

1. Home	2
1.1 Project Details	6
1.2 Project Introduction	9
1.2.1 Technical details about the project	10
1.2.2 Sprint process introduction	11
1.2.3 Project Development Environment Setting Guide	13
1.2.4 Project Product Demo	31
1.3 Requirements	33
1.3.1 Requirements 1(Code cleaning)	34
1.3.2 Requirements 2(Initialize frontend & Backend)	35
1.3.3 Requirements 3(Solve previous project left bugs)	36
1.3.4 Requirements 4Solve previous project left security problems	37
1.3.5 Requirements 5 (Test in sprint 2)	38
1.3.6 Requirements 6 (Looking for possible security issues)	39
1.3.7 Requirements 7(Ensure accessibility)	40
1.3.8 Requirements 8(Disconnect Garmin Connect)	41
1.3.9 Requirements 9(Change Garmin connect account)	42
1.3.10 Requirements 10(Reset password function)	43
1.3.11 Requirements 11(Cloud front-end deployment)	44
1.3.12 Requirements 12(Fix transferring_issue)	45
1.3.13 Requirements 13(Add SMTP email server to the backend)	46
1.3.14 Requirements 14(Integrationw with SpringSecurity)	47
1.4 Project Planning	48
1.4.1 Sprint 1 Plan	49
1.4.2 Sprint 2 Plan	50
1.4.3 Sprint 3 Plan	51
1.4.4 Sprint 4 Plan	52
1.4.5 Trello redirect page	53
1.4.6 User Story	55
1.4.6.1 Sprint 1 related User Story	56
1.4.6.2 Sprint 2 related User Story	57
1.4.6.3 Sprint 3 related User Story	62
1.5 Development	64
1.5.1 Progress specification	65
1.5.1.1 Code cleaning	67
1.5.1.2 Check Availability for Username and Email	70
1.5.1.3 Receiving data from Garmin connect	73
1.5.1.4 Add password constraints in frontend	74
1.5.1.5 Encrypt password	75
1.5.1.6 Fix the map in frontend	77
1.5.1.7 Sprint 2 Fix API Test	80
1.5.1.8 Fix transferring security issue	81
1.5.1.9 Cloud front-end deployment	83
1.5.1.10 Reset password function	87
1.5.1.11 Change Garmin connect account	89
1.5.1.12 Disconnect Garmin Connect	90
1.5.1.13 Add SMTP email server to the backend	92
1.5.1.14 Integrate with Spring Security	95
1.5.2 Project Github Repositories	98
1.5.2.1 Sprint 2 Branch specification	100
1.5.3 Peer-to-peer code review	101
1.5.4 ChatGPT code review	104
1.6 Testing	107
1.6.1 Test	108
1.6.1.1 Test ActivityDaolmpl	109
1.6.1.2 Test ActivityDataRetrieveController	114
1.6.1.3 Test DashboardStatisticsDao	119
1.6.1.4 Test generateRequestTokenSecret	122
1.6.1.5 Test ActivityServiceImpl	123
1.6.1.6 Test UserServiceImpl	126
1.6.1.7 Test LoginController	127
1.6.1.8 Test of Code Cleaning	128
1.6.1.9 Test of Login/Register page	130
1.6.1.10 Test RegistryController	132
1.6.1.11 Test TokenDao	133
1.6.1.12 Test UserDao	134
1.6.1.13 Test of Cloud front-end deployment	136
1.6.1.14 Test Reset Password Function	138
1.6.1.15 Test Disconnect and Change Garmin Account	139
1.6.1.16 Test SMTP Service	140
1.6.1.17 Test fixngt ransferring issue	141
1.6.1.18 Test Spring Security Integration	142
1.6.1.19 Test Receiving Data from Garmin	143
1.6.2 Troubleshooting articles	144
1.6.2.1 Former project problems	145
1.7 Project review	152
1.7.1 Sprint2_Review	153
1.7.2 Sprint3_Review	160

Home

Project Overview

GA-BoxJelly is a software project aimed at integrating data from Garmin watches into the CoachingMate system to enhance workout guidance for exercisers. Part of the work in the project has been completed by the former team, such as the visualization of Garmin watch data in CM, etc. The primary goal of the project is mainly to deploy the project on the AWS server and further improve the UI/UX. This will be achieved through secure connections and AWS data storage, as well as user-friendly interfaces for both coaches and exercisers.

Recent space activity



Yukun Li

Receiving data from Garmin connect 4 •



Lingkang Zhou

Sprint3_Review 5 •



Yukun Li

Test Receiving Data from Garmin 5

Add SMTP email server to the backend 12 •



Yuhang YAO

Integrate with Spring Security 13 •

Key Features:

- Secure and data integration: GA-BoxJelly will establish connections between Garmin watches and the CoachingMate system, allowing for data transmission after workout sessions.
- Organized and secure data storage: The collected data will be stored on a secure server database, ensuring privacy and easy access for analysis and guidance by coaches.
- Comprehensive data visualization and analysis: Coaches will have access to a user-friendly dashboard that enables them to analyze exercisers' data, identify trends, and provide personalized workout guidance based on individual needs and goals.
- Customizable data sharing: Exercisers will have control over which data they share with the CoachingMate system, allowing them to maintain their privacy while still benefiting from personalized coaching.
- Personalized feedback and progress tracking: Exercisers will be able to view their workout data within the CoachingMate system and receive tailored feedback from coaches, helping them improve their performance and reach their fitness goals.

By integrating Garmin watch data into the CoachingMate system, GA-BoxJelly will empower coaches to provide more accurate and personalized workout guidance, ultimately leading to better fitness outcomes for exercisers.

Background:

In recent years, the use of fitness wearables, such as Garmin watches, has increased exponentially as more people seek to monitor their workouts and improve their fitness levels. These devices collect valuable data, such as heart rate, distance traveled, and calories burned, which can provide insights into an individual's workout performance and progress. However, leveraging this data to provide personalized workout guidance has remained a challenge, especially for fitness coaches who work with multiple clients. Because these data were all stored locally which cannot be shared with coach of CoachingMate originally.

CoachingMate, a fitness coaching platform, aims to address this challenge by integrating data from Garmin watches into its system. This will enable coaches to access and analyze their clients' workout data and provide tailored feedback and guidance based on individual needs, ultimately helping exercisers achieve better results.

Goals:

Current goal would be discover potential improvements to the UI/UX and code tidying and code optimization for backend as much as possible the then implement improvements to the CM application.

Scope:

The scope of the project is to analyse the exiting UX/UI, come up with a prioritised list of bugs and UX improvements, and implement the improvements into the AWS instance of CM.

Space contributors

- [Yukun Li](#) (4)
- [Lingkang Zhou](#) (5)
- [Yuhang YAO](#) (13)
- [Rui Liu](#) (1)
- [Xiuyuan Zhu](#) (24)
- ...

TODO: Home Page Design

- project specification 2023-3-16
- rate of process 2023-3-16
- page map 2023-3-16
- ... [Yukun Li](#) [Lingkang Zhou](#) @.....

Stakeholders

xxx	Dr Eduardo Araujo Oliveira	Paul.calverley
-----	----------------------------	----------------

img		
Client	Subject Coordinator	Project Supervisor
email	eduardo.oliveira@unimelb.edu.au	paul.calverley@unimelb.edu.au

Project Team

Lingkang Zhou	Rui Liu	Xiuyuan Zhu
		
Product owner	Development	Scrum Master
lingkangz@student.unimelb.edu.au	rlli2@student.unimelb.edu.au	xiuzhu@student.unimelb.edu.au

Yuhang YAO	Yukun Li
------------	----------



Development

Development

yuhao1@student.unimelb.edu.au

yukun2@student.unimelb.edu.au

Project Details

Project Overview:

GA-BoxJelly

GA-BoxJelly is a software project aimed at integrating data from Garmin watches into the CoachingMate system to enhance workout guidance for exercisers. Part of the work in the project has been completed by the former team, such as the visualization of Garmin watch data in CM, etc. The primary goal of the project is mainly to deploy the project on the AWS server and further improve the security. This will be achieved through secure connections and AWS data storage, as well as user-friendly interfaces for both coaches and exercisers.

