WIT
FIT

WITFIT

Working in Tandem, For Fitness

Team: Women In Tech
Lab Group: BCS3

Joshua Khoo	U2021421C
Foo Zhi Kai	U2022416G
Gladys Loh	U20
Gabriel Tang	U2021970J
Bryan Leow	U2021729K

Executive Summary

- Introduction
- Overview of Infrastructure
- How we approach the process of building the application
- How users travel through the application
- How we plan to continue growing the application
- Conclusion

Ethan

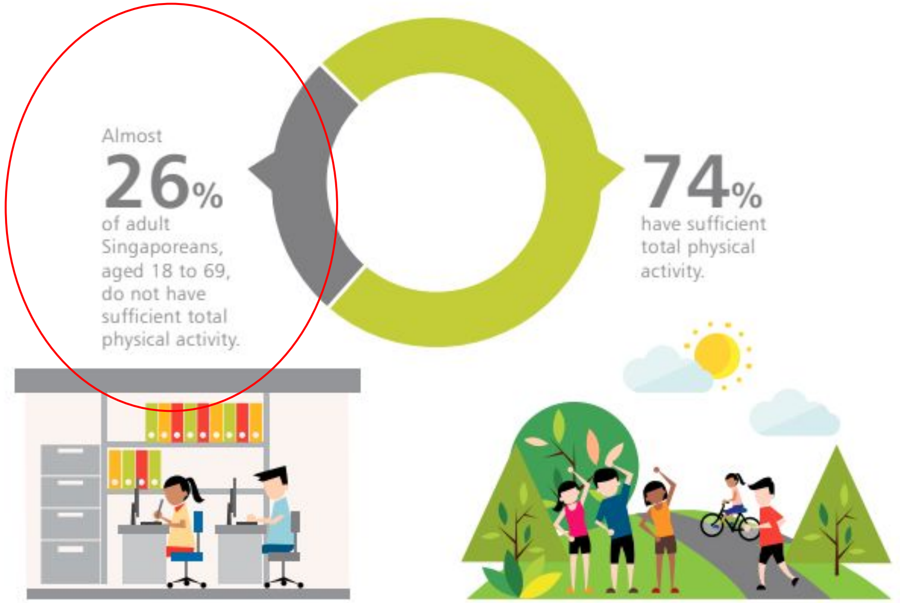
Current setting in Singapore

THE STRAITS TIMES
RESET 2021 WEBINAR SERIES
KEEPING SINGAPORE HEALTHY
PRUDENTIAL
RESET 2021 WEBINAR SERIES
KEEPING SINGAPORE HEALTHY

LEVERAGING TECH TO PROMOTE FITNESS

Ethan

Current setting in Singapore



Introducing

WIT
FIT

About Us

What is WITFIT?

- Cross platform mobile application
- Allow users to track workouts
- Allow users to find a buddy to achieve fitness goals

Target Audience:

- People of all ages looking to keep fit
- Anyone seeking a companion to keep fit

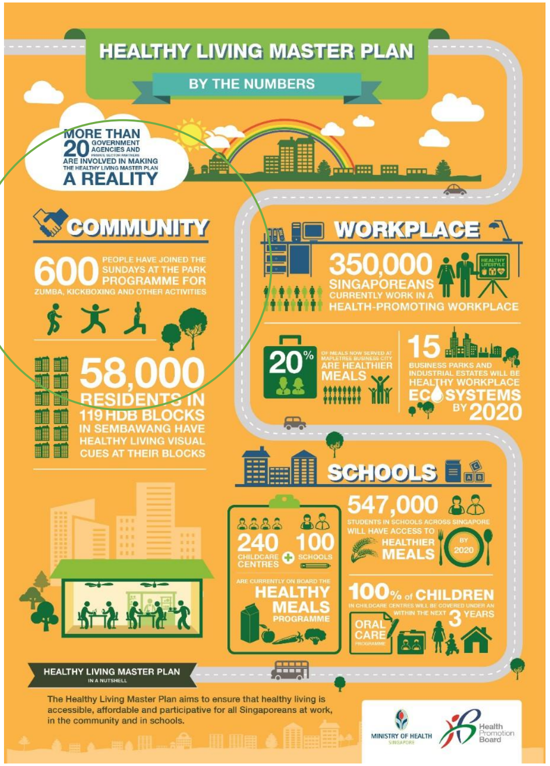
Business Problem

Why WITFIT?

- Currently little fitness applications that caters to needs of both men and women
- NO application that really makes women feel 'special'
- Many users hopping between fitness applications to find a one size fits all
  - Exhausting and unsustainable process

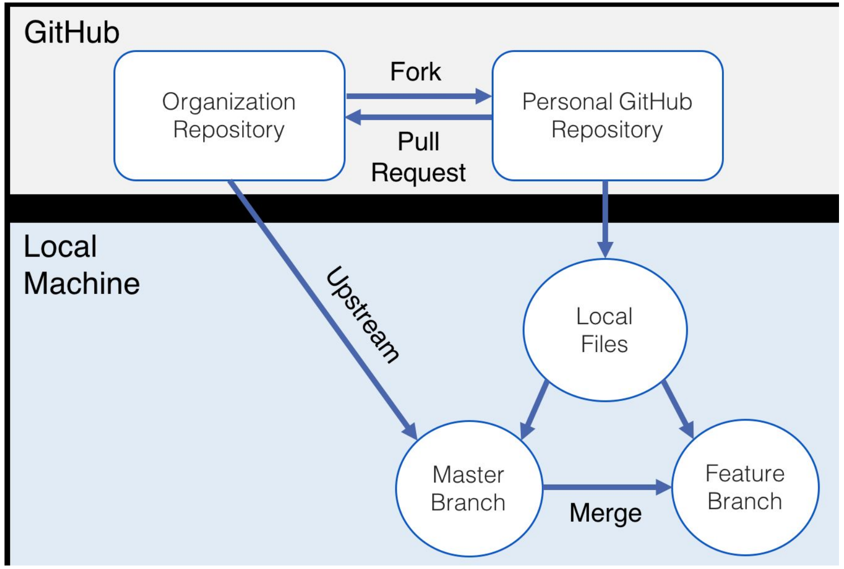
Business Problem

Singapore’s goals of a healthy nation


HEALTHY LIVING MASTER PLAN
BY THE NUMBERS
MORE THAN 20 GOVERNMENT AGENCIES AND ARE INVOLVED IN MAKING THE HEALTHY LIVING MASTER PLAN A REALITY
COMMUNITY
600 PEOPLE HAVE JOINED THE SUNDAYS AT THE PARK PROGRAMME FOR ZUMBA, KICKBOXING AND OTHER ACTIVITIES
58,000 RESIDENTS IN 119 HDB BLOCKS IN SEMBAWANG HAVE HEALTHY LIVING VISUAL CUES AT THEIR BLOCKS
WORKPLACE
350,000 SINGAPOREANS CURRENTLY WORK IN A HEALTH-PROMOTING WORKPLACE
20% ARE HEALTHIER MEALS
15 BUSINESS PARKS AND INDUSTRIAL ESTATES WILL BE HEALTHY WORKPLACE ECOSYSTEMS BY 2020
SCHOOLS
547,000 STUDENTS IN SCHOOLS ACROSS SINGAPORE WILL HAVE ACCESS TO HEALTHIER MEALS BY 2020
240 CHILDCARE CENTRES
100 SCHOOLS
ARE CURRENTLY ON BOARD THE HEALTHY MEALS PROGRAMME
100% of CHILDREN IN CHILDCARE CENTRES WILL BE COVERED UNDER AN ORAL CARE PROGRAMME WITHIN THE NEXT 3 YEARS
HEALTHY LIVING MASTER PLAN IN A NUTSHELL
The Healthy Living Master Plan aims to ensure that healthy living is accessible, affordable and participative for all Singaporeans at work, in the community and in schools.
MINISTRY OF HEALTH
Health Promotion Board

