority as features of the app such as (Update personal information, conduct workout, create customised workout routine, display workout summary and create buddy account) will not be available without logging in.

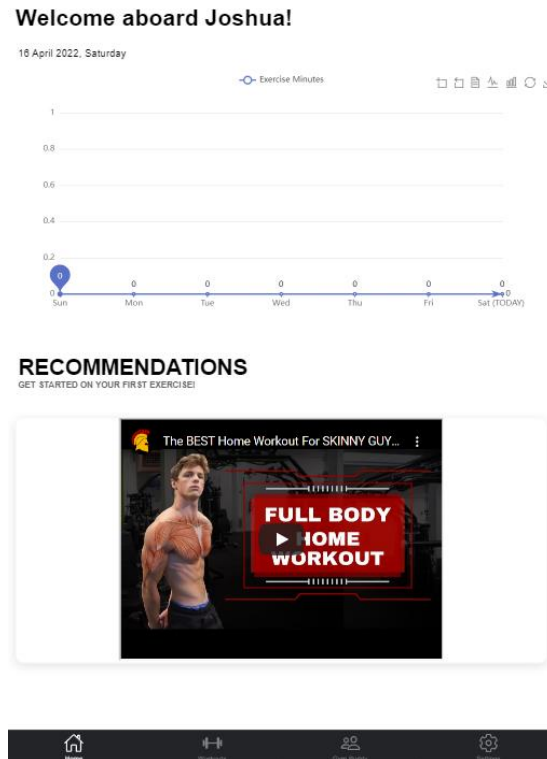
4.3.2 Stimulus/Response Sequences

1. User opens app and log in page is displayed

2. User enters username and password and presses log in button
3. System validates username and password using database and logs user in
 - a. If invalid, app displayed warning message and returns to log in page
4. Home page is displayed and welcome message is shown

4.3.3 Functional Requirements

1. If users already have an account, users must be able to log in. After logging in, users must be able to sign out whenever they want.



4.3 App Home page

4.4 Update Personal Information

4.4.1 Description and Priority

User logs into app using and update personal information. This feature is of lower priority as this function will only be invoked if ther user wants to edit their personal information

4.4.2 Stimulus/Response Sequences

1. User selects edit profile on workout interface
2. Pre-filled user information is displayed and app will prompt user to edit personal information
3. Users are allowed to update their height, weight, fitness goals and injuries
4. Female users are able to update their last period date

5. App validates entry and update database
6. If valid, user will be navigated to workout creator interface
 - a. If not all fields are filled, app displays warning message and returns to prompt user to edit their personal information

4.4.3 Functional Requirements

1. The app must be able to allow users to update their details prior to each workout
 - a. The app must showcase previously inputted details.
 - b. The app must allow users to update their height and weight.
 - c. The app must allow users about the presence of any physical injuries
 - i. The app must prompt users about the type of physical injuries.
 - ii. The app must prompt users about the area of injuries.
 - d. The app must allow users to update their fitness goals.
 - e. If the user is female, the app must allow users to update the date of their last period.
 - i. The app shall provide the option to skip this step if they are unsure of the date of their last period.
 - ii. The app shall provide the option to skip this step if the user is pregnant.

4.3 Update personal information page

4.5 Recommend Personalised Workout

4.5.1 Description and Priority

User selects a recommended personalised workout that will be generated by the app. This feature is high priority as user selects workout based on their personal information input hence it is important for the app to recommend a workout that is suitable for the user.

4.5.2 Stimulus/Response Sequences

1. On successful login, user selects add a workout
2. App prompts user to update user's details (Continuous from [4.4 Update Personal Information](#))
3. User updates personal details if there are any updates.
4. App prompts user to input workout details.

5. After processing updated user and workout details, app will create a workout by using the workout algorithm (processes injuries, goals and list of exercises of API).
6. App displayed created workout

4.5.3 Functional Requirements

1. Users must be able to create their own customised workout routine.
 - a. Users must be able to enter a workout routine title.
 - b. Users shall enter a description of their workout routine.
 - c. Users must be allowed to select the intensity of their workout based on the metabolic equivalents.
 - d. Users must be allowed to select the duration of the workout.
 - e. Users must be allowed to select the location to do their workout.
 - f. Users must be allowed to select the type of equipment they have.
2. The app must display the workout information before they start.
 - a. The app must display the title of their workout routine.
 - b. The app must display the workout description.
 - c. The app must display the duration of the workout.
 - d. The app must display the location that the workout will take place.
 - i. If the user selects to workout at the gym, the user must be able to find the closest gym to them.
 - e. The app must display the full workout routine and the time taken for warm-up, exercise and cool-down.
 - f. The app shall display the sets and reps suggested for each exercise.
 - g. The app shall display the option to start the workout now, later or delete the workout recommended.

< CREATE WORKOUT

Workout Name

Description

Intensity

☐ LOW < 3.0 METs

☐ MODERATE 3.0 - 6.0 METs

☐ VIGOROUS 6.0 - 9.0 METs

☐ HARD > 9.0 METs

Duration

☐ SHORT AND QUICK 15 mins

☐ BACK ON TRACK 30 mins

☐ BUILT DIFFERENT 60 mins

☐ BEAST MODE 90 mins +

Location

☐ GYM

☐ HOME

☐ OUTDOOR

Equipment

☐ BASIC EQUIPMENT

☐ MINIMAL EQUIPMENT

☐ NO EQUIPMENT

CREATE WORKOUT

Home Workout Profile Settings

4.5 Page displaying workout details that must be inputted by the user

The screenshot displays a workout routine interface. At the top, it shows 'WARM UP' with a duration of '5 mins' and 'EXERCISES' with a duration of '20 mins'. Below this is a list of exercises with their respective sets and reps:

THRUSTER	3 sets	10 reps
LOW BOX SQUAT - WIDE STANCE	3 sets	10 reps
DECLINE BENCH PRESS DUMBBELL	3 sets	10 reps
DUMBBELL LUNGES WALKING	3 sets	10 reps
TURKISH GET-UP	3 sets	10 reps

Below the exercise list, it shows 'COOL DOWN' with a duration of '5 mins'. At the bottom, there are two buttons: 'START WORKOUT' (black) and 'MAYBE I START LATER' (red outline). Below the 'MAYBE I START LATER' button is a link 'Delete Workout' in blue text.

4.5 Workout generated

4.6 Conduct Workout

4.6.1 Description and Priority

Facilitates the overall flow of a workout routine. This feature is high priority as user will almost always use this functionality when wanting to exercise, which ranges from 3-7 times per week per user.