Project Goal:

- Fix the entire project program
- Ensure the stability of the system
- Improve system security
- Ensure the sustainable maintenance of the system.

By achieving these goals, GA-BoxJelly-CoachingMate project will make the coachingmate system more complete and safe, harnessing the power of Garmin watch data to drive better fitness outcomes.

Key Features:

- Secure and real-time data integration: GA-BoxJelly will establish secure connections between Garmin watches and the CoachingMate system, allowing for real-time data transmission during workout sessions.
- Organized and secure data storage: The collected data will be stored on a secure server, ensuring privacy and easy access for analysis and guidance by coaches.
- Comprehensive data visualization and analysis: Coaches will have access to a user-friendly dashboard that enables them to analyze exercisers' data, identify trends, and provide personalized workout guidance based on individual needs and goals.
- Customizable data sharing: Exercisers will have control over which data they share with the CoachingMate system, allowing them to maintain their privacy while still benefiting from personalized coaching.
- Personalized feedback and progress tracking: Exercisers will be able to view their workout data within the CoachingMate system and receive tailored feedback from coaches, helping them improve their performance and reach their fitness goals.

By integrating Garmin watch data into the CoachingMate system, GA-BoxJelly will empower coaches to provide more accurate and personalized workout guidance, ultimately leading to better fitness outcomes for exercisers.

Background:

In recent years, the use of fitness wearables, such as Garmin watches, has increased exponentially as more people seek to monitor their workouts and improve their fitness levels. These devices collect valuable data, such as heart rate, distance traveled, and calories burned, which can provide insights into an individual's workout performance and progress. However, leveraging this data to provide personalized workout guidance has remained a challenge, especially for fitness coaches who work with multiple clients. Because these data were all stored locally which cannot be shared with coach of CoachingMate originally.

CoachingMate, a fitness coaching platform, aims to address this challenge by integrating data from Garmin watches into its system. This will enable coaches to access and analyze their clients' workout data and provide tailored feedback and guidance based on individual needs, ultimately helping exercisers achieve better results.

Scope

The final product of this project will be a **well-designed webpage** that displays the workout data for each activity and athletes, and let coaches monitor their athletes' performance.

The key criteria of this project is for users to get familiar with their progress in the training program and help them analyse and improve their performance.

In-scope :

- Key functions to be achieved including:
- Let coaches monitor their athletes' performance
- More features of analyzing data.
- Let users set permissions for what data can be shared for coaches to see

Out-scope :

As the data acquisition and front-end dashboard were realized by former group last year, the connection of Garmin API and build up of the front-end will not be involved in this project.

In scope:

Epic ID	Epic	UserStory ID	User	Story	Moscow Priority	Acceptance criteria
1	Integration of data from Garmin watches to a server	1a	CoachingMate owner	I want to securely connect to exerciser's Garmin watches, so that I can receive their workout data in real-time.	must	The user can securely connect to exerciser's Garmin watches.
		2a	exerciser	I want to easily connect my watch to the CoachingMate system, so that my workout data is available for my coach to review.	must	The data is synced to the coaching mate system
		3a	CoachingMate owner	I want to store the collected data from Garmin watches in an organized and secure manner, so that I can access it for analysis and guidance.	should	The data is well-displayed in the dashboard so that the user can drag or analysis data easily
		4a	exerciser	I want to have control over which data is shared with the CoachingMate system, so that I can maintain my privacy.	must	The user can control which data is allowed to be shared to others.
2	Display of collected data for better workout guidance	1b	CoachingMate owner	I want to easily visualize and analyze the collected data from exercisers, so that I can provide personalized workout guidance.	should	The user can visualize and analyze the collected data from exercisers.
		2b	exerciser	I want to view my workout data in the CoachingMate system, so that I can track my progress and achievements.	must	The user can view the workout data in the CoachingMate system.
		3b	CoachingMate owner	I want to set up alerts and notifications based on exerciser's data, so that I can provide timely advice and support during their workout sessions.	should	Exerciser data can be configured for alerts and notifications.
		4b	exerciser	I want to receive personalized feedback from my coach based on my workout data, so that I can improve my performance and reach my fitness goals.	should	The user can receive personalized feedback from the coach based on the workout data.

Project Introduction

Technical details about the project

Project Management Tools

- [Confluence Space](#)
- [Trello](#)
- Slack
- Github

Development Environment

- Frontend:
 - node: v14.17.6
 - vue-cli: 2.9.6
 - react: 16.9.34
- Backend: Java 8
- Database: MongoDB
- Framework: Frontend-backend structure

Development tools

- IDE: VScode
- Program: Linux, MacOS, Windows
- Brower: Chrome

Sprint process introduction

Story points in Trello

Example

The screenshot shows a Trello card for a user story titled "Send an email server to the backend". The card is in the "In Progress" column. On the right side of the card, a modal window is open for adding power-ups. The "Story Points" field is highlighted with a red circle. Below it, there is a section for adding a story point to the card, with a "Story Points" input field containing the value "17". A row of numbered buttons (1, 2, 3, 5, 7, 11, 13, 17, 19) is shown, with "17" being the selected value. A "REMOVE" button is also visible.

Step 1: Identifying Tasks

- We break down each user story into tasks. We ensure that each task is small enough to be achievable within a single sprint.
- We ensure that every member of our team understands these tasks.

Step 2: Discussing Complexity and Effort

- As a team, we discuss the complexity and the effort required for each task. This collaborative effort ensures a common understanding across our team.
- We use a known task as a baseline for comparing other tasks.

Step 3: Assigning Story Points

- We use a numerical scale like the Fibonacci sequence (1, 2, 3, 5, 8, 13, and so on) to assign story points to each task. The higher the number, the more complex the task.
- We typically do this in a "planning poker session" during the Sprint planning meeting, where each team member makes an estimate and then all reveal their estimates at the same time to avoid influencing each other.

Project Development Environment Setting Guide

This page would be a Getting Started Guide for BoxJelly project.

Getting Started(HandOver)

Welcome to the BoxJelly project. As a user who wants to deploy and use this project, it is important to understand that the project consists of two parts: the front-end web page, which displays data charts, and the back-end, which communicates with the Garmin Connect platform and processes data. The back-end also includes a basic database program used for storing data. To properly deploy this project and make it fully functional, the front-end, back-end, and database must all be completed in sequence according to the steps in this guide.

1. DataBase:

requiredDebian10 instance (Recommended service provider: Vultr)

1. Import MongoDB's public GPG key:

First, import MongoDB's public GPG key to ensure the integrity and authenticity of the downloaded packages. Run the following command:

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```

2. Add the MongoDB repository:

Create a list file containing the MongoDB repository. For MongoDB 4.4, run the following command:

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/debian buster/mongodb-org/4.4 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
```

Then, update the local package database:

```
sudo apt update
```

3. Install MongoDB using the following command:

```
sudo apt install -y mongodb-org
```

4. Start the MongoDB service:

After the installation is complete, start the MongoDB service and set it to start automatically at system boot:

```
sudo systemctl start mongod
sudo systemctl enable mongod
```

5. Verify the MongoDB installation:

Check the status of the MongoDB service using the following command:

```
sudo systemctl status mongod
```

If you see Active: active (running) status, it means MongoDB has been successfully installed and is running.

Now, you have successfully deployed MongoDB on a Vultr Debian 10 instance. To ensure system security, make sure to properly configure MongoDB, manage permissions, and set up the firewall when deploying it in a real-world scenario. For more information, refer to the MongoDB official documentation (<https://docs.mongodb.com/manual/administration/security-checklist/>).

2. BackEnd:

[Change the database of your own \(ignore this if you want to use the old database from the previous year\)](#)

GA-RedBack-main - application-dev.properties

```

#oauthAccessToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/access_token
#callBackURL.url=https://coaching-mate0121.herokuapp.com/auth/accessToken

requestToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/request_token
oauthConfirm.url=http://connect.garmin.com/oauthConfirm
oauthAccessToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/access_token
callBackURL.url=https://ga-boxjelly.herokuapp.com/auth/accessToken
#callBackURL.url=https://localhost:8080/auth/accessToken