BUILDING AN ACTIVE AND HEALTHY COMMUNITY THROUGH DEVELOPING A HEALTHY LIFESTYLE

Project Management

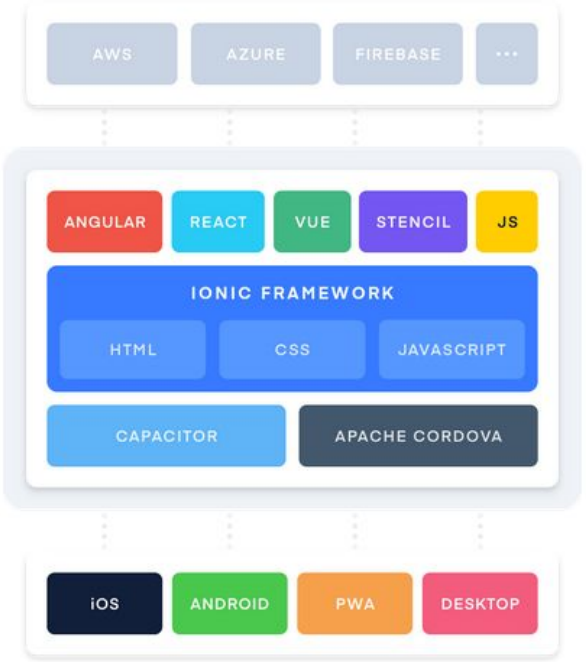


Why we decided to use GitHub

Having a GitHub repo made it easy for us to keep track of the entire project

- Provides easy storage for different kinds of files as the project develop.
- Comprehensive history for each file which makes it easy to explore the changes that occurred to it at different time points
- Allows us to easily review each other's code and suggest changes

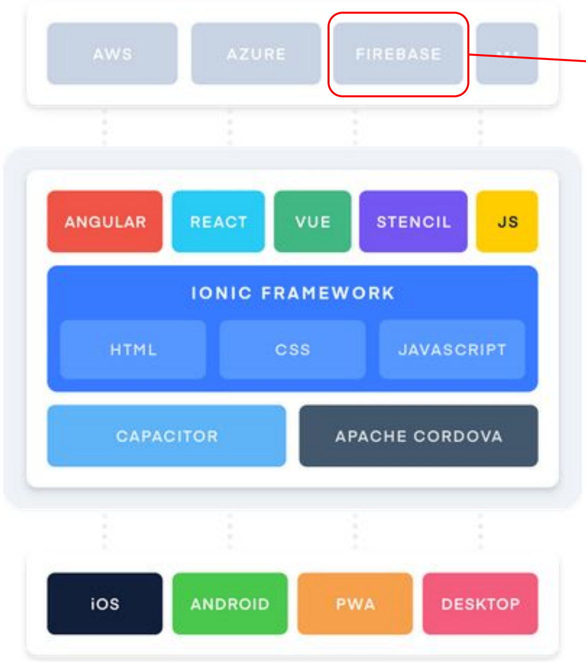
Technology Stack



Why we decided to use the Ionic framework

- Based on well-known technologies i.e. Angular, HTML, CSS and JavaScript
- Compatibility with React, Angular and Vue frameworks and supports Cordova plugins
- Ionic has a wide range of tools, plugins and UI components
- Ionic is backed by a vibrant community of developers and sufficient documentation

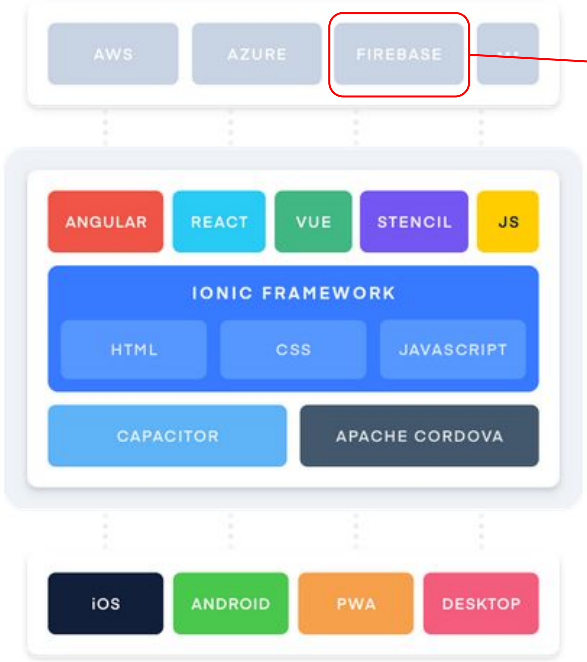
Technology Stack



Why we decided to use FireBase

- A wide range of services and features
- **Free basic plan**
- Concise documentation
- **Quick and easy integration and setup**

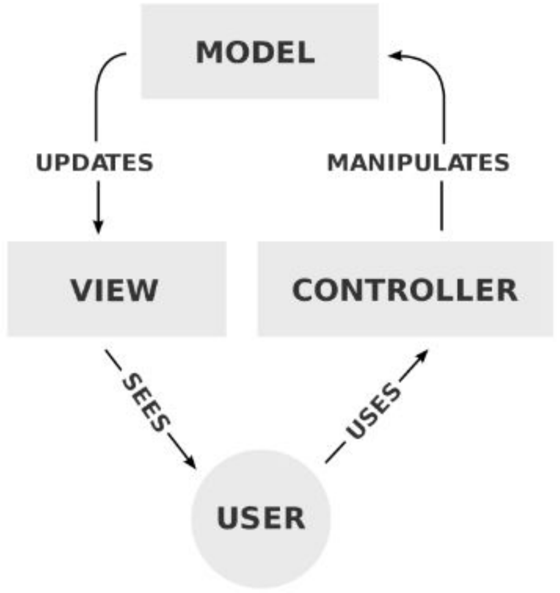
Technology Stack



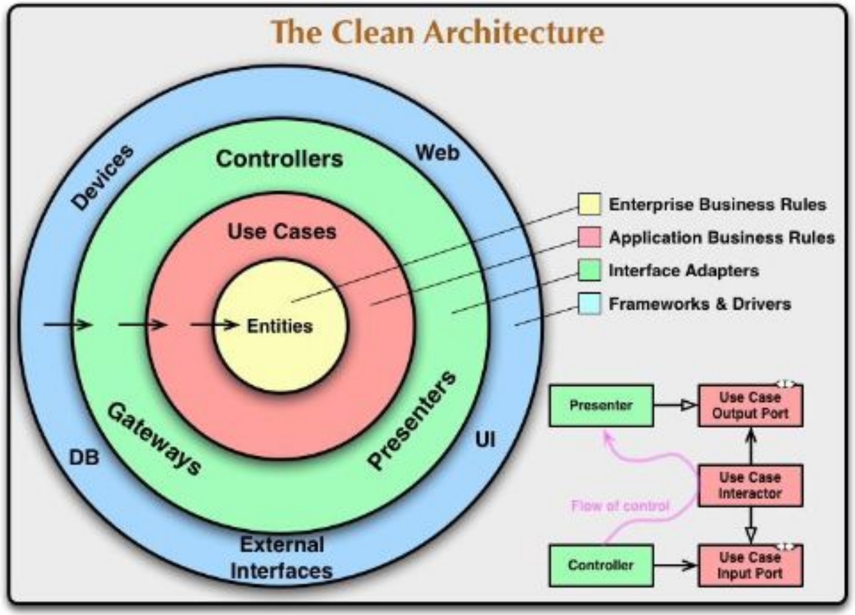
Why we decided to use FireBase

- A wide range of services and features
- **Free basic plan**
- Concise documentation
- **Quick and easy integration and setup**

Software Architecture Design

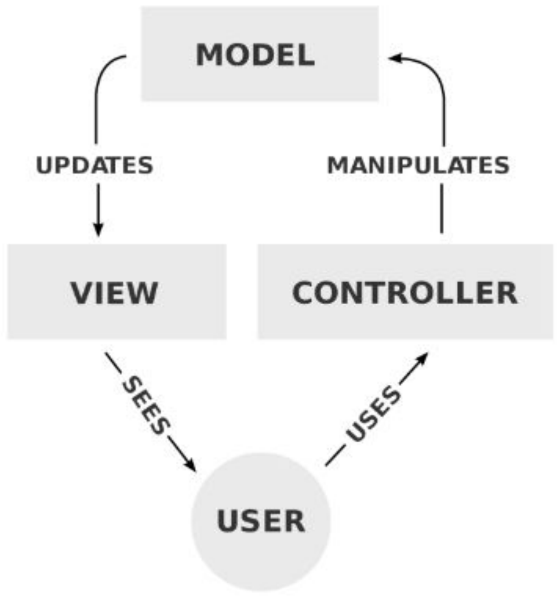


MVC Pattern



Clean Architecture

Why our group decided to use the MVC Design Pattern?