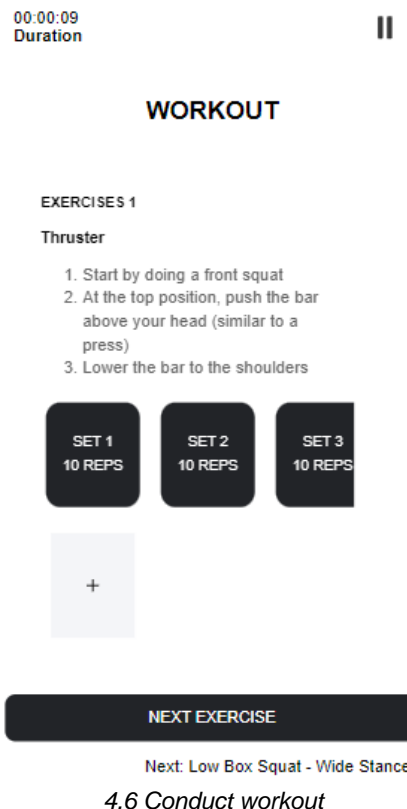
4.6.2 Stimulus/Response Sequences

1. On successful login, user selects the desired workout they wish to start
2. App displays a timer and a video from youtube for warm up
3. Once warm up is completed, user can proceed to do exercises
4. User can choose to modify sets and reps before moving on to the next exercise
5. App will mark the selected workout as done after completion
6. If user choose to pause workout, app will prompt user if they want to continue the workout
 - a. If "Stop workout" is selected, app returns to workout page
 - b. If "Resume" is selected, timer continues to run

4.6.3 Functional Requirements

1. Users must be able to manually start the workout.

- a. The app must display a timer to show the length of their workout.
 - i. Users must be able to pause and start the timer.
2. The app must display the selected warm up video from the Youtube API.
 - a. Users must be able to play the video from the app.
 - b. Users shall be able to hear the audio if sound is enabled.
 - c. The app must prompt the users to finish their warm-up so they can move on to the exercises.
3. The app must display the exercise that the user should perform.
 - a. The app must display the name of the exercise
 - b. The app must display a description of the exercise.
 - c. The app must display the recommended number of sets and reps the user should perform.
4. Users must be able to add more sets or remove sets
 - a. Users must be able to complete the set.
 - b. Users must be able to move on to the next exercise after all sets are completed.
 - c. Users must be able to pause the workout at any time and go to the edit workout interface to edit the number of reps.
5. The app must display the selected cool down video from the Youtube API after finishing all the exercises.
 - a. Users must be able to play the video from the app
 - b. Users shall be able to hear the audio if sound is enabled.
 - c. The app must prompt the users to finish their cool-down to complete their workout.



4.7 Edit Workout Routine

4.7.1 Description and Priority

Users are able to edit their recommended workout routine. This feature is medium priority as user will be able to edit their workout according to their needs..

4.7.2 Stimulus/Response Sequences

1. On successful login, user selects the workout to be edited
2. App displays repetitions and sets to edit
3. Once update is verified and submitted, app displays updated workout
 - a. If incomplete information is submitted, app will display warning message and allow user to re-enter

4.7.3 Functional Requirements

1. The user is able to edit workouts by making modifications to the number of repetitions.
 - a. The user is also able to delete past workouts from the record

The screenshot shows the 'EDIT WORKOUT' screen. At the top, there is a back arrow and the title 'EDIT WORKOUT'. Below the title, there is a section for 'workout' with the text 'I love to workout'. The main content area shows a list of workouts. The first workout is 'Thruster' with three sets of 10 reps each. The second workout is 'Low Box Squat - Wide Stance'. A modal form is displayed over the 'Low Box Squat - Wide Stance' workout, allowing the user to edit the number of sets and repetitions. The modal has two input fields: 'Number of sets' with a value of 3 and 'Number of repetitions' with a value of 10. Below the modal, there is an 'UPDATE' button.

4.7 Edit workout

4.8 Display Workout Summary

4.8.1 Description and Priority

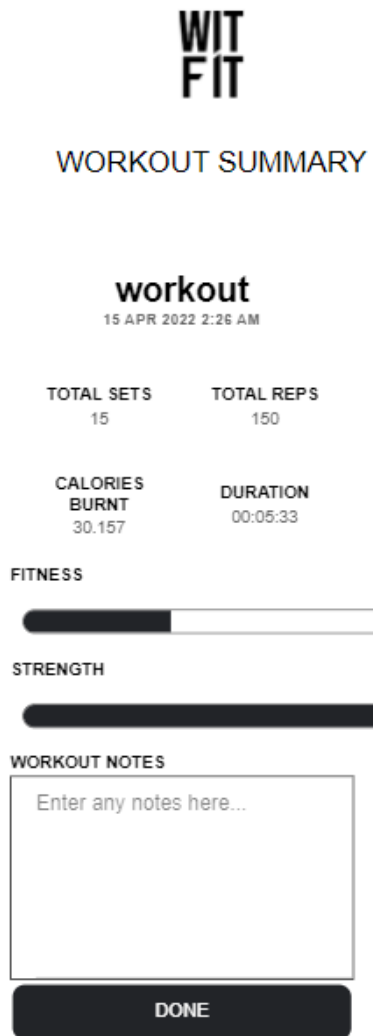
App is able to display summary of completed workout. This feature is medium priority as it is a key component of user experience

4.8.2 Stimulus/Response Sequences

1. Once user completes a workout, app displays a summary of the workout
2. Summary will display the time and date of the workout completed
3. App will calculate the total calories burnt, fitness and strength achieved
4. Summary will also display some basic statistics like total sets and reps of workout

4.8.3 Functional Requirements

1. The app must show the workout summary after the workout is complete
 - a. The app will display the time and date the workout was completed.
 - i. The app will also display some basic statistics like total sets, total reps, total calories and duration of workout



4.8 Workout summary

4.9 Display Weekly Statistics

4.9.1 Description and Priority

App is able to display summary statistics of the workouts completed for the week. This feature is high priority as this functionality will live on prominent location such as the home page where it is regularly exposed to users

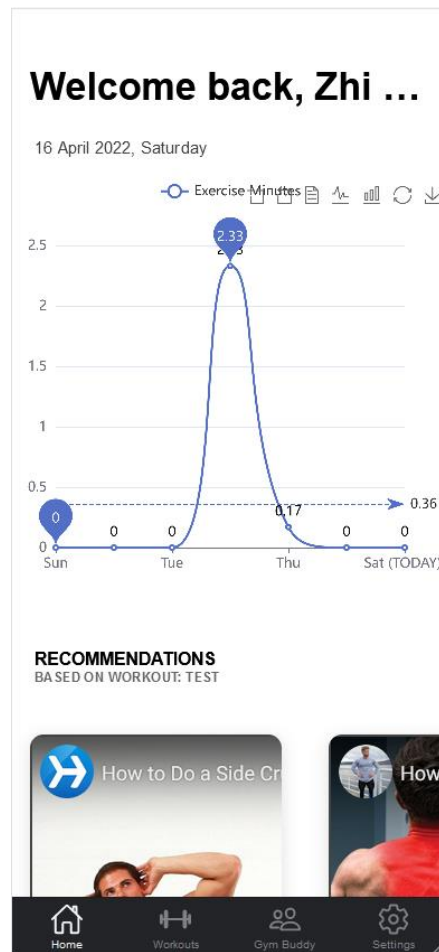
4.9.2 Stimulus/Response Sequences

1. App displays a weekly workout progress chart on the home page
2. User swipes the weekly progress chart.
3. App displays a table summary.
4. User selects 'See More'.