#redis configuration
#spring.redis.database=0
#spring.redis.host=8.129.65.2
#spring.redis.port=6379
#spring.redis.password=717294
#spring.redis.timeout=3000ms

spring.data.mongodb.uri = mongodb://coachingmate:123456@127.0.0.1:27017/coachingmate

#mongodb for local configuration
#spring.data.mongodb.database=coachingmate
#spring.data.mongodb.port=27017

#spring.data.mongodb.uri = mongodb+srv://ga_boxjelly:Ga_boxjelly2022@cluster1.mjmt2.mongodb.net/coachingmate?retryWrites=true&ssl=true&replicaSet=atlas-1w6y6m

```

Terminal: Local +

```

on [connectionId{localValue:1, serverValue:17}] to 127.0.0.1:27017
2023-03-18 12:04:53.346 INFO [cluster-ClusterId{value='64150e3542f5412b86e90070', description='null'}-127.0.0.1:27017] org.mongodb.driver.cluster.info:71 -Monitor thread successfully connected to server with description ServerDescription{address=127.0.0.1:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=1690084}
2023-03-18 12:04:53.865 INFO [main] o.s.scheduling.concurrent.ThreadPoolTaskExecutor.initialize:181 -Initializing ExecutorService

```

Version Control Run TODO Problems Terminal Services Profiler Dependencies Build

Unused property

Note: The old database already has a lot of sample data inside, it can be harder to find the new data added.

GA-RedBack-main - application-dev.properties

```

#server.port=8080
server.port=${port:8080}

## the type of the key store (JKS or PKCS12)
#server.ssl.key-store-type=pkcs12

## the path to the key store that contains the SSL certificate.
#server.ssl.key-store=classpath:cert2.p12
## the password used to access the key store
#server.ssl.key-store-password=password
## the password used to access the key in the key store
#server.ssl.key-password=password

spring.swagger2.enabled=true

spring.banner.location=banner.txt
spring.devtools.restart.trigger-file=trigger.txt
spring.thymeleaf.cache=false

#Consumer Credentials
#oauth.consumerkey = d258c644-e0d8-4177-9d0c-b5b4478a2c95
#oauth.consumerSecret = 2qzSIbfivQmYXA2Uz0837cWE7HLxwoSB

```

Terminal: Local +

```

on [connectionId{localValue:1, serverValue:17}] to 127.0.0.1:27017
2023-03-18 12:04:53.346 INFO [cluster-ClusterId{value='64150e3542f5412b86e90070', description='null'}-127.0.0.1:27017] org.mongodb.driver.cluster.info:71 -Monitor thread successfully connected to server with description ServerDescription{address=127.0.0.1:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=1690084}
2023-03-18 12:04:53.865 INFO [main] o.s.scheduling.concurrent.ThreadPoolTaskExecutor.initialize:181 -Initializing ExecutorService

```

Version Control Run TODO Problems Terminal Services Profiler Dependencies Build

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (yesterday 9:44 pm) 2:1 (24 chars) LF ISO-8859-1 4 spaces

If the default port 8080 is occupied locally, it needs to be modified.

GA-RedBack-main – application-dev.properties

```

#server.port=8080
server.port=${port:8080}
#
## the type of the key store (JKS or PKCS12)
#server.ssl.key-store-type=pkcs12
#
## the path to the key store that contains the SSL certificate.
#server.ssl.key-store=classpath:cert2.p12
#
## the password used to access the key store
#server.ssl.key-store-password=password
#
##the password used to access the key in the key store
#server.ssl.key-password=password
#
spring.swagger2.enabled=true
spring.banner.location=banner.txt
spring.devtools.restart.trigger-file=trigger.txt
spring.thymeleaf.cache=false
#Consumer Credentials
#oauth.consumerKey = d258c644-e0d8-4177-9d0c-b5b4478a2c95
#oauth.consumerSecret = 2qzSIibfjvQmY0XA2Uz0837cWE7HLxwoSB

```

Terminal: Local +

```

on [connectionId{localValue:1, serverValue:17}] to 127.0.0.1:27017
2023-03-18 12:04:53.346 INFO [cluster-ClusterId{value='64150e3542f5412b86e90070', description='null'}-127.0.0.1:27017] org.mongodb.driver.cluster.info:71 -Monitor thread successfully connected to server with description ServerDescription{address=127.0.0.1:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=1690084}
2023-03-18 12:04:53.865 INFO [main] o.s.scheduling.concurrent.ThreadPoolTaskExecutor.initialize:181 -Initializing ExecutorService

```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (yesterday 9:44 pm) 2:1 (24 chars) LF ISO-8859-1 4 spaces

Change the callback URL of your own

Data samples backend/backenddata/regisgtry-controller To enter the website here, you need to change the push URL.

cm-backend – GarminPushController.java

```

public ResponseEntity<String> acceptPushedActivity(@RequestBody String activities) {
    logger.info("start push data");
    //请求参数打印
    logger.info("Activities: " + activities);
    Gson gson = new GsonBuilder().create();
    //ActivityBean activityBean = gson.fromJson(activities, ActivityBean.class);
    Activity[] test=gson.fromJson(activities, Activity[].class);
    for(Activity activity : test){
        if(activity != null){
            activityService.insertActivity(activity);
        }
    }
}

//logger.info("properties: " + properties);
//logger.info("upload :" + activityList);
activityService.insertActivity(activityList);
activityService.statisticsActivities(accesToken);
HttpHeaders httpHeaders = new HttpHeaders();
httpHeaders.set("Location", "public/garmin_raw");
return ResponseEntity.accepted().headers(httpHeaders).body("Accept the pushed activities");
}

//configure this url to endpoint configuration, and the garmin endpoint will transfer the data to this server
@PostMapping("/pushActivity", produces = "application/json")
@RequestMapping(value = "/pushActivityDetail", method = RequestMethod.POST, produces = "application/json")
@ApiOperation(value = "push data url", notes = "configure this url to endpoint configuration, ")

```

Run: CoachingmateApplication

2023-03-19 12:53:57.744 INFO [main] o.s.w.s.m.m.MockMvc.perform:111 - 正在共享屏幕

Build completed successfully in 987 ms (3 minutes ago)

OnSever

requiredDebian10 instance (Recommended service provider: Vultr)

1. Update the system:

First, open a terminal on your instance and run the following commands to update all packages:

```
sudo apt update  
sudo apt upgrade -y
```

2. To run the jar file, you need Java installed on your instance. It is recommended to use OpenJDK. To install Java 8, run the following command:

```
sudo apt install openjdk-11-jdk -y
```

After the installation is complete, verify that Java was successfully installed:

```
java -version
```

3. Download the jar file:

First, make sure 'wget' tools are installed:

```
sudo apt install wget -y
```

Then, use one of the following commands to download the jar file:

```
wget https://github.com/COMP90082-2023-SM1/GA-BoxJelly/raw/server-YK/src/cm-backend/target/coachingmate-0.0.1-SNAPSHOT.jar
```

4. Run the jar file: Now you can run the downloaded jar file. Use the following command:

```
java -jar coachingmate-0.0.1-SNAPSHOT.jar
```

If you want to run the jar file in the background and keep it running after closing the terminal, you can use the nohup command:

```
nohup java -jar coachingmate-0.0.1-SNAPSHOT.jar &
```

3. FrontEnd:

Run locally

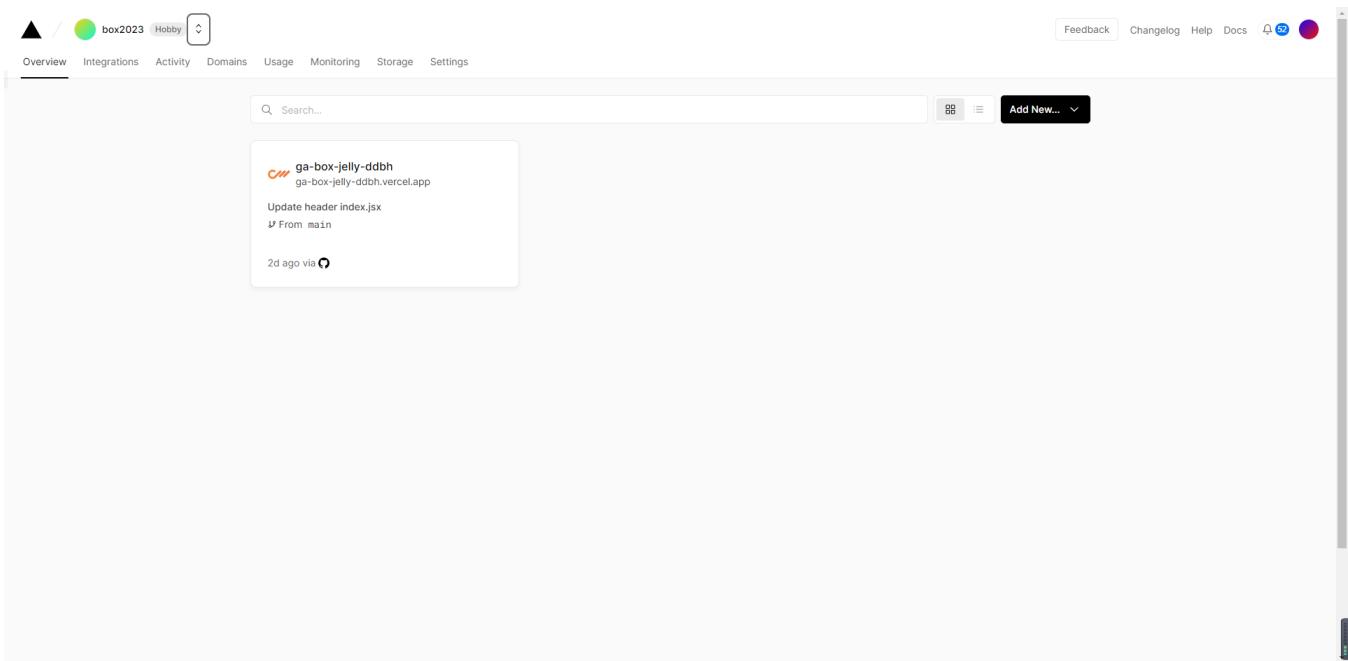
required: pre-installed: Nodejs V16.19.1

1. Navigate to src\cm-frontend-react folder
2. Run npm install
3. Run npm start

Online

Deploy online

1. Create a new project on Vercel



2. Connect it with Github

Let's build something new.

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

The image shows two side-by-side interface panels. The left panel is titled 'Import Git Repository' and features a search bar at the top. Below the search bar is a dropdown menu showing 'COMP90082-2023-SM1' selected. A list of repositories follows, with 'KevinZhoulk' and 'COMP90082-2023-SM1' highlighted. Other items include '+ Add GitHub Account', 'Switch Git Provider', 'comp90082-2023-sm1-resources · 50d ago', and 'FA-Web-App · 48d ago', each with an 'Import' button. At the bottom is a link 'Import Third-Party Git Repository →'. The right panel is titled 'Clone Template' and displays four template cards: 'NEXT.js' (with a preview showing a dark theme), 'SvelteKit' (with a preview showing a 'WELCOME' screen), 'Nuxt.js' (with a preview showing a dark theme), and 'Vite' (with a preview showing a yellow logo). Below the templates is a link 'Browse All Templates →'.

3. Import the required repository

Import Git Repository

COMP90082-2023-SM1

- GA-BoxJelly · 14d ago Import
- PR-Bluering · 10d ago Import
- comp90082-2023-sm1-resources · 50d ago Import
- FA-Web-App · 48d ago Import

Import Third-Party Git Repository →

Clone Template

- Next.js
- SvelteKit
- Nuxt.js
- Vite

Browse All Templates →

4. Select the source code folder