MVC Pattern

The pattern divides the application into 3 components : Model, View and Controller.

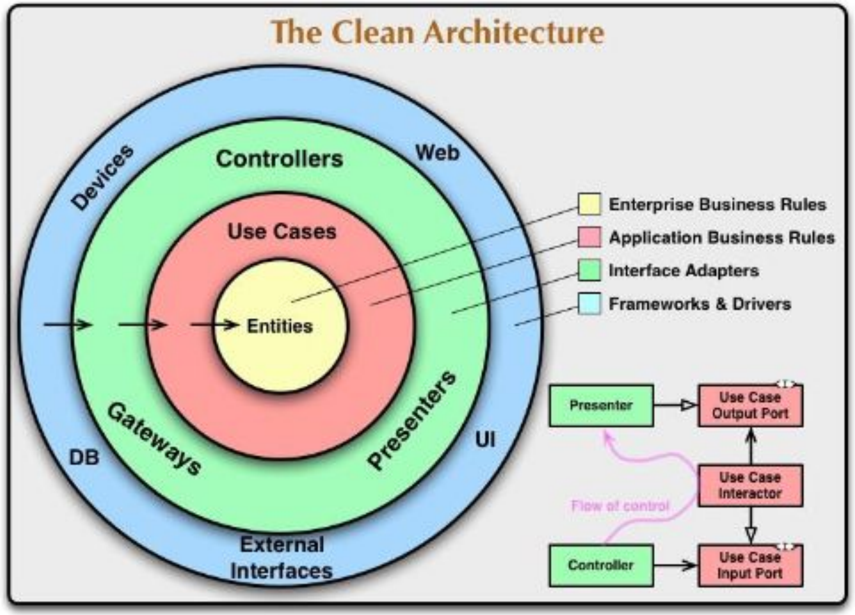
This design distinguishes the presentation of data from how the data is accepted from the user to the data shown

- Improves reusability of code
- Improves ability to develop the application in parallel since components are independent of each other in nature

Software Architecture Design

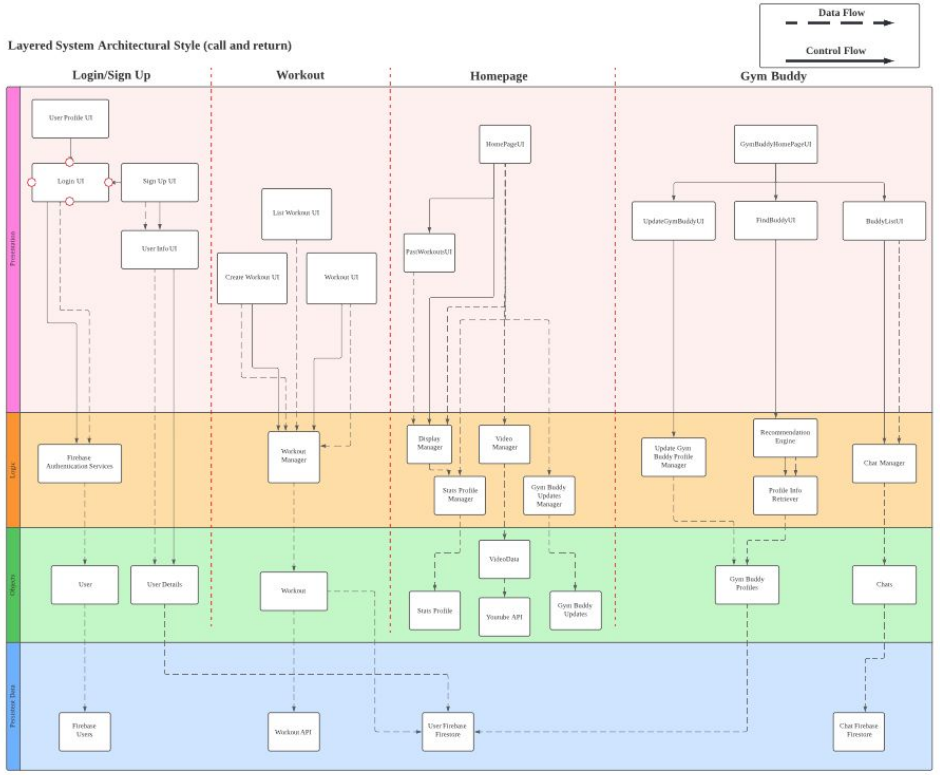
Follows the concepts of clean code and implements SOLID principles

- Allows the designing of applications with very low connection and independent of technical implementation details
- Improves the testability of the application



Clean Architecture

Software Architecture Design



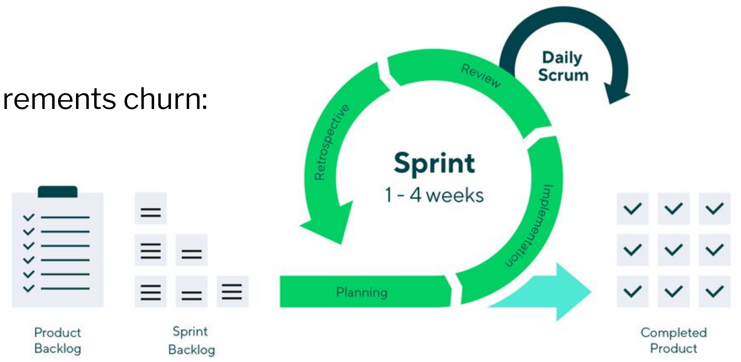
Entire applications boils down to 4 layers and 4 components

- Clear segregation of layers and components allow for greater degree of parallel development
- Ease of delegation of responsibility
- Minimise dependency and improves cohesiveness

Agile Model - Scrum

Why scrum?

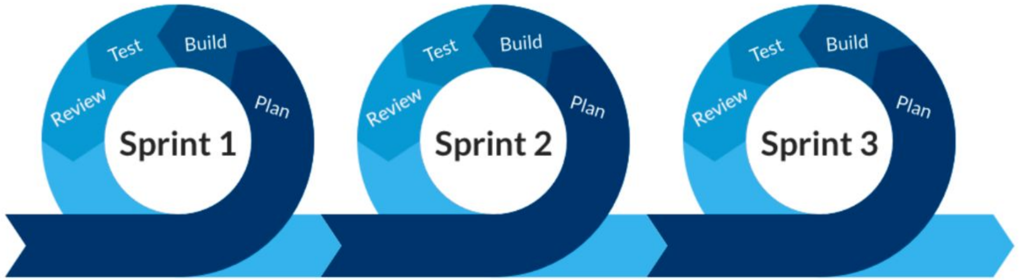
- We recognise that customers can change their minds about what they want and need
- Unpredictability
- Initially: Simple features based on customer requirements
- As time develops: More sophisticated features based on requirements churn:
  - Matchmaking Algorithms
  - Recommendation Algorithms
  - Chat System



Agile Model - Scrum

Scrum in practice!