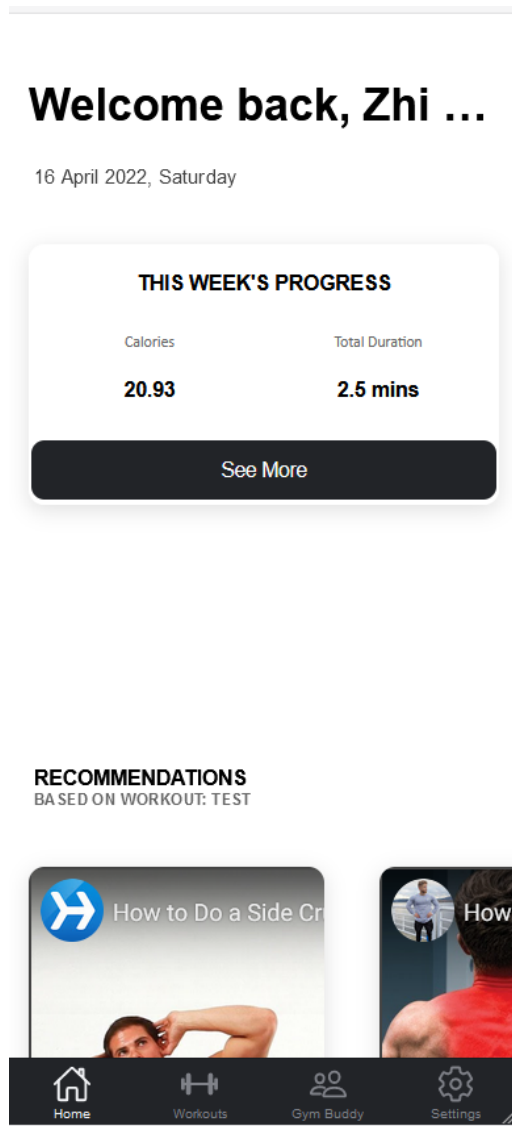
5. App displays completed workouts (Continuous from 4.11 *View completed workouts*)

4.9.3 Functional Requirements

1. The app shall display a weekly workout progress chart on the home page based on completed workouts.
 - a. Users shall be able to toggle the chart to see their weekly workout progress summary.



4.9 Home page displaying weekly progress chart



4.9 Home page displaying table summary

4.10 Display created workouts

4.10.1 Description and Priority

User is able to access and view past workouts. This feature is medium priority as it is a key component for users to track and view all their workouts that they have created. This function is an important motivating force for them to continue using the app

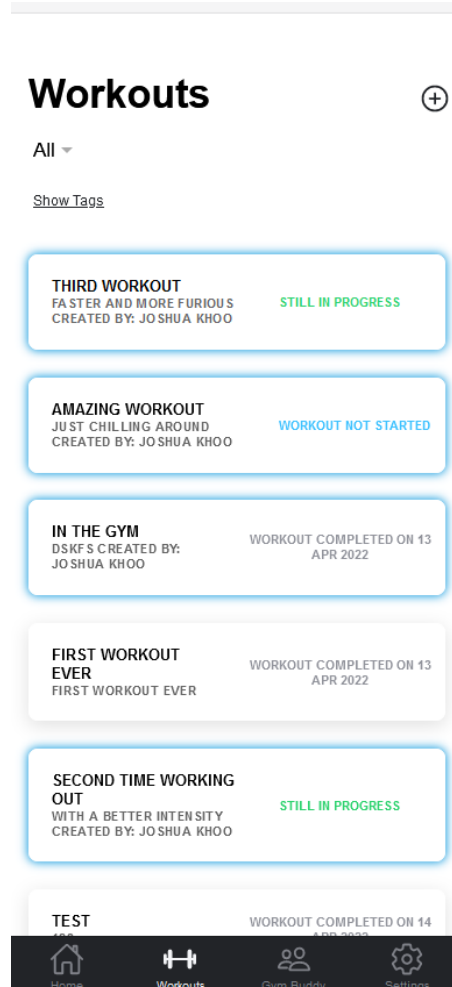
4.10.2 Stimulus/Response Sequences

1. App will display a list of all workouts with their title, description and whether it is completed
2. Users are able to filter their workouts based on their progress and whether it was created by a gym buddy.

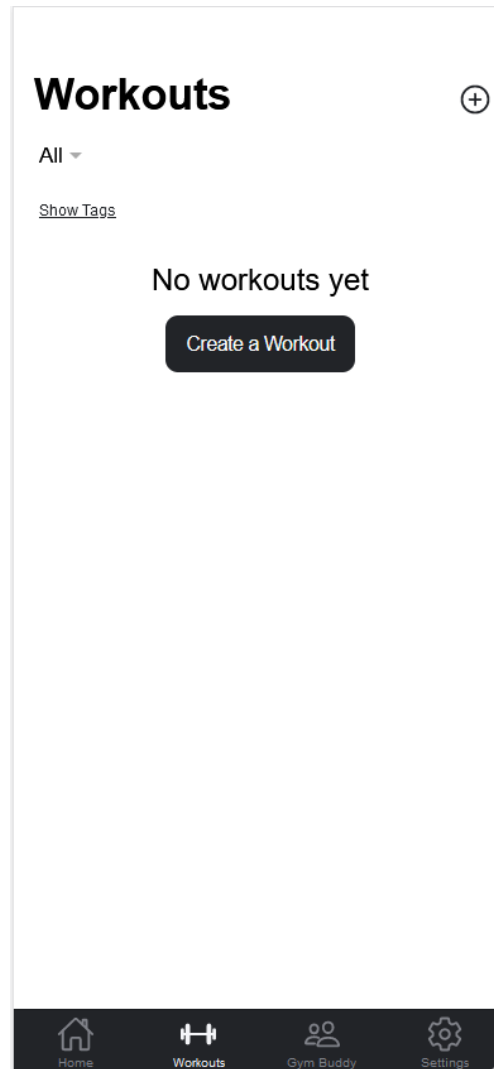
3. App will display “No workouts created” if user has not done any workout

4.10.3 Functional Requirements

1. The app shall display the workouts the user has created in the main workout interface.
 - a. The user shall be able to filter workouts that are not yet started.
 - b. The user shall be able to filter workouts that are in progress.
 - c. The user shall be able to filter workouts that are completed.
 - d. The user shall be able to filter My Buddy’s Workouts, which will be highlighted in a blue border.
 - e. The user shall be able to show tags of the workout corresponding to the options they selected when creating the workout.
 - f. The app shall display the completion status of each workout.
2. The app must prompt users to create a workout if they have never created a workout.