The screenshot shows the 'Configure Project' step of the deployment process. On the left, the project structure is visible, including a 'GIT REPOSITORY' section for 'COMP90082-2023-SM1/GA-BoxJelly' with branches 'main' and 'J'. Below this are links to 'Import a different Git Repository' and 'Browse Templates'. The main area shows a 'Configure Project' tab selected. A modal window titled 'Root Directory' is open, displaying a tree view of the project's directory structure. The directory 'cm-frontend-react' is highlighted with a red box. At the bottom of the modal are 'Cancel' and 'Continue' buttons.

Note:

Change the build command to `npm run build`

Change the install Command to `npm install`

Project Name
ga-box-jelly-zub9

Framework Preset
Create React App

Root Directory
src/cm-frontend-react Edit

Build and Output Settings

Build Command ?
npm run build OVERRIDE

Output Directory ?
build OVERRIDE

Install Command ?
npm install OVERRIDE

> Environment Variables

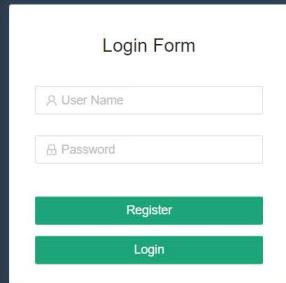
Deploy

5.Click the Deploy button.

User guide:

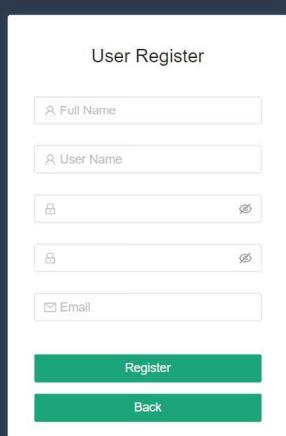
1.0 User Register & Connect to Garmin

1) Click the Register button



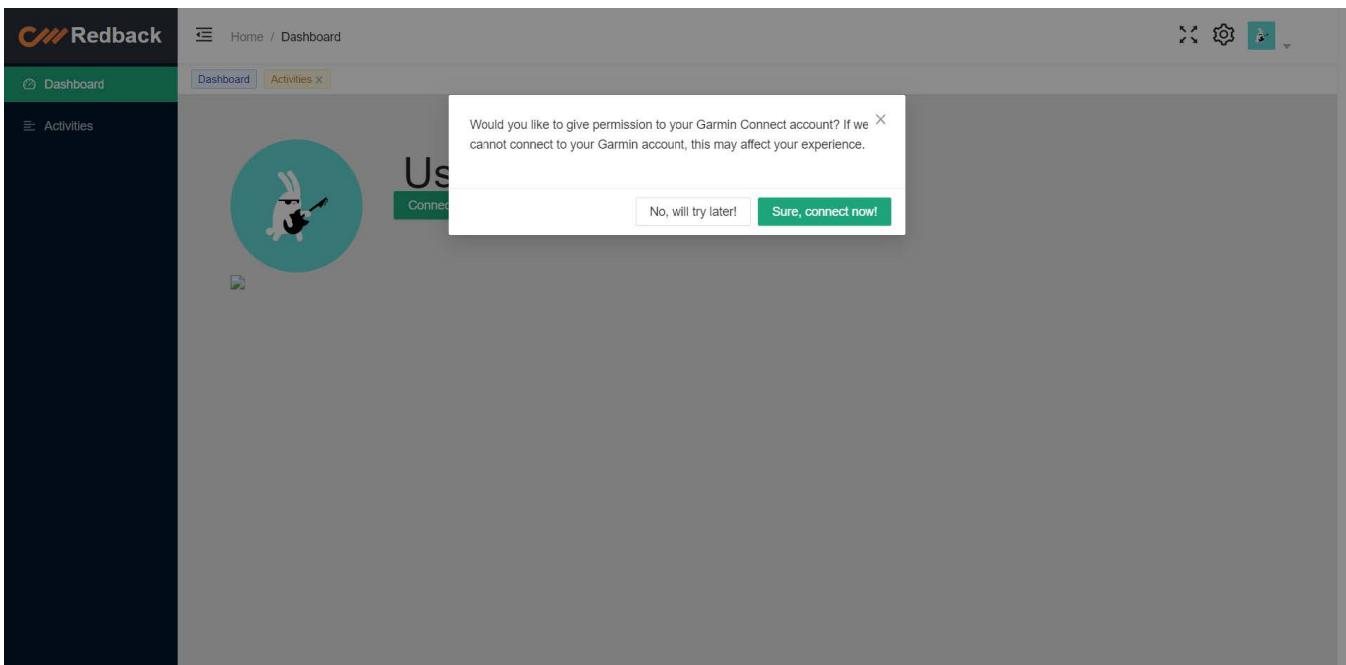
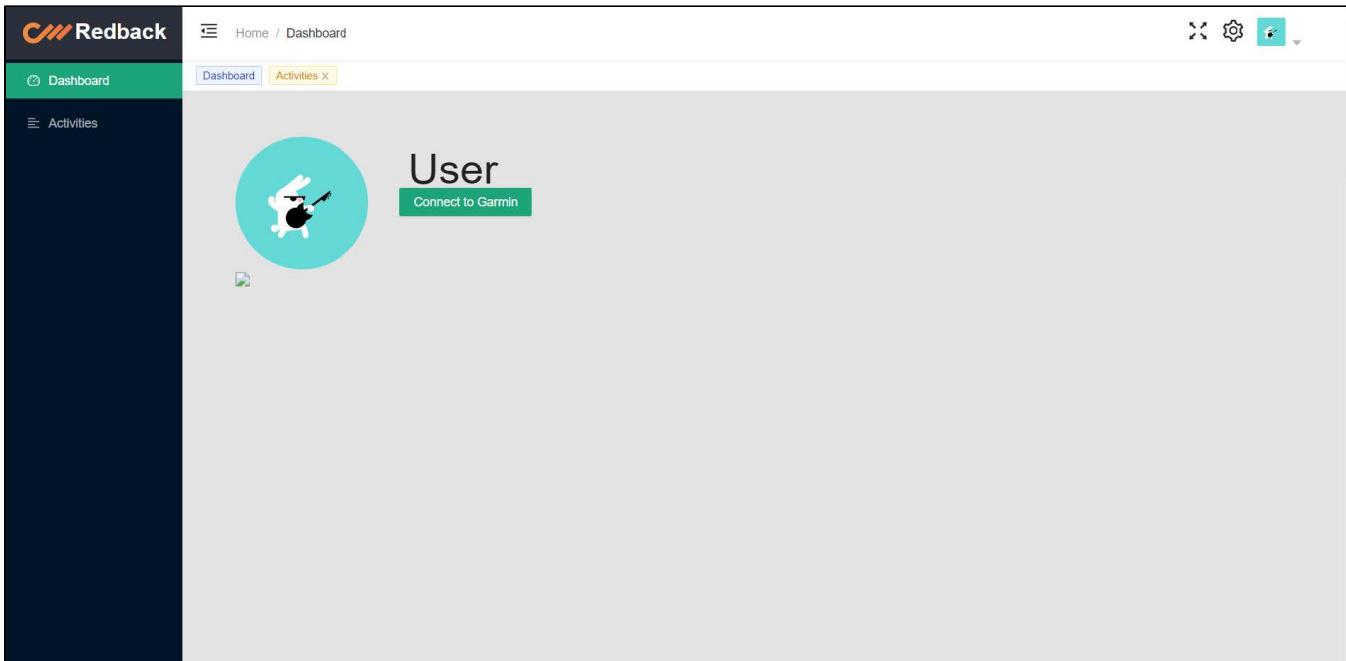
A screenshot of a mobile application's login screen. The title "Login Form" is at the top. Below it are two input fields: "User Name" with a user icon and "Password" with a lock icon. At the bottom are two green buttons: "Register" and "Login".

2) Fill in the form, then click Register Button



A screenshot of a mobile application's registration screen. The title "User Register" is at the top. It contains five input fields: "Full Name" (user icon), "User Name" (user icon), "Password" (eye icon), "Confirm Password" (eye icon), and "Email" (envelope icon). At the bottom are two green buttons: "Register" and "Back".