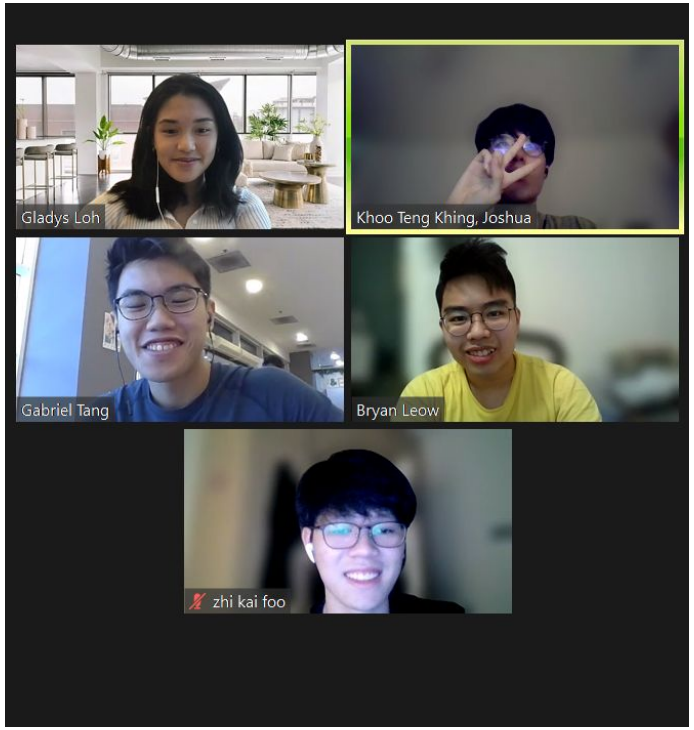
- Our team split the development process into 3 major sprints
- Each sprint lasted 2 weeks
- Each sprint cycle:
  - Planning
  - Building
  - Testing
  - Review
- End of each sprint: Working product with new features - submitted software to customer for constant feedback and improvement



Agile Model - Scrum

Scrum in practice:

- Daily standups to resolve blocking issues
- Roleplaying of different stakeholders
    - 1 Scrum Master - maintains processes
    - 1 Product owner - stakeholders and the business
    - 2 Team members - actual implementation etc
    - 1 Customer - specify requirements for each increment
- Update each team members progress



Joel

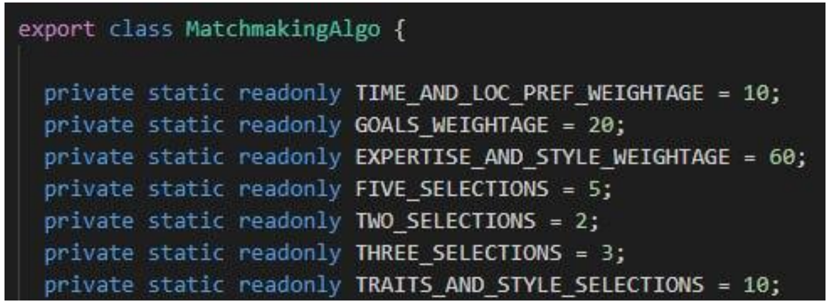
PRACTICE KISSing

"keep it simple, stupid” (KISS). Do not try to write ‘clever’ code.

Software maintenance, reusability and readability

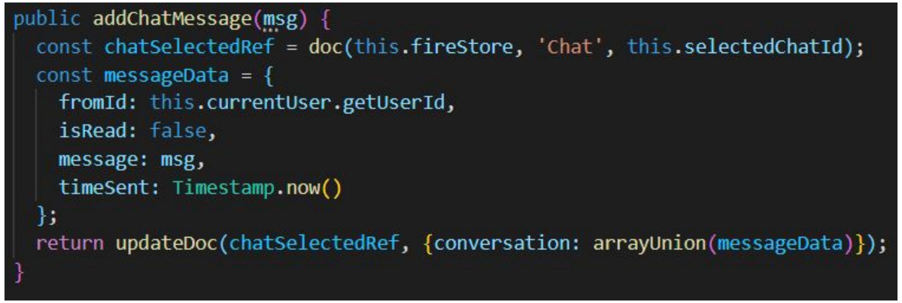
PRACTICE KISSing

- Make the happy path prominent
- Avoid magic numbers
- Follow naming conventions
- Use name to explain
- Javadoc explain why and what not how
- Refactoring iteratively



```
export class MatchmakingAlgo {

  private static readonly TIME_AND_LOC_PREF_WEIGHTAGE = 10;
  private static readonly GOALS_WEIGHTAGE = 20;
  private static readonly EXPERTISE_AND_STYLE_WEIGHTAGE = 60;
  private static readonly FIVE_SELECTIONS = 5;
  private static readonly TWO_SELECTIONS = 2;
  private static readonly THREE_SELECTIONS = 3;
  private static readonly TRAITS_AND_STYLE_SELECTIONS = 10;
```

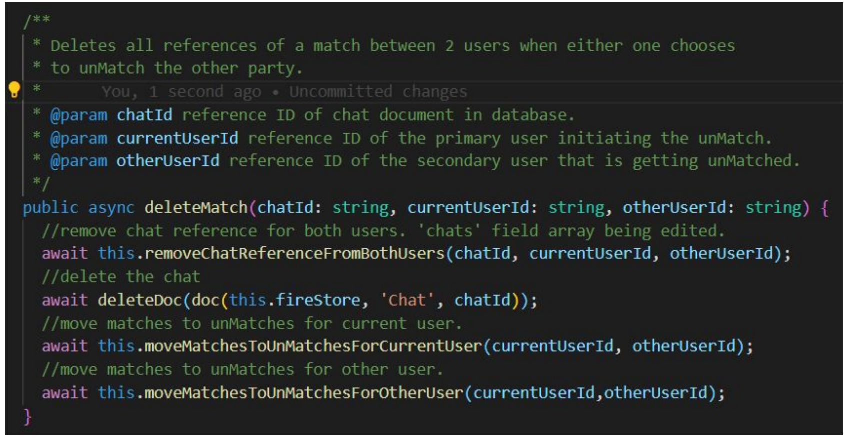


```
public addChatMessage(msg) {
  const chatSelectedRef = doc(this.fireStore, 'Chat', this.selectedChatId);
  const messageData = {
    fromId: this.currentUser.getUserId,
    isRead: false,
    message: msg,
    timeSent: Timestamp.now()
  };
  return updateDoc(chatSelectedRef, {conversation: arrayUnion(messageData)});
}
```