4.10 Workout page displaying created workouts



4.10 Workout page when no workout has been created

4.11 View completed workouts

4.11.1 Description and Priority

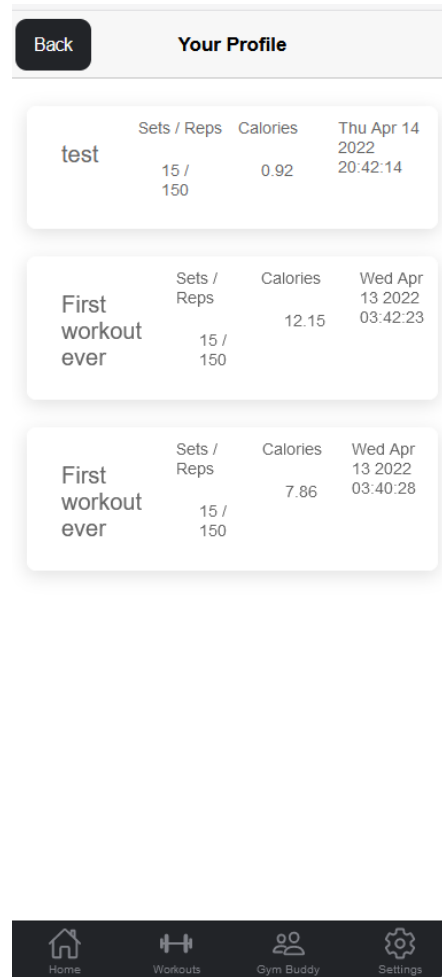
User is able to view past workouts. This feature is high priority as this functionality will live on a prominent location such as the homepage, where it is regularly exposed to users

4.11.2 Stimulus/Response Sequences

1. App displays completed workouts in the past week
2. App calculates and displays total calories burnt and duration spent exercising across the week

4.11.3 Functional Requirements

1. Users shall be able to see more of their progress when they click on the 'See More' button which will display their completed workouts.



4.11 Page showing workout completed in the past week

4.12 Display Recommended Videos

4.12.1 Description and Priority

User is able to view a list of Youtube videos recommended to them. This feature is medium priority as this functionality will live on a prominent location such as the homepage, where is it regularly exposed to users.

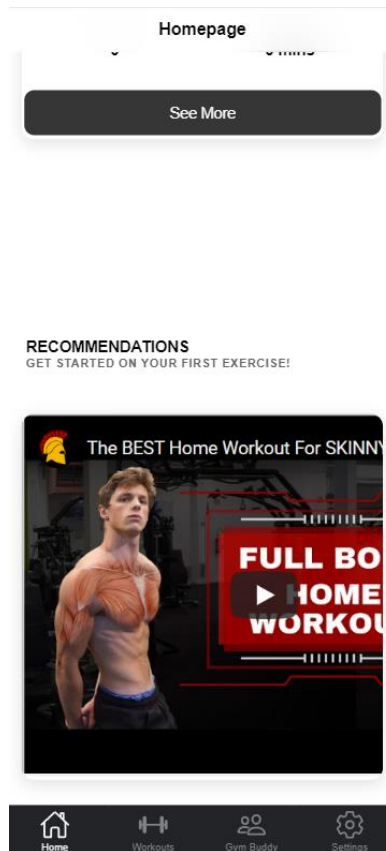
4.12.2 Stimulus/Response Sequences

1. On successful login, User is brought to the app's home page
2. System retrieves information about User's completed Workout (Workout name, exercise names, and user's gender)
3. System calls YoutubeService to search Youtube for these exercise names + gender
4. YoutubeService compiles the recommended videos and returns them to app system

- a. If YoutubeService is unavailable, it will return a default placeholder recommendation video
5. App displays this information on-screen.

4.12.3 Functional Requirements

1. The app shall display Youtube video recommendations based on the user's most recent workouts.
 - a. Users should be able to click on the videos which will route the user to youtube directly.



4.12 Recommended youtube video

4.13 Create Gym Buddy Account

4.13.1 Description and Priority

User is able to create user profile for gym buddy application function. This feature is high priority as users will not be able to log in to access any of the “find a gym buddy” functionalities without an account

4.13.2 Stimulus/Response Sequences

1. App will prompt user to enter their gym buddy details which includes abrief introduction,preferred workout time, preferred buddy, gender, goals, areas of expertise, style of training, preference of training location and photo of themself.
2. Upon successful verification, account is created.
 - a. If not all fields are filled, app will not allow user to submit form and prompts to complete
3. If user choose not to upload image, app will update the database with a default image based on gender of the user

4.13.3 Functional Requirements

1. The app shall prompt the user to create a gym buddy account for first time gym buddy users.
 - a. The app must display the user's name based on their WITFIT account.
 - b. The app must prompt the user to upload at least 1 photo of themselves.
 - i. The user's profile shall be displayed with a default male or female image based on their gender if no image is uploaded.
 - c. The app must prompt the user for the preferred gender of their buddy.
 - d. The app must prompt the user to input a brief description of themselves.
 - e. The app must prompt the user for their preferred workout timings
 - i. The app must allow users to select multiple blocks of timings.
 - f. The app must prompt the user for the goals they want to achieve with their buddy.
 - i. The app must display a list of goals.
 - ii. The app must allow the user to select up to 3 goals.
 - g. The app must prompt the user for their strengths (area of expertise)
 - i. The app must display a list of areas of strengths.
 - ii. The app must allow users to select up to 3 areas of strength
 - h. The app must prompt the user for their style of training.
 - i. The app must display a list of styles of training.
 - ii. The app can allow the users to select up to two styles of training.
 - i. The app must prompt the user to enter their preferred location of training.
 - i. The app must display a list of potential training locations.
 - ii. The app must allow users to select up to 2 desired training locations.
 - j. The app must prompt the user to enter the desired traits of their buddies.
 - i. The app must display a list of traits for their buddies.
 - ii. The app must allow users to select up to 3 desired training traits.
 - k. The app must prompt the user to enter the desired training styles of their buddies.
 - i. The app must display a list of training styles for their buddies.
 - ii. The app must allow users to select up to 2 desired training styles.

Gym Buddy Sign Up

ad cd

Brief Introduction

Brief intro here...

Choose Images to Upload

Browse... No file selected.