3) Click connect to Garmin, then choose to connect now



4) Fill in the form and agree on the Policy



Connect with CoachingMate-Boxjelly?

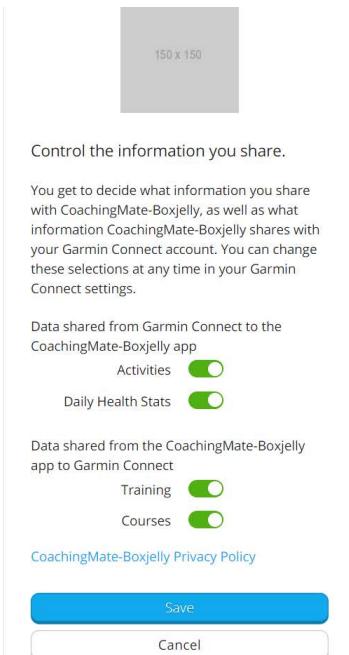
By connecting with CoachingMate-Boxjelly, you agree to share information from your Garmin Connect account to enhance your experience with CoachingMate-Boxjelly. This may include activities, location, heart rate and related metrics, calories burned and other health or personal data.

Do you agree to share information from your Garmin Connect account with CoachingMate-Boxjelly? You can opt out and disconnect from CoachingMate-Boxjelly at any time in your Garmin Connect settings.

[CoachingMate-Boxjelly Privacy Policy](#)

Agree

Do Not Agree



Control the information you share.

You get to decide what information you share with CoachingMate-Boxjelly, as well as what information CoachingMate-Boxjelly shares with your Garmin Connect account. You can change these selections at any time in your Garmin Connect settings.

Data shared from Garmin Connect to the CoachingMate-Boxjelly app

Activities

Daily Health Stats

Data shared from the CoachingMate-Boxjelly app to Garmin Connect

Training

Courses

[CoachingMate-Boxjelly Privacy Policy](#)

Save

Cancel



Connect with CoachingMate-Boxjelly?

By connecting with CoachingMate-Boxjelly, you agree to share information from your Garmin Connect account to enhance your experience with CoachingMate-Boxjelly. This may include activities, location, heart rate and related metrics, calories burned and other health or personal data.