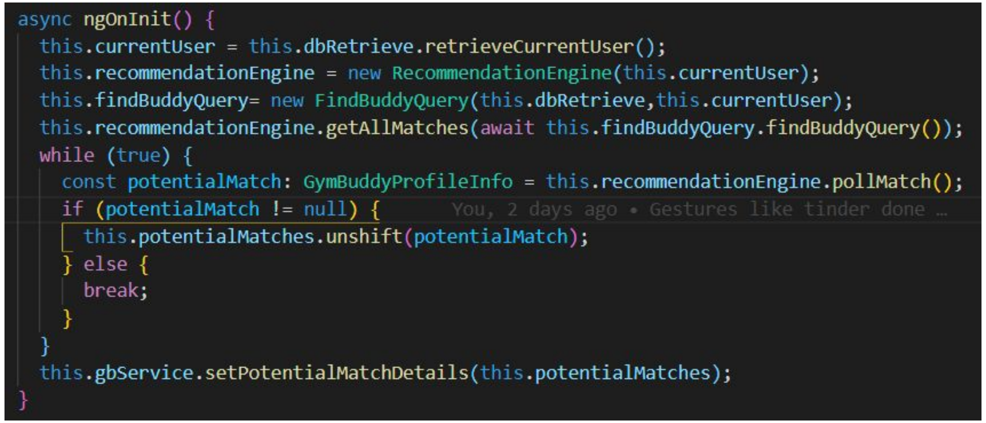
Joel

SLAP hard

Single Level of Abstraction Principle (SLAP) - using **same level of abstraction** for each code fragment



```
/**
 * Deletes all references of a match between 2 users when either one chooses
 * to unMatch the other party.
 *        You, 1 second ago • Uncommitted changes
 * @param chatId reference ID of chat document in database.
 * @param currentUserId reference ID of the primary user initiating the unMatch.
 * @param otherUserId reference ID of the secondary user that is getting unMatched.
 */
public async deleteMatch(chatId: string, currentUserId: string, otherUserId: string) {
  //remove chat reference for both users. 'chats' field array being edited.
  await this.removeChatReferenceFromBothUsers(chatId, currentUserId, otherUserId);
  //delete the chat
  await deleteDoc(doc(this.fireStore, 'Chat', chatId));
  //move matches to unMatches for current user.
  await this.moveMatchesToUnMatchesForCurrentUser(currentUserId, otherUserId);
  //move matches to unMatches for other user.
  await this.moveMatchesToUnMatchesForOtherUser(currentUserId,otherUserId);
}
```



```
async ngOnInit() {
  this.currentUser = this.dbRetrieve.retrieveCurrentUser();
  this.recommendationEngine = new RecommendationEngine(this.currentUser);
  this.findBuddyQuery= new FindBuddyQuery(this.dbRetrieve,this.currentUser);
  this.recommendationEngine.getAllMatches(await this.findBuddyQuery.findBuddyQuery());
  while (true) {
    const potentialMatch: GymBuddyProfileInfo = this.recommendationEngine.pollMatch();
    if (potentialMatch != null) {      You, 2 days ago • Gestures like tinder done …
      this.potentialMatches.unshift(potentialMatch);
    } else {
      break;
    }
  }
  this.gbService.setPotentialMatchDetails(this.potentialMatches);
}
```

User Journey

01

LOGIN

Get your first glimpse of what **WITFIT** is all about

Simply enter your personal details and a customised workout will be created **just for you**

CREATE WORKOUT

02

03

FIND A BUDDY

Connect with people with similar interests and passions

Conduct your workouts with your buddy and **track your progress**

CONDUCT WORKOUT

04

05

LIVE A HEALTHY LIFESTYLE

User Journey

01

LOGIN

Get your first glimpse of what **WITFIT** is all about

02

CREATE WORKOUT

Simply enter your personal details and a customised workout will be created **just for you**

03

FIND A BUDDY

Connect with people with similar interests and passions

04

CONDUCT WORKOUT

Conduct your workouts with your buddy and **track your progress**

05

LIVE A HEALTHY LIFESTYLE

Features

Firebase Authentication
Login
Firebase

Firebase Authentication

Provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users

Promote Inclusivity

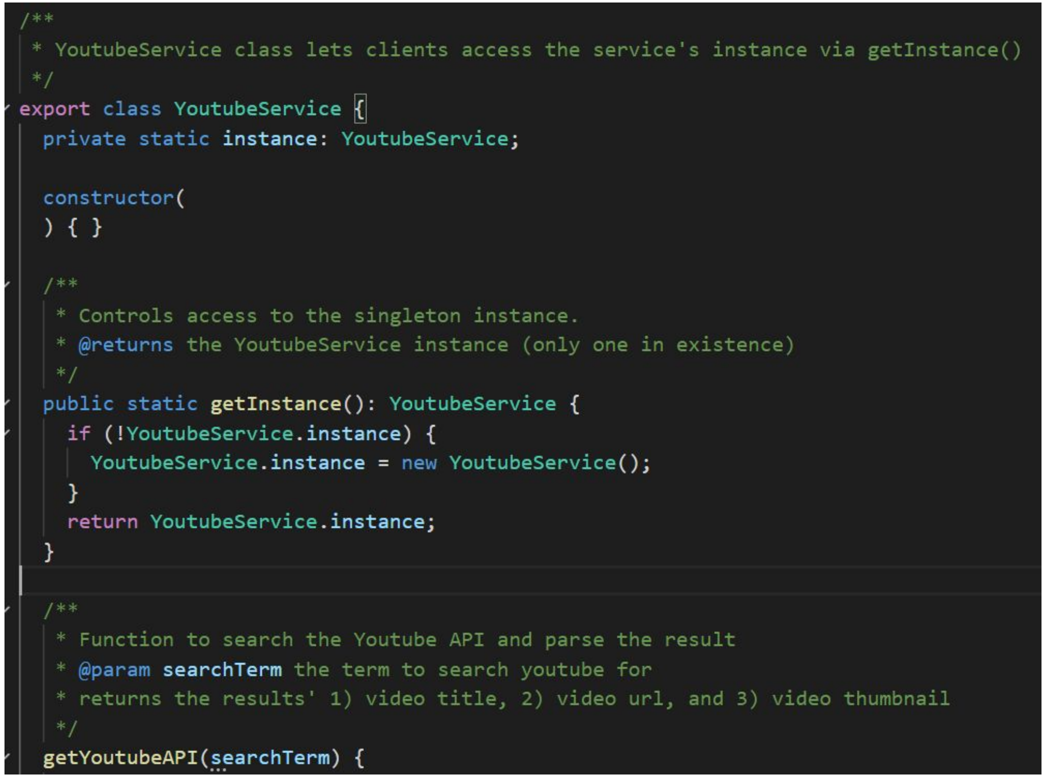
Conscious UI and UX design to cater to all demographics

Joshua

Design Pattern : Singleton

- YoutubeService: Connects to YouTube servers via an API endpoint
- **Problem:** Service will be requested from multiple parts of the system (Homepage, Conduct Workout). Crucial to manage API calls due to rate limits and total usage limits
- **Solution:** Use Singleton to control object creation and define a single entry point to YoutubeService

-



```
/**
 * YoutubeService class lets clients access the service's instance via getInstance()
 */
export class YoutubeService {
  private static instance: YoutubeService;

  constructor(
  ) { }

  /**
   * Controls access to the singleton instance.
   * @returns the YoutubeService instance (only one in existence)
   */
  public static getInstance(): YoutubeService {
    if (!YoutubeService.instance) {
      YoutubeService.instance = new YoutubeService();
    }
    return YoutubeService.instance;
  }

  /**
   * Function to search the Youtube API and parse the result
   * @param searchTerm the term to search youtube for
   * returns the results' 1) video title, 2) video url, and 3) video thumbnail
   */
  getYoutubeAPI(searchTerm) {
```

Joshua

User Journey

01

LOGIN

Get your first glimpse of what **WITFIT** is all about

Simply enter your personal details and a customised workout will be created **just for you**

CREATE WORKOUT

02

03

FIND A BUDDY

Connect with people with similar interests and passions

Conduct your workouts with your buddy and **track your progress**

CONDUCT WORKOUT

04

05

LIVE A HEALTHY LIFESTYLE

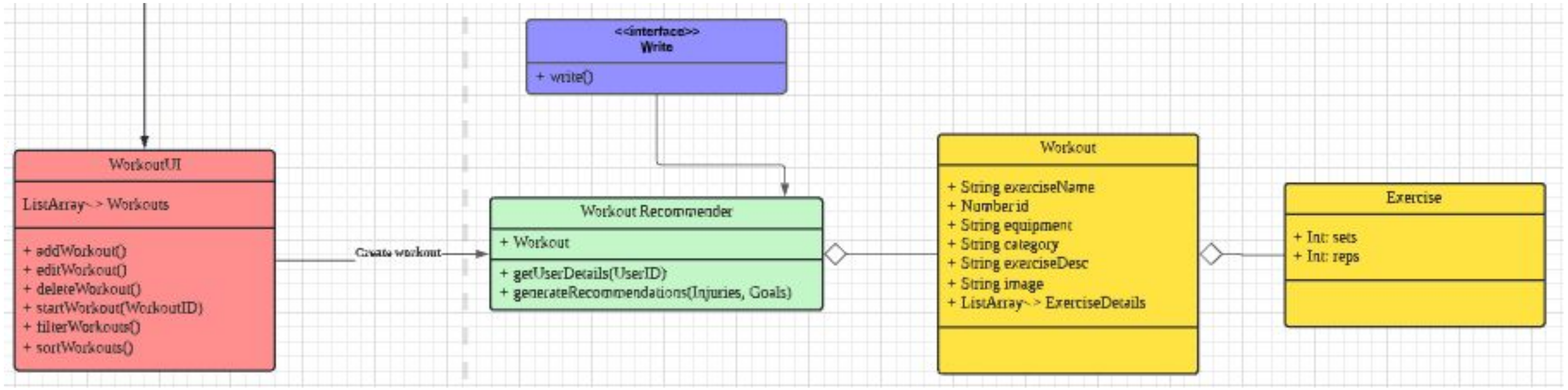
Features

Personalised workout

Enriches user information with scientifically backed knowledge to provide an optimal workout experience