Preferred Workout Time

☐ MORNING 5AM-11AM

☐ AFTERNOON 11AM-5pm

☐ EVENING 5PM-11PM

☐ LATE NIGHT 11PM-2AM

☐ NO PREFERENCE ANYTIME

Preferred Buddy Gender

☐ MALE

☐ FEMALE

☐ NO PREFERENCE

Goals you want to achieve with your buddy

Select up to 3 strengths

Home Workouts Gym Buddy Settings

4.13 Gym buddy account creation page

4.14 Display Gym Buddy Home Page

4.14.1 Description and Priority

App will display the gym buddy home page with options to explore gym buddy functionalities. This feature is high priority as functionalities such as (Update gym buddy preferences, View suggested Buddy details, Request to pair up with buddy, View all chats, Chat with buddy, Send message, Remove buddy, Share workout) will not be available without implementing this functionality.

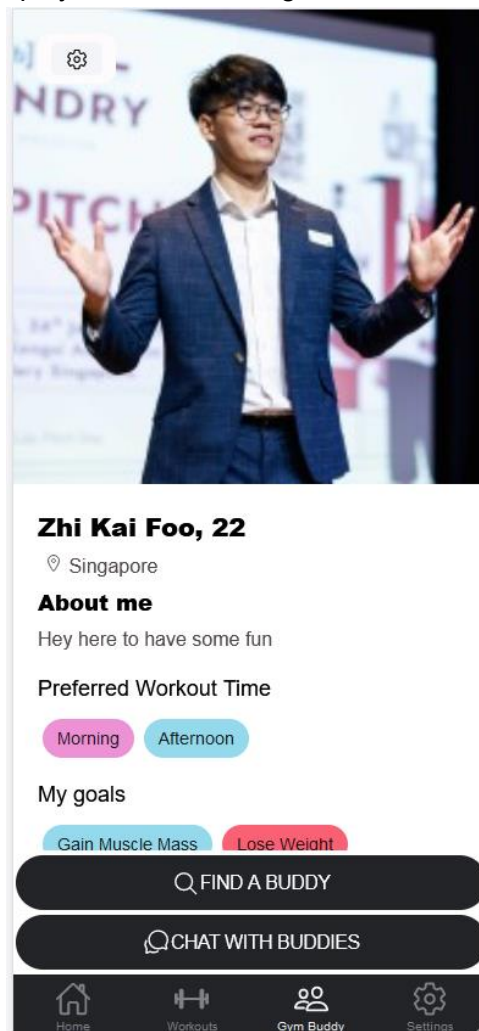
4.14.2 Stimulus/Response Sequences

1. User selects "Gym Buddy" tab

2. App verifies account with database and displays user's gym buddy profile indicating successful log in

4.14.3 Functional Requirements

1. The app shall display the user's profile if a gym buddy account has been created before.
 - a. The app shall display the user's uploaded image.
 - i. The app shall display a default image depending on gender if no image was uploaded.
 - b. The app shall display the user's name and age.
 - c. The app shall display the user's brief introduction.
 - d. The app shall display the user's preferred workout time.
 - e. The app shall display the user's goals.
 - f. The app shall display the user's strengths.



4.14 Gym buddy home page

4.15 Update Gym Buddy Account Preferences

4.15.1 Description and Priority

User is able to update their account preference. This feature is medium priority as users will be able to update their gym buddy details, allowing the algorithm to provide them with a more accurate match.

4.15.2 Stimulus/Response Sequences

1. App will prompt user to enter brief introduction (preferred workout time, preferred buddy, gender, goals, areas of expertise, style of training, preference of training location and photo of themselves)
2. Upon successful verification, account preferences will be updated
 - a. If no new updates, nothing changes
3. If user choose not to upload image, app will update the database with a default image link based on gender of the user

4.15.3 Functional Requirements

1. The app shall prompt the user to update their gym buddy account.
 - a. The app must display the user's name based on their WITFIT account.
 - b. The app must display previously inputted gym buddy details.
 - c. The app must prompt the user to upload at least 1 photo of themselves.
 - i. The user's profile shall be displayed with a default male or female image based on their gender if no image is uploaded.
 - d. The app must allow users to update their preferred gender of their buddy.
 - e. The app must allow users to update their brief description of themselves.
 - f. The app must allow users to update their preferred workout timings.
 - i. The app must allow users to select multiple blocks of timings.
 - g. The app must allow users to update the goals they want to achieve with their buddy.
 - i. The app must display a list of goals
 - ii. The app must allow the user to select up to 3 goals
 - h. The app must allow users to update their strengths (area of expertise)
 - i. The app must display a list of areas of strengths
 - ii. The app must allow users to select up to 3 areas of strength.
 - i. The app must allow users to update their style of training.
 - i. The app must display a list of styles of training
 - ii. The app can allow the users to select up to two styles of training.
 - j. The app must allow users to update their preferred location of training.
 - i. The app must display a list of potential training locations.
 - ii. The app must allow users to select up to 2 desired training locations.
 - k. The app must allow users to update their desired traits of their buddies.
 - i. The app must display a list of traits for their buddies.
 - ii. The app must allow users to select up to 3 desired training traits.

- I. The app must allow users to update their desired training styles of their buddies.
 - i. The app must display a list of training styles for their buddies.
 - ii. The app must allow users to select up to 2 desired training styles.

UPDATE YOUR PROFILE

Name
Zhi Kai Foo

Brief Introduction
Hey here to have some fun

Choose Images to Upload
Browse... No file selected.

Preferred Workout Time

☒ Morning 5AM-11AM

☒ Afternoon 11AM-5pm

☐ EVENING 5PM-11PM

☐ LATE NIGHT 11PM-2AM

☐ NO PREFERENCE ANYTIME

Preferred Buddy Gender

☒ MALE

☐ FEMALE

Home Workouts Gym Buddy Settings

4.15 Gym buddy update account preference page

4.16 View Suggested Buddy Details

4.16.1 Description and Priority

User is able to view one suggested buddy. This feature is medium priority as this functionality will enable users to find buddies to match. It is a core feature of the gym buddy functional requirement without which it will not work to fulfil its purpose. Having a workout partner is important to motivate users to continue working out and keep each other accountable for their fitness goals.