Do you agree to share information from your Garmin Connect account with CoachingMate-Boxjelly? You can opt out and disconnect from CoachingMate-Boxjelly at any time in your Garmin Connect settings.

[CoachingMate-Boxjelly Privacy Policy](#)

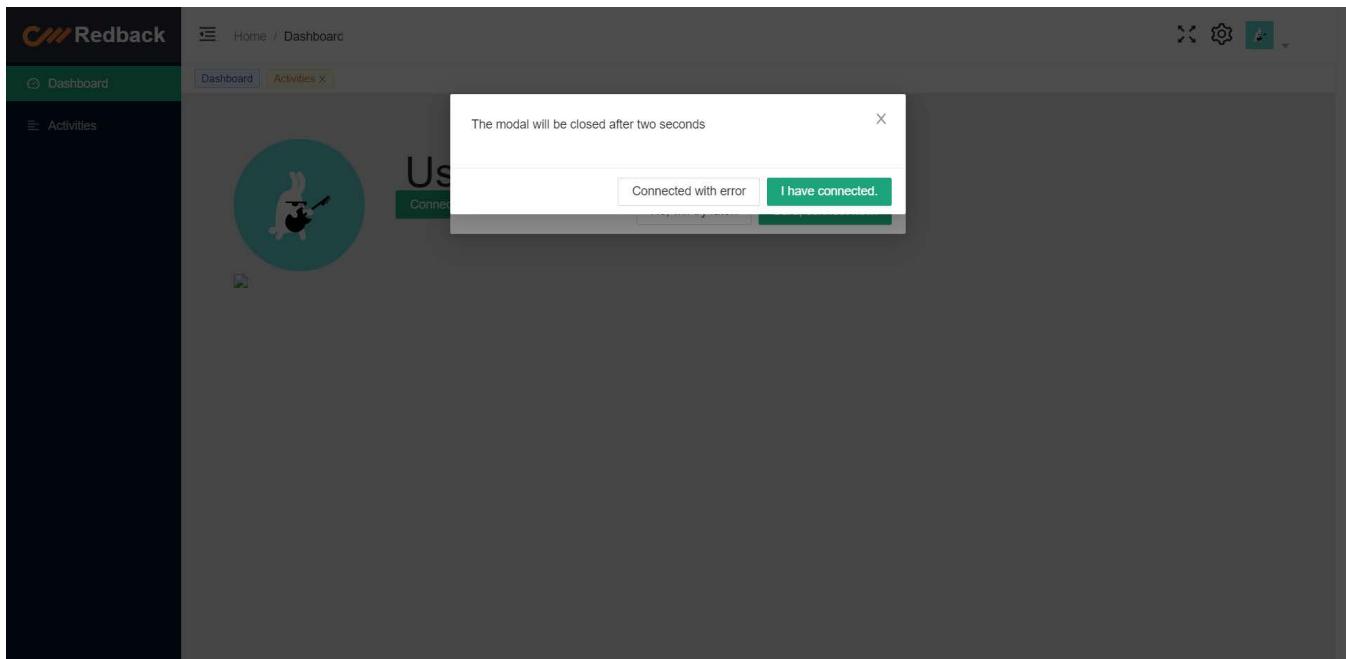
Garmin Account Sign-In

Email

Password ([Forgot?](#))

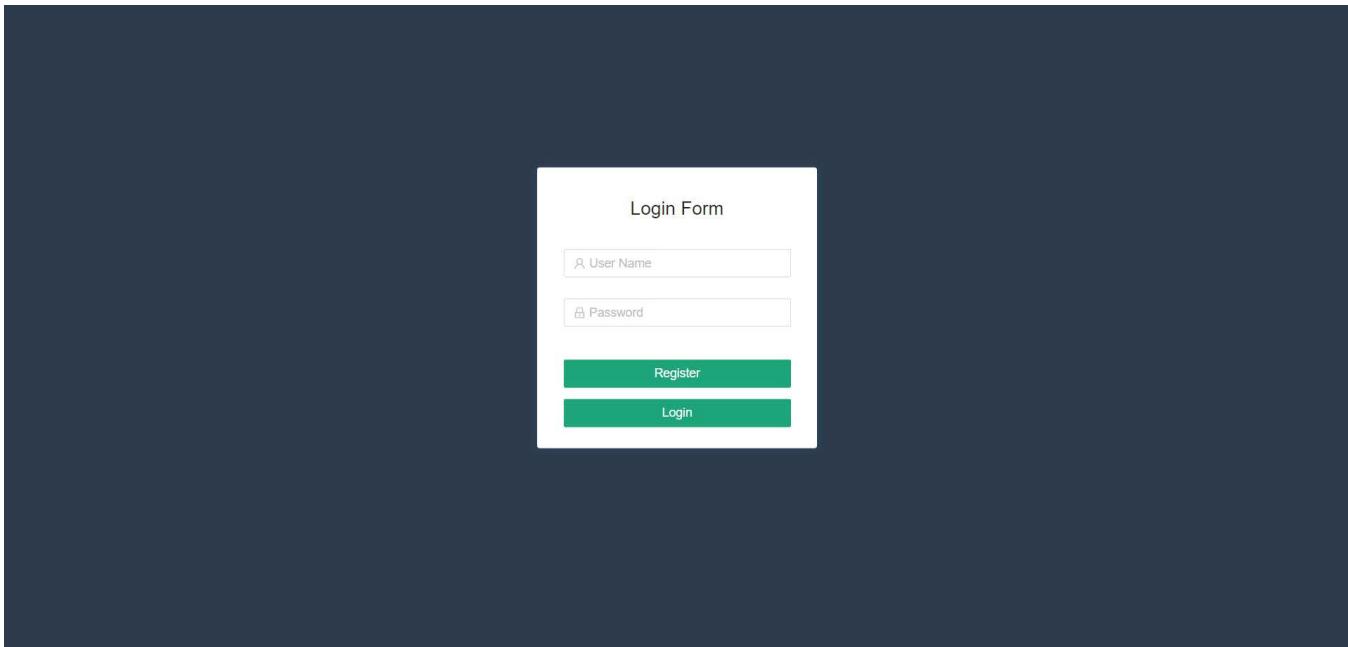
Remember Me

5) Click I have connected button



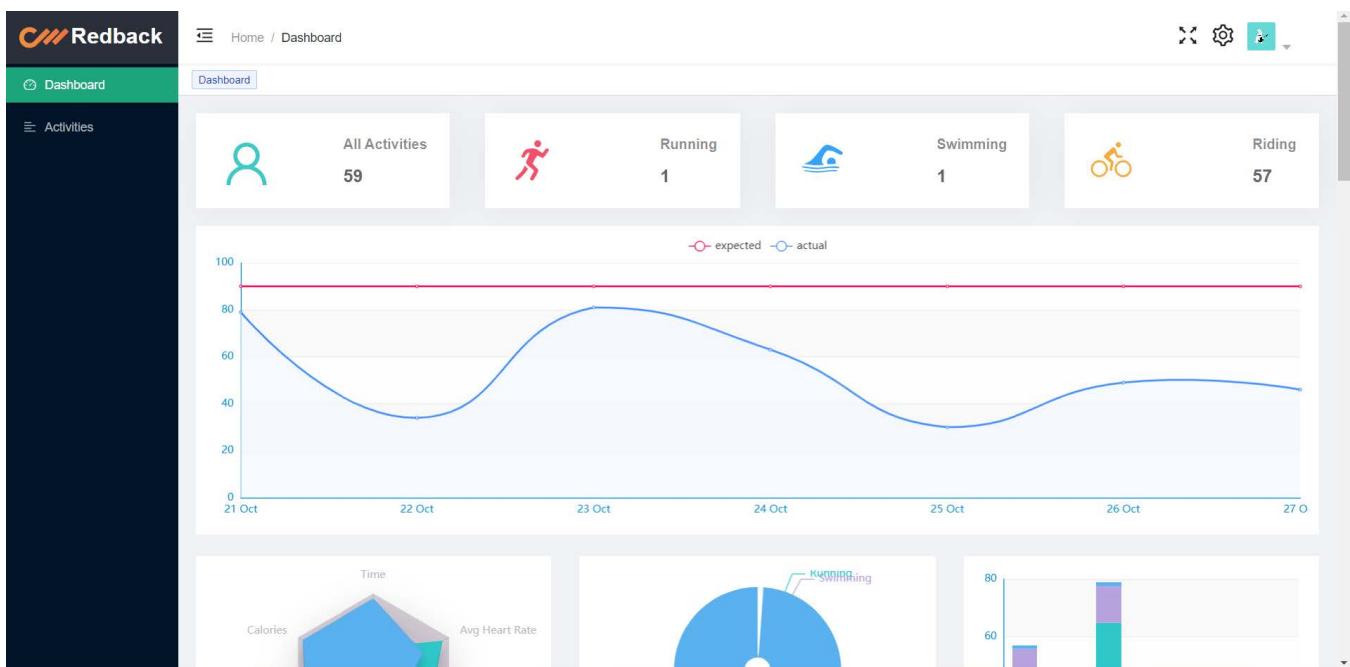
2.0 User login & View Activity

1) Click the activity button on the left



2.2 View activity list

1) Click the activity button on the left



Redback

Home / Activities

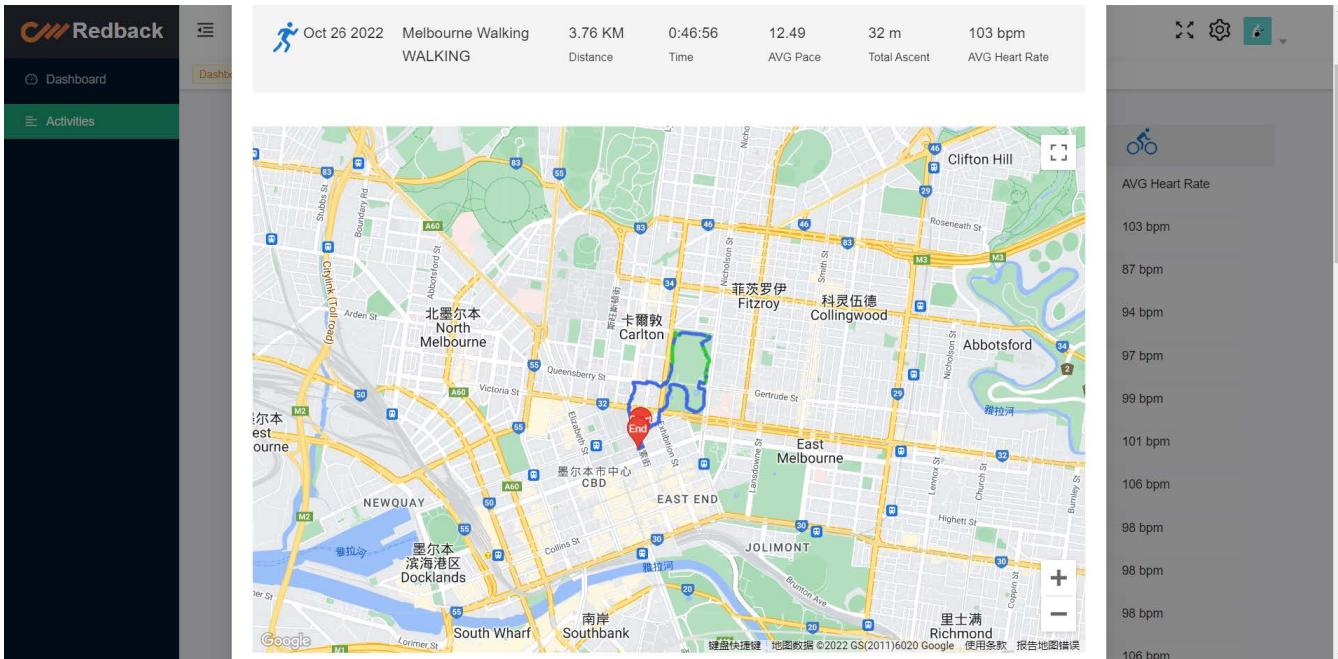
Dashboard Activities

All

Activity	Distance	Time	Avg Pace	Total Ascent	Avg Heart Rate
Oct 26 2022 Melbourne Walking WALKING	3.76 KM	0:46:56	12.49	32 m	103 bpm
Oct 25 2022 Melbourne Walking WALKING	4.17 KM	0:49:53	11.95	82 m	87 bpm
Oct 24 2022 Melbourne Walking WALKING	1.64 KM	0:22:17	13.61	59 m	94 bpm
Oct 24 2022 Melbourne Walking WALKING	0.61 KM	0:08:24	13.79	21 m	97 bpm
Oct 23 2022 Melbourne Walking WALKING	4.74 KM	1:03:06	13.32	73 m	99 bpm
Oct 22 2022 Melbourne Walking WALKING	3.80 KM	1:21:25	21.45	32 m	101 bpm
Oct 21 2022 Melbourne Walking WALKING	1.93 KM	0:34:40	17.94	16 m	106 bpm
Oct 20 2022 Melbourne Walking WALKING	14.04 KM	1:19:41	5.67	194 m	98 bpm
Oct 19 2022 Melbourne Walking WALKING	3.95 KM	0:55:25	14.03	76 m	98 bpm
Oct 17 2022 Melbourne Walking WALKING	2.34 KM	0:30:24	12.97	60 m	98 bpm
Oct 17 2022 Melbourne Walking	1.70 KM	0:21:09	12.44	34 m	106 bpm

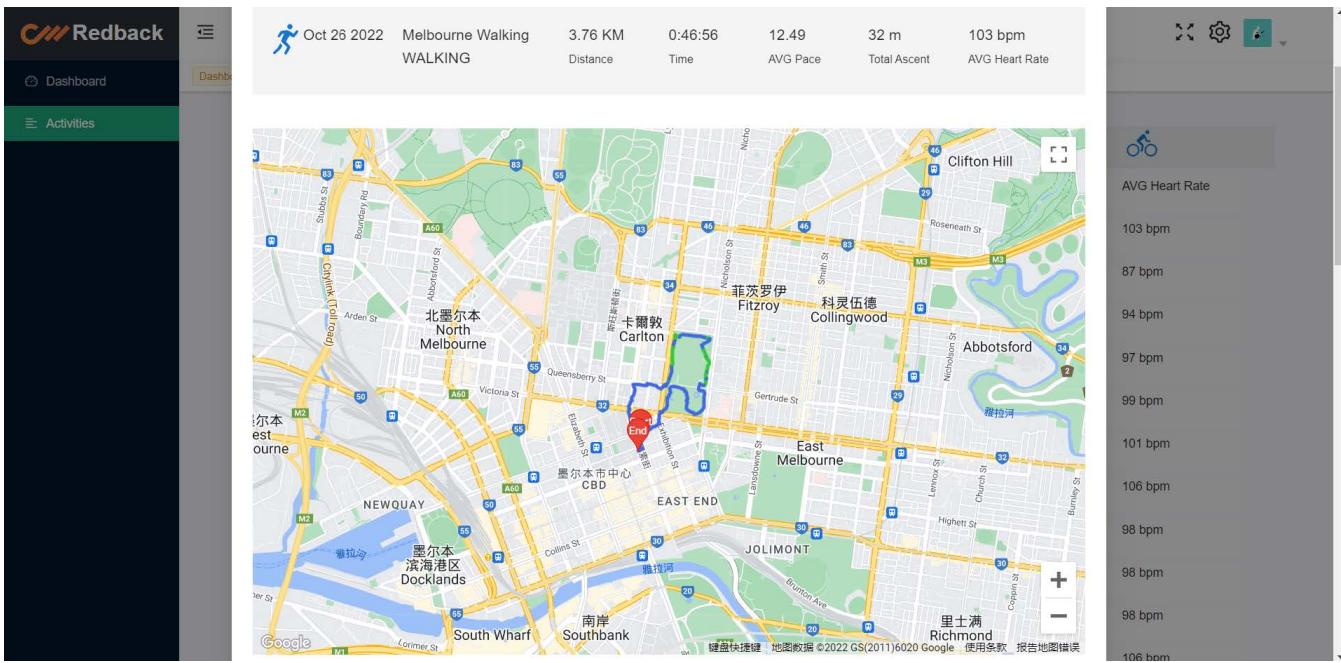
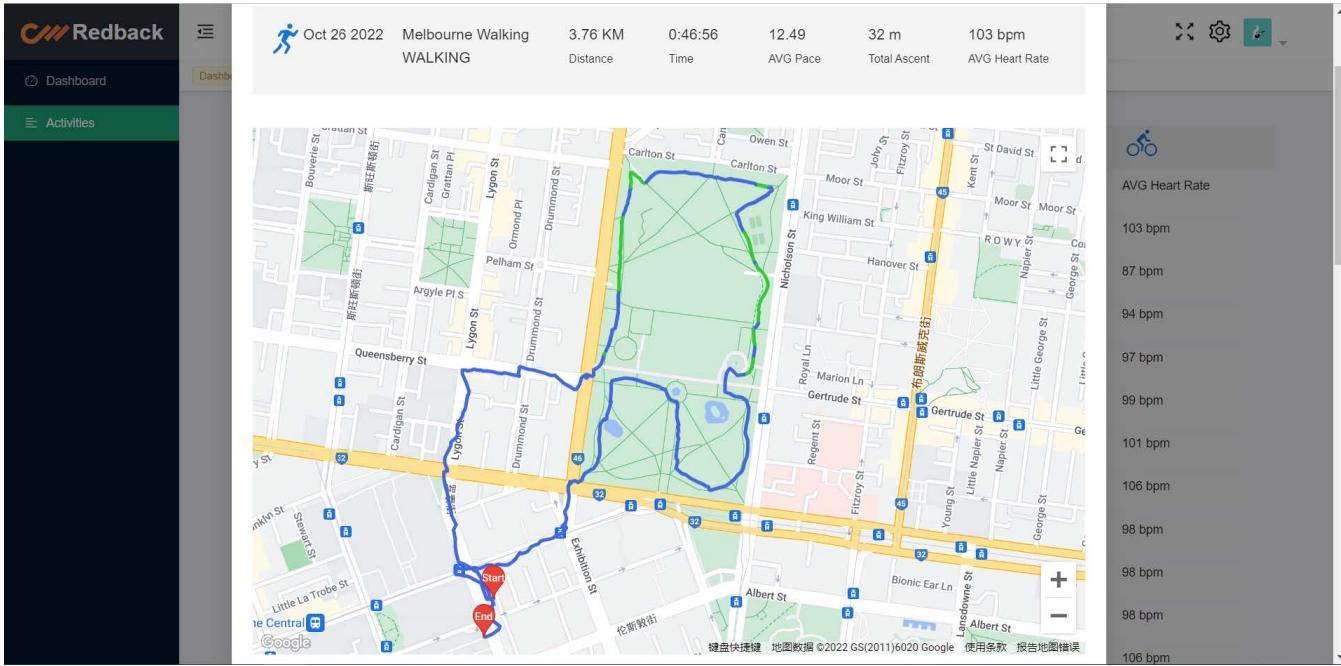
2.3 View basic data

1) Choose one activity and then view the basic data.



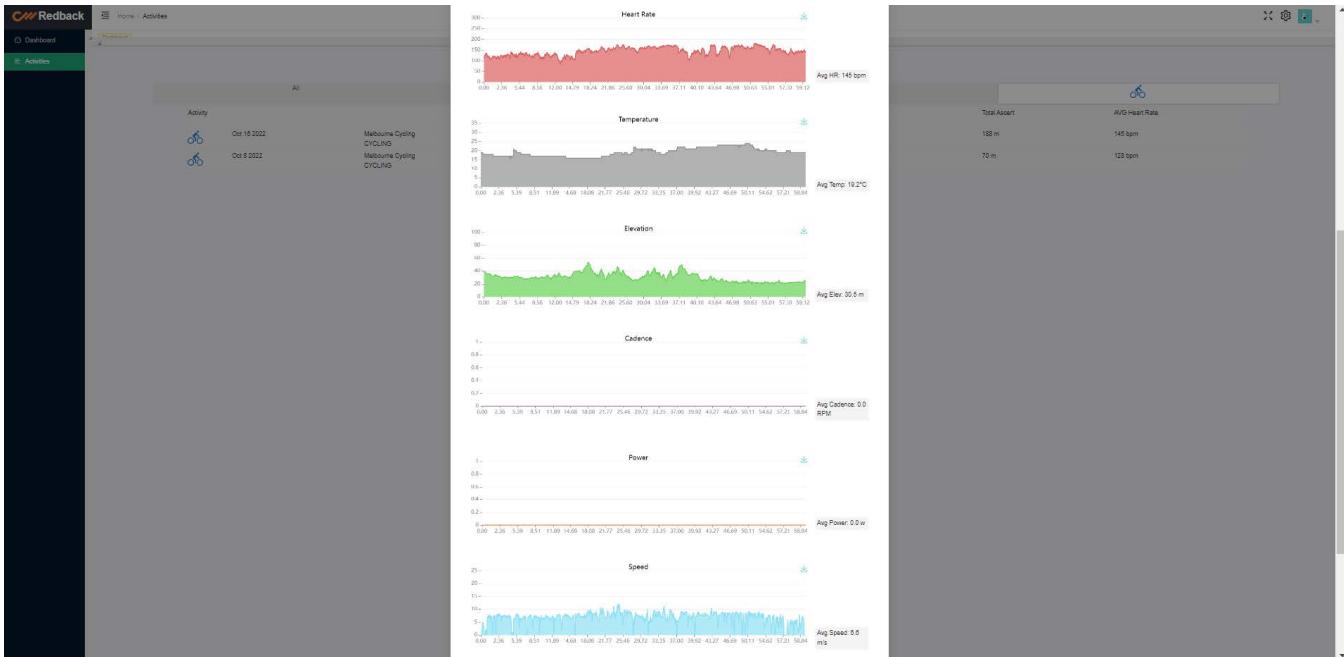
2.4 View heat map