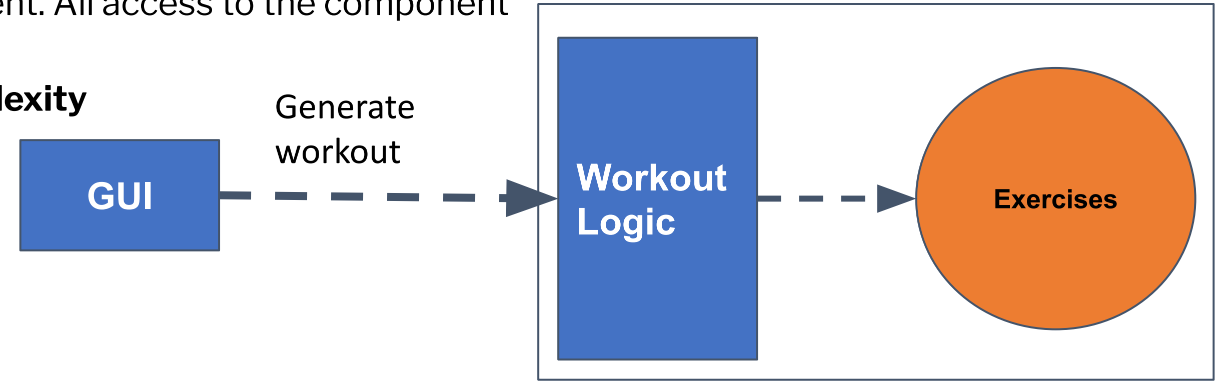
Joshua

Class Diagram

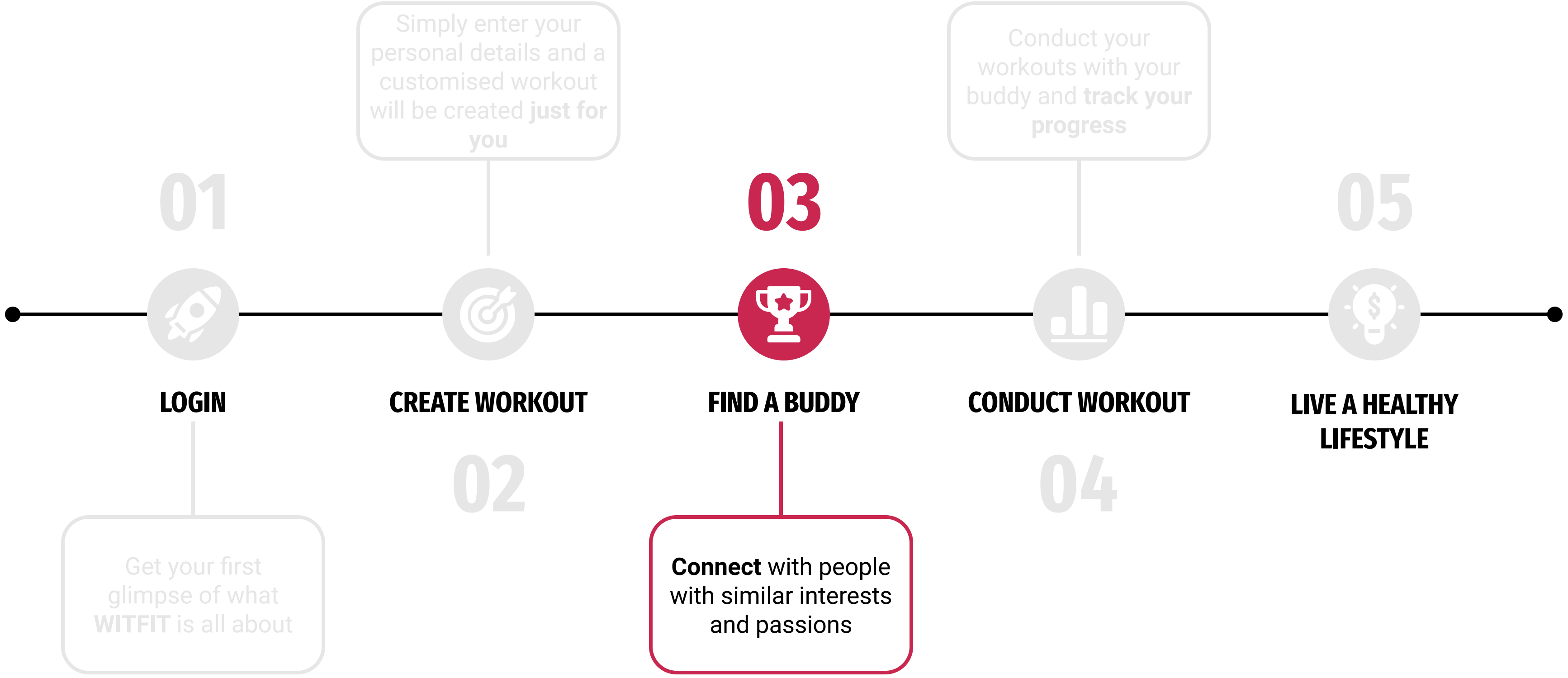


Design Pattern : Facade

- We require the access of exercises based on the user's fitness goal to generate workout catered to their needs
- **Problem:** Access to the component should be allowed without exposing its internal details (UI component should access functionality of logic component without knowing it contains exercise class within it)
- **Solution:** Include a Facade class that sits between component internals and users of the component. All access to the component happens through the facade class
- **Loose coupling + Minimises complexity**



User Journey


01
LOGIN
Get your first glimpse of what **WITFIT** is all about
02
CREATE WORKOUT
Simply enter your personal details and a customised workout will be created **just for you**
03
FIND A BUDDY
Connect with people with similar interests and passions
04
CONDUCT WORKOUT
Conduct your workouts with your buddy and **track your progress**
05
LIVE A HEALTHY LIFESTYLE

Features

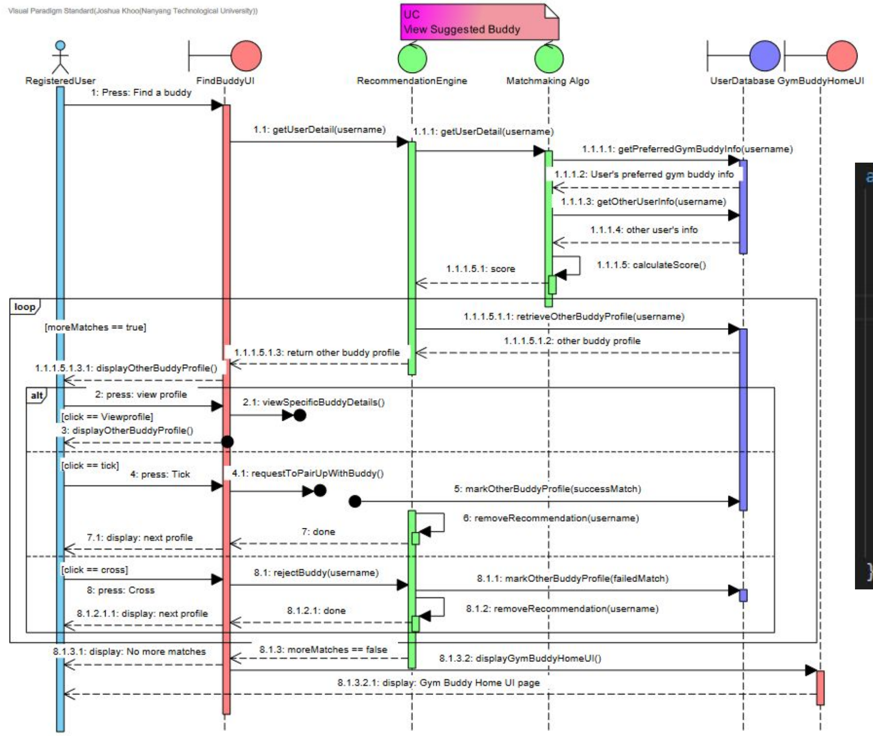
Personalised Matchmaking System

Gathers user information with the aim to provide users with the opportunity to connect with like-minded individuals