4.16.2 Stimulus/Response Sequences

1. User clicks "Find a buddy"
2. System retrieves current user information from data base
3. System retrieves a dictionary of gym buddy profiles filtered by current user's preferences from data base

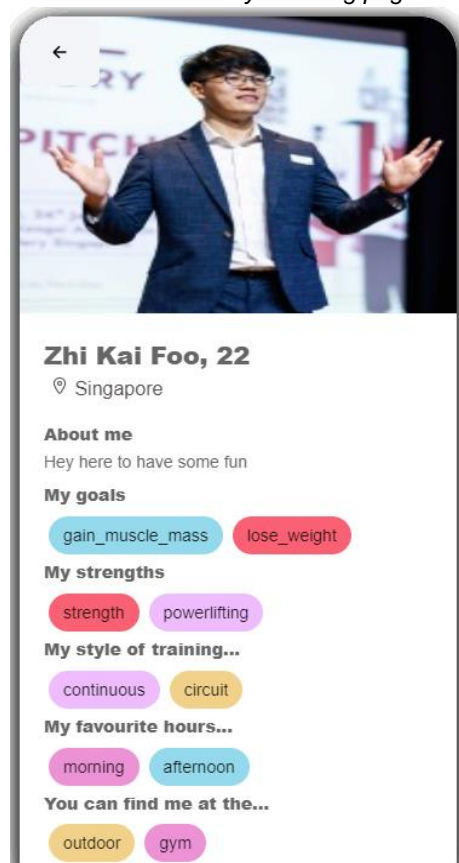
4. System calculates the matching score between current user and other profiles
5. The system then polls the score of all users and arrange the scores in decreasing order
6. System sends an array of potential matches
7. The app displays the suggested buddy profile with the highest score for swiping (Choosing - left to reject, right to accept)
 - a. If no more suggested profile, app will display a blank screen

4.16.3 Functional Requirements

1. The app must allow users to find a buddy.
 - a. The app must display a welcome message and prompt the user to agree with certain usage rules.
 - i. The app must display a “Be Motivated” message.
 - ii. The app must display a “Build Your Own Community” message.
 - iii. The app must display a “Have Fun” message.
 - b. The app shall use an algorithm to suggest potential gym buddies.
 - i. The algorithm shall include preferences of the user.
 - ii. The algorithm will place priority on gender preference.
 - c. The app must allow users to swipe right to match a buddy for swipe left to not match the recommended match.
 - i. The user must swipe pass a certain horizontal distance before the app will display the next match.
 - ii. The app shall display a green interface while the user is swiping right.
 - iii. The app shall display a red interface while the user is swiping left.
 - iv. The app shall display detailed information of the suggested buddy.
2. The app shall display the recommended buddy’s profile picture.
3. The app shall display the recommended buddy’s name and age.
4. The app shall display the recommended buddy’s brief introduction.
5. The app shall display the recommended buddy’s goals.
6. The app shall display the recommended buddy’s strengths.
7. The app shall display the recommended buddy’s style of training.
8. The app shall display the recommended buddy’s favourite hours.
9. The app shall display the recommended buddy’s preferred location.
10. The app must allow users to go back at any point in time.



4.16 Find a buddy boarding page



4.16 Page showing suggested buddies

4.17 Request to pair up with buddy

4.17.1 Description and Priority

User selects to pair up with suggest buddy. This feature is medium priority as this functionality will allow users to request to pair up with a buddy. It is a core feature of the gym buddy functional requirement without which it will not work to fulfil its purpose. Having a workout partner is important to motivate users to continue working out and keep each other accountable for their fitness goals.

4.17.2 Stimulus/Response Sequences

1. When the user swipes right (Indicating a match)
2. System checks with recommended user had previously matched with current user
3. Once validated, system will update the current user information and create a new chat (If match)
 - a. If recommender user has yet to match with current user, app will display next available buddy
4. App will display next available buddy

4.17.3 Functional Requirements

1. The app must allow users to pair up with a suggested buddy.
 - a. If the buddy has already opted to pair up, then there is a match.
 - i. The app must display a new chat tagged as new in the buddy list interface.
 - ii. The app must display the buddy in the user's buddy list interface.
 - iii. The app must also display the user in the buddy's buddy list interface.
 - b. If the buddy has yet to opt to pair up or the buddy has already opted not to pair up, there will not be a match.
 - i. The suggested buddy will be removed from the user's list of suggested buddies.
 - c. If the buddy opts to pair up subsequently, there will be a match.
 - i. The app must display the buddy in the user's buddy list interface.

4.18 View all chats

4.18.1 Description and Priority

User is able to view all existing chats with the user's matches. This feature is medium priority as this function represents the main platform for users to interact with their buddies.

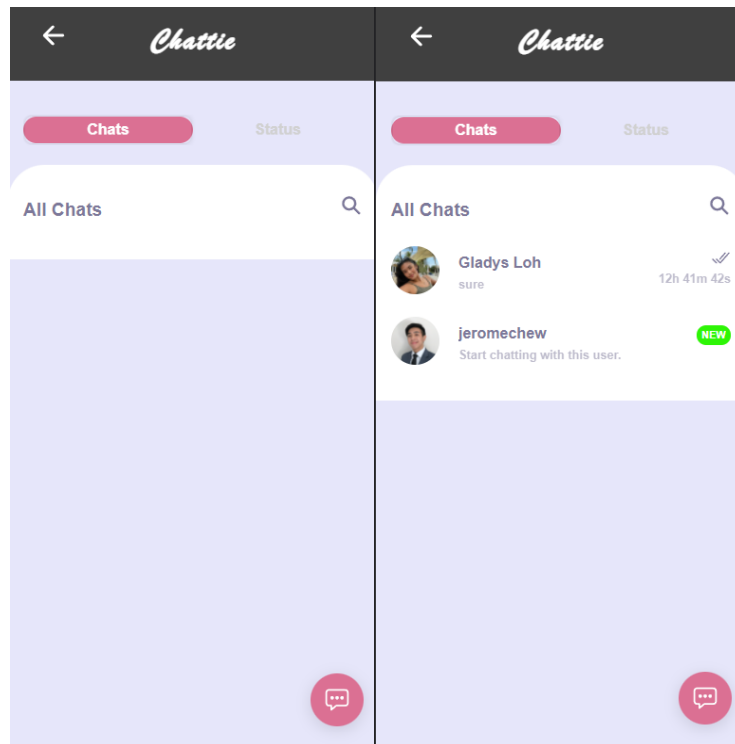
4.18.2 Stimulus/Response Sequences

1. User selects 'View Buddy List'.
2. System retrieves current chat user, all chat information

3. Buddy list home page retrieves the full name, profile picture, last message and timestamp from the system
4. App displays all the chats to the user.

4.18.3 Functional Requirements

1. The app must display a list of buddies that are successful matches.
 - a. The app shall display the buddy's profile picture, last message, full name and timestamp of last message for each chat.
 - b. The app must allow users to click into each chat and send messages freely if a match is made.



4.18 View all chats page

4.19 View Chat With Buddy

4.19.1 Description and Priority

User is chooses to click into a specific chat from the list of all chats. This feature is medium priority as this functionality will allow users to chat with a matched buddy. It is a core feature of the gym buddy functional requirement without which it will not work to fulfil its purpose. Having a workout partner is important to motivate users to continue working out and keep each other accountable for their fitness goals.

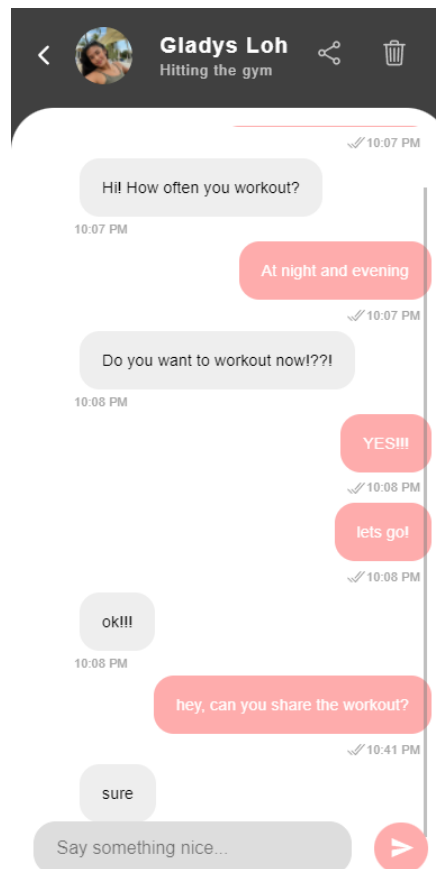
4.19.2 Stimulus/Response Sequences

1. User clicks view chat.

2. System returns all chat messages to the chat page which displays the chat messages to the user
3. The system notifies updates the chat page whenever there is a new message which then displays the latest chat message to the user

4.19.3 Functional Requirements

1. The app must display the exchange of messages between the user and the buddy in real time



4.19 Chat with buddy

4.20 Send Message

4.20.1 Description and Priority

User chooses to send a message in a chat rom with another buddy. This feature is medium priority as this functionality will allow users to chat with a matched buddy. This use case is necessary, else it defeats the purpose of having the chat functionality altogether.

4.20.2 Stimulus/Response Sequences

1. User enters message to send to gym buddy.

2. User sends message.
3. App displays sent message in the chat.

4.20.3 Functional Requirements

1. The app must be able to send and receive messages

4.21 Remove Buddy

4.21.1 Description and Priority

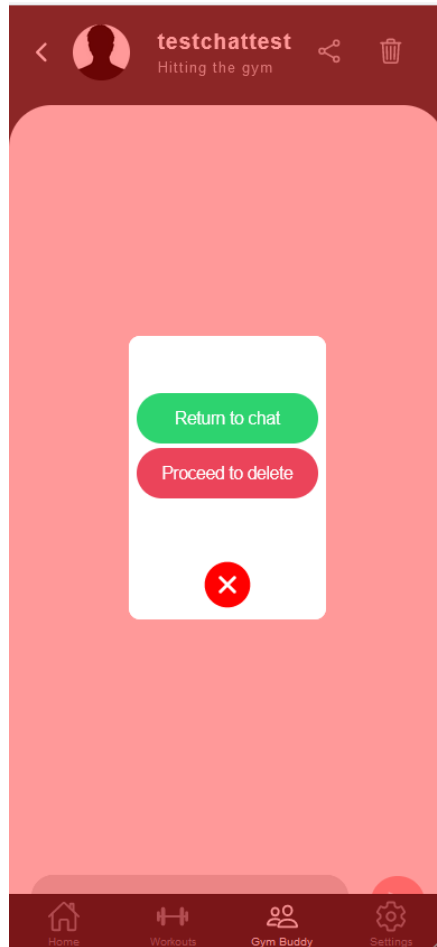
User can remove a buddy from their match list. This feature is low priority as this function is only to enhance overall user experience, allowing users to remove buddies that they do not desire, or feel uncomfortable chatting with.

4.21.2 Stimulus/Response Sequences

1. User clicks the trash icon.
2. App displays option for user to return to chat or proceed to delete
3. User selects the proceed to delete button.
4. Chat is removed from user's buddy list.

4.21.3 Functional Requirements

1. The app must allow users to have an option to remove the buddies.
 - a. The app shall display a confirmation removal message.
 - i. The user shall be given the option to proceed.
2. The app must remove the buddy from the user's buddy list display.
3. The app must remove the user from the buddy's buddy list display as well.
 - a. The user shall be given the option to go back.
4. The app shall return to the chat list display and exclude the chat that has been removed.



4.21 Confirmation prompt for user to delete the chat

4.22 Share Workout

4.22.1 Description and Priority

User can choose to share their workouts with a buddy This feature is medium priority as this function integrates the gym buddy functionality of our application with the workout functionality and is a core feature to allow users to share a common goal of being fit together.