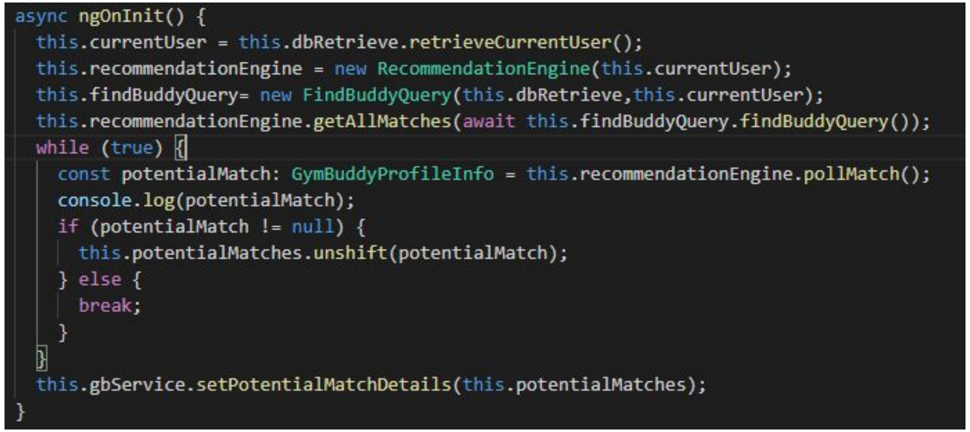
Real Time Chat Application

Provides an optimal in-app experience

Sequence Diagram : Matchmaking Algorithm



Visual Paradigm Standard(Joshua Khoo(Nanyang Technological University))
UC
View Suggested Buddy
RegisteredUser
FindBuddyUI
RecommendationEngine
Matchmaking Algo
UserDatabase
GymBuddyHomeUI
1: Press: Find a buddy
1.1: getUserDetail(username)
1.1.1: getUserDetail(username)
1.1.1.1: getPreferredGymBuddyInfo(username)
1.1.1.2: User's preferred gym buddy info
1.1.1.3: getOtherUserInfo(username)
1.1.1.4: other user's info
1.1.1.5: calculateScore()
1.1.1.5.1: score
loop
[moreMatches == true]
1.1.1.5.1.1: retrieveOtherBuddyProfile(username)
1.1.1.5.1.2: other buddy profile
1.1.1.5.1.3: return other buddy profile
1.1.1.5.1.3.1: displayOtherBuddyProfile()
alt
2: press: view profile
2.1: viewSpecificBuddyDetails()
[click == Viewprofile]
3: displayOtherBuddyProfile()
[click == tick]
4: press: Tick
4.1: requestToPairUpWithBuddy()
5: markOtherBuddyProfile(successMatch)
6: removeRecommendation(username)
7: done
7.1: display: next profile
[click == cross]
8: press: Cross
8.1: rejectBuddy(username)
8.1.1: markOtherBuddyProfile(failedMatch)
8.1.2: removeRecommendation(username)
8.1.2.1: done
8.1.2.1.1: display: next profile
8.1.3: moreMatches == false
8.1.3.1: display: No more matches
8.1.3.2: displayGymBuddyHomeUI()
8.1.3.2.1: display: Gym Buddy Home UI page


```
async ngOnInit() {
  this.currentUser = this.dbRetrieve.retrieveCurrentUser();
  this.recommendationEngine = new RecommendationEngine(this.currentUser);
  this.findBuddyQuery= new FindBuddyQuery(this.dbRetrieve,this.currentUser);
  this.recommendationEngine.getAllMatches(await this.findBuddyQuery.findBuddyQuery());
  while (true) {
    const potentialMatch: GymBuddyProfileInfo = this.recommendationEngine.pollMatch();
    console.log(potentialMatch);
    if (potentialMatch != null) {
      this.potentialMatches.unshift(potentialMatch);
    } else {
      break;
    }
  }
  this.gbService.setPotentialMatchDetails(this.potentialMatches);
}
```

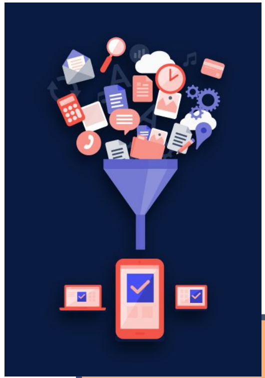
Joshua

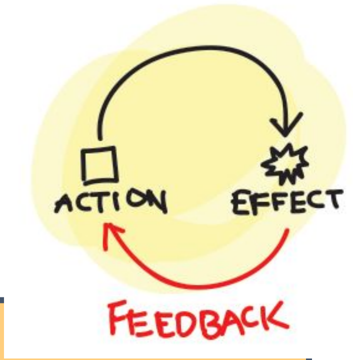
Matchmaking Algorithm

Matchmaking Algorithm

Content Filtering

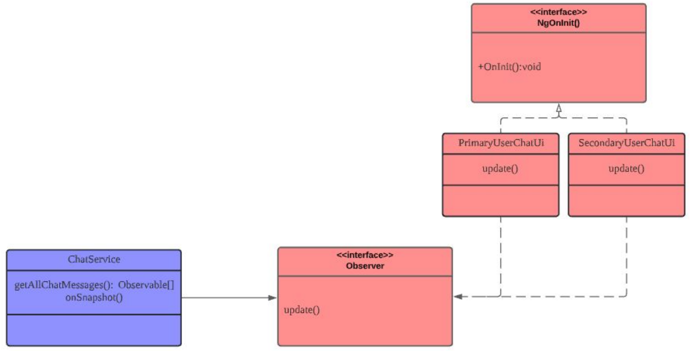
User behaviour and feedback loop

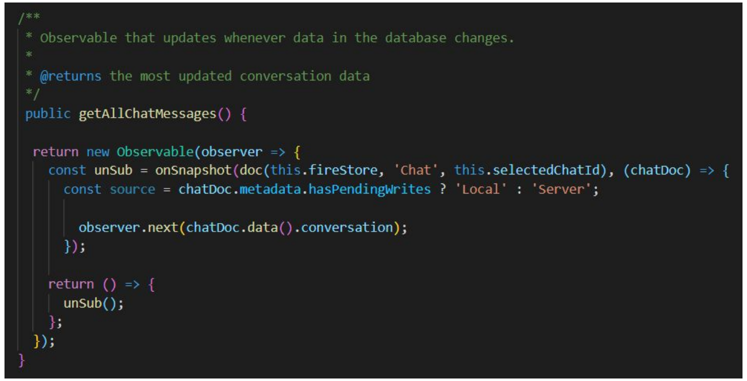




Design Pattern : Observer pattern (Chat)