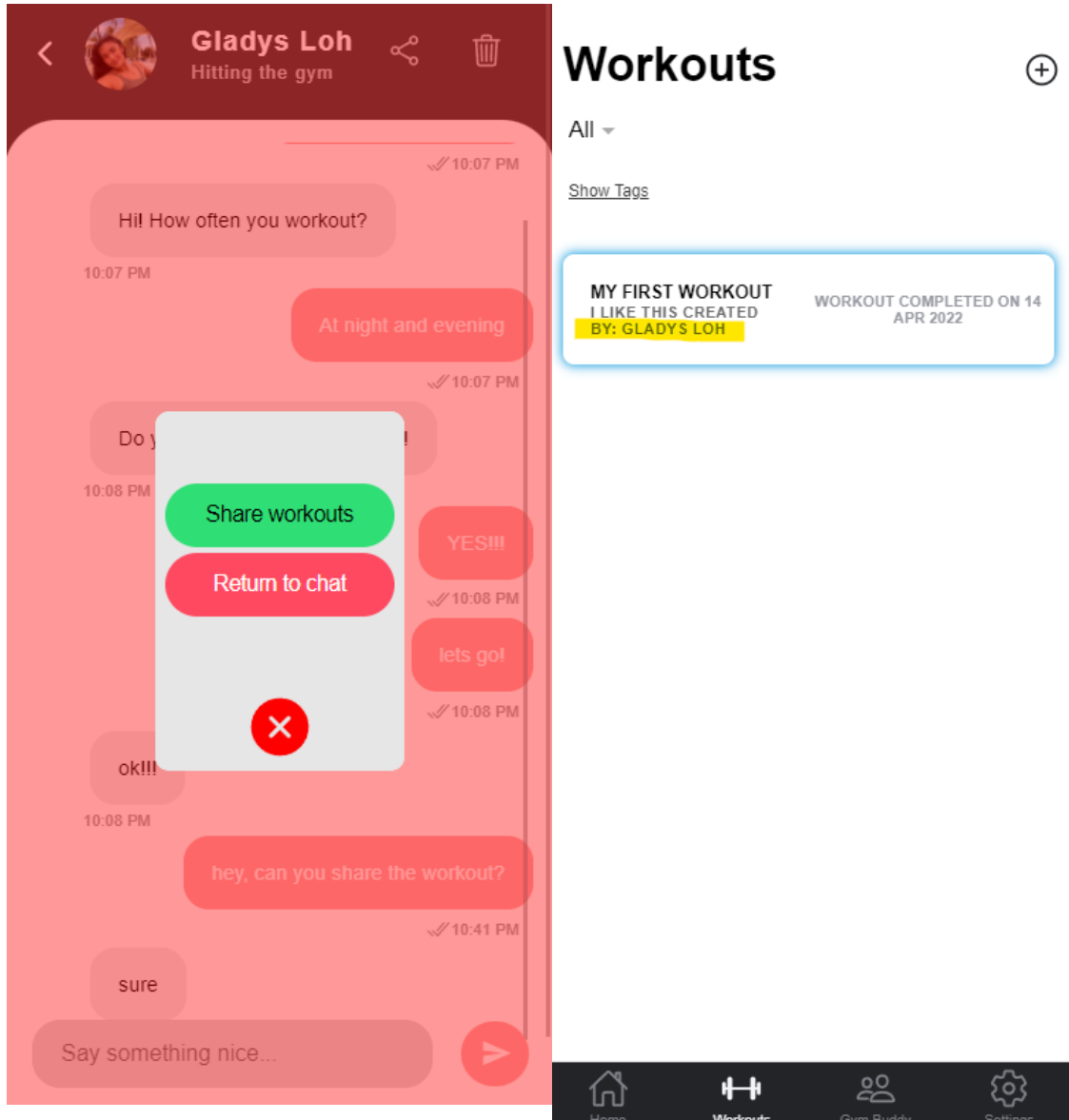
4.22.2 Stimulus/Response Sequences

1. User clicks on the share workout button in the Chat interface.
2. App displays a confirmation prompt to either share workout or return to chat.
3. User selects share workouts.
4. App will share user's workout with gym buddy.
5. Gym buddy's app will display the shared workout in their workout tab.

4.22.3 Functional Requirements

1. The app must allow users to have an option to share their workouts.
 - a. The app shall display a confirmation share workout message

- i. The user shall be given the option to proceed.
2. The app must share non-duplicated workouts with the buddy.
3. The buddy must be able to see the shared workouts in the Workout home user interface with the shared workouts highlighted by a blue border.
 - a. The user shall be given the option to go back.



4.22 Sharing of workout

5 Other Nonfunctional Requirements

5.1 Performance Requirements

User can choose to share their workouts with a buddy This feature is medium priority as this function integrates the gym buddy functionality of our application with the workout functionality and is a core feature to allow users to share a common goal of being fit together.

1. Each page must load within 2 seconds, with the exception of the creation customised workout option.
2. The application must be able to service up to 100 concurrent users.
3. The application must control up to 2 drill presses concurrently.
4. Assuming a good internet connection, the chat messages shall be displayed for each user within 1 second of sending the message.
5. The app must be compatible on Android and iOS devices.
6. The app must interface with the application programming interface from data.gov.sg to fetch gym information
7. The app must interface with Youtube and workout applicable programming interfaces to fetch exercises and workout information.
8. The app must interface with Spotify's applicable programming interface to get music playlist information.
9. The database used to store information shall be able to migrate information to other devices easily.
10. The system should be able to handle a load of up to 100 users at any one time.

5.2 Safety Requirements

1. The app must provide workout recommendation that prevents aggravation of existing injuries.
2. The app must provide workout recommendation that will not result in physical injury to the user.
3. The app must provide overall safety guidelines regarding workouts.
4. The app must recommend users to perform a warm up prior to the actual workout to reduce the risk of injury.
5. The app must recommend users to perform a cool down after the actual workout to reduce the risk of injury.
6. The app must inform users of the proper guidelines when using the find a buddy feature

5.3 Security Requirements

1. Passwords shall be hashed using SHA256, and must contain at least 6 characters.
2. The app shall use Firebase Authentication and Firestore Security Rules to handle serverless authentication, authorisation and data validation.
3. The app shall use Identity and Access Management (IAM) to manage access to the database.

5.4 Software Quality Attributes

a. Reliability:

1. Users must be able to use the app without any issues, 24 hours a day, 7 days a week.

b. Maintainability:

1. The application design shall adhere to SOLID principles for code maintainability.
2. The application shall adhere to design patterns - Observable for the chat interface, Singleton for Youtube API & largely MVC pattern, for software extensibility.
3. The app shall be documented using JSDocs to ensure that developers can easily take over code, for maintainability.
4. The code used in the application prototype must conform to ESLint Conventions for readability.

c. Usability:

1. The app (including help messages) shall all be in English.
2. First time users must be able to learn how to use the different functionalities within the application within 5 minutes.
3. After account creation, 80% of first-time users must be able to enter a simple query such as creating a workout or finding a buddy within 2 minutes of starting to use the system.
4. If the user's network connection is poor or unavailable, the app must display a toast prompting the user to try again with a stable network connection.
5. The user should receive meaningful alerts like 'User created', 'Workout saved' in the application flow to aid user experience.

d. Supportability:

1. The app must be able to support new features or upgrades in the future.
2. The app must have monthly updates to resolve bug fixes.

5.5 Business Rules

1. Only users with a specified gender (i.e. not 'Others') will receive 4.12 Recommended videos personalised to their gender under the Workout Summary and Recommendation facet.
2. Only users with the gender of 'Female' will be given the option to input their menstrual cycle information.
3. Only users who registered for a main app account through 4.1 Create main app account under Account Management will be able to patronise the application's services. However, not every user is required to go through 4.13 Create gym buddy account.
4. Only users who registered for a gym buddy account through 4.13 Create gym buddy account will be able to utilise functionality under the facet of Gym Buddy Interactivity.
5. Users will not come with 'default' workouts. They will have to generate one themselves based on their user information in 4.5 Recommend workout.

6. Users must have a workout stored with their account in order to 4.10 Conduct workouts and hence complete a workout for displaying under 4.11 View finished workouts.
7. Users will only be able to 4.17 Request to pair with other users who also have a Gym Buddy account.
8. Users will only be able to 4.20 Send message to each other when they are paired, and neither user has 4.21 Remove buddy on the other.

6 Other Requirements

e. Infrastructure:

1. The database shall be stored in the Firebase cloud platform using Cloud Firestore
2. All platforms must interface with the database to get user information
3. All user input must be stored in the database upon exiting the program.
4. All user input shall be stored locally on the device during application runtime.
5. The app shall be developed using the Ionic Angular Framework which uses Typescript.

Appendix A: Glossary

For all the terms necessary to properly interpret the SRS, please reference '1.3. BCS3-WIT: Data Dictionary'.

Appendix B: Analysis Models

Class diagrams, please reference File 2.1. BCS3-WIT: Class Diagram

Sequence diagrams, please reference File 2.2. BCS3-WIT: Sequence Diagram

State-transition diagrams, please reference File 2.3. BCS3-WIT: Dialog Map