- We require any updates in chat messages to be in **real-time.**
- **Problem:** The ‘observed’ object does not want to be coupled to objects that are ‘observing’ it.
- **Solution:** Force the communication through an interface known to both parties.
- Speed up update: The subscriber gets pinged right before there is a system write to storage -> **faster response times.**





```
/**
 * Observable that updates whenever data in the database changes.
 *
 * @returns the most updated conversation data
 */
public getAllChatMessages() {

  return new Observable(observer => {
    const unSub = onSnapshot(doc(this.fireStore, 'Chat', this.selectedChatId), (chatDoc) => {
      const source = chatDoc.metadata.hasPendingWrites ? 'Local' : 'Server';

        observer.next(chatDoc.data().conversation);
      });

    return () => {
      unSub();
    };
  });
}
```

Joshua

User Journey

01

LOGIN

Get your first glimpse of what **WITFIT** is all about

02

CREATE WORKOUT

Simply enter your personal details and a customised workout will be created **just for you**

03

FIND A BUDDY

Connect with people with similar interests and passions

04

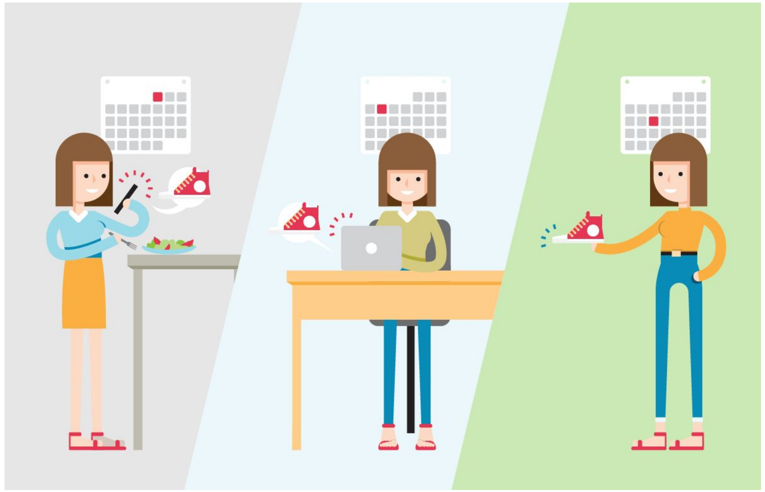
CONDUCT WORKOUT

Conduct your workouts with your buddy and **track your progress**

05

LIVE A HEALTHY LIFESTYLE

Features



Seamless Experience

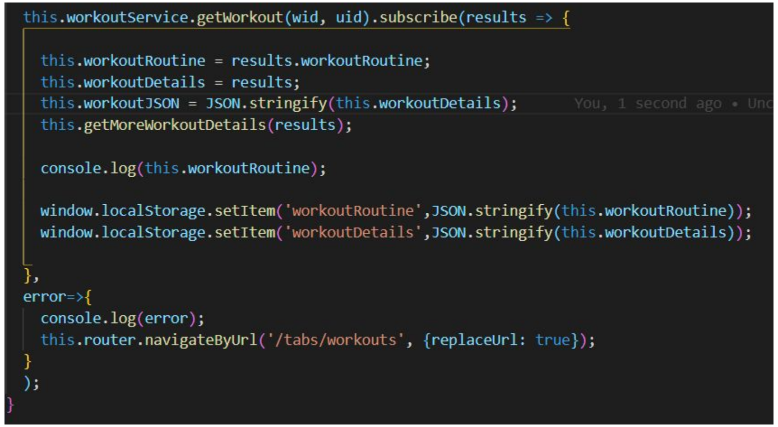
Enables users to easily track their progress throughout the workout session

Prepare for the future

Comprehensive dashboard and personalized videos prepares users for their next workout

Joshua

Design Pattern : Observer pattern (Workouts)



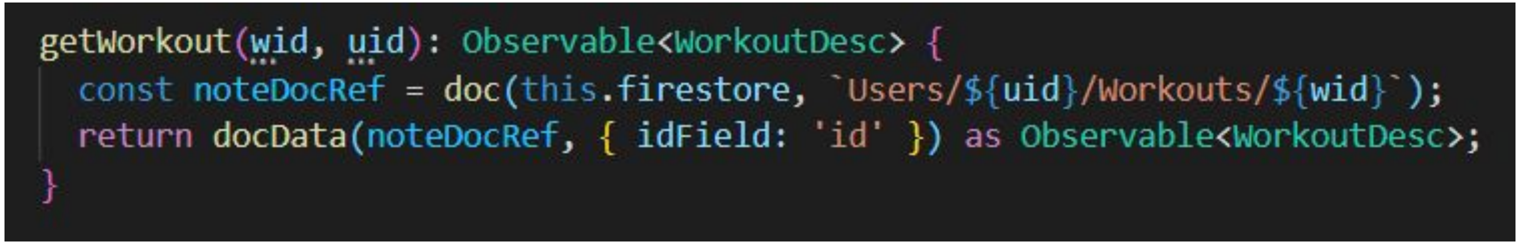
```
this.workoutService.getWorkout(wid, uid).subscribe(results => {

  this.workoutRoutine = results.workoutRoutine;
  this.workoutDetails = results;
  this.workoutJSON = JSON.stringify(this.workoutDetails);      You, 1 second ago • Unc
  this.getMoreWorkoutDetails(results);

  console.log(this.workoutRoutine);

  window.localStorage.setItem('workoutRoutine',JSON.stringify(this.workoutRoutine));
  window.localStorage.setItem('workoutDetails',JSON.stringify(this.workoutDetails));

},
error=>{
  console.log(error);
  this.router.navigateByUrl('/tabs/workouts', {replaceUrl: true});
}
);
}
```



```
getWorkout(wid, uid): Observable<WorkoutDesc> {
  const noteDocRef = doc(this.firestore, `Users/${uid}/Workouts/${wid}`);
  return docData(noteDocRef, { idField: 'id' }) as Observable<WorkoutDesc>;
}
```

Joshua

User Journey

01

LOGIN

Get your first glimpse of what **WITFIT** is all about

02

CREATE WORKOUT

Simply enter your personal details and a customised workout will be created **just for you**

03

FIND A BUDDY

Connect with people with similar interests and passions

04

CONDUCT WORKOUT

Conduct your workouts with your buddy and **track your progress**

05

LIVE A HEALTHY LIFESTYLE

User Journey

Simply enter your personal details and a customised workout will be created **just for you**

Conduct your workouts with your buddy and **track your progress**

WITFIT aims to provide users with the **right tools and resources** to live a healthy lifestyle

LIFESTYLE

02

04

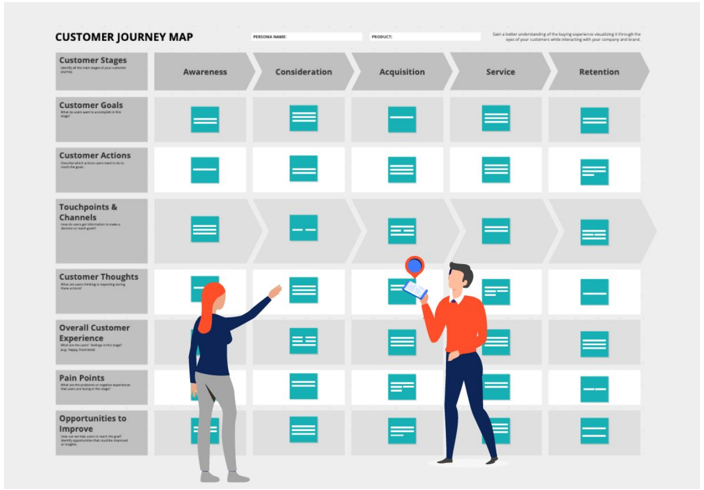
Get your first glimpse of what **WITFIT** is all about

Connect with people with similar interests and passions

Future Improvements

Improve machine learning algorithms

Our group aims to gather more data and create an even more personalised experience for our users



Further research into user journey and experience

Our group believes that having a seamless user experience is key to user retention and the growth of the application

WITFIT

Working in Tandem, For Fitness

Thank You