- 1) Choose one of the activities and then view the heat map
- 2) Use the mouse and "+" and "-" functions on the map to move & zoom in and zoom out.
- 3) Colour is the strength of the heart rate



2.5 View the charts

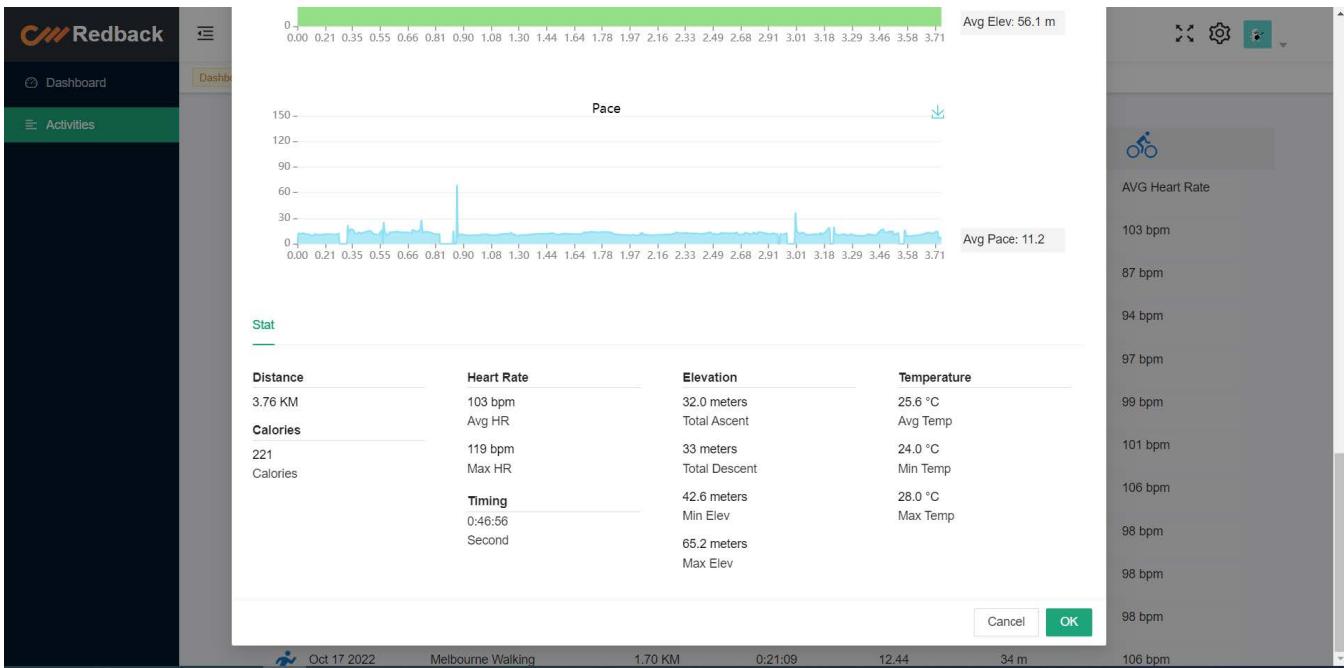
- 1) Choose one of the activities and then view the charts
- 2) Click the distance/ Time button above the chart to view the chart between distance and time;
- 3) View Different charts between Running, Swimming and Riding.





2.6 View the detailed data

1) Choose one of the activities and then view the detailed data



2.7 View activity list by sport type

1) Click the different icons above the list

The screenshot shows the Redback app displaying a list of activities. The top navigation bar includes the Redback logo, a dashboard link, and an "Activities" link. Below the navigation, there are three filter icons: a person walking (All), a runner (Walking), and a cyclist (Cycling). The table lists ten activities, each with a small icon, date, name, distance, time, pace, ascent, and heart rate:

Activity	Distance	Time	AVG Pace	Total Ascent	Avg Heart Rate
Oct 26 2022 Melbourne Walking WALKING	3.76 KM	0:46:56	12.49	32 m	103 bpm
Oct 25 2022 Melbourne Walking WALKING	4.17 KM	0:49:53	11.95	82 m	87 bpm
Oct 24 2022 Melbourne Walking WALKING	1.64 KM	0:22:17	13.61	59 m	94 bpm
Oct 24 2022 Melbourne Walking WALKING	0.61 KM	0:08:24	13.79	21 m	97 bpm
Oct 23 2022 Melbourne Walking WALKING	4.74 KM	1:03:06	13.32	73 m	99 bpm
Oct 22 2022 Melbourne Walking WALKING	3.80 KM	1:21:25	21.45	32 m	101 bpm
Oct 21 2022 Melbourne Walking WALKING	1.93 KM	0:34:40	17.94	16 m	106 bpm
Oct 20 2022 Melbourne Walking WALKING	14.04 KM	1:19:41	5.67	194 m	98 bpm
Oct 19 2022 Melbourne Walking WALKING	3.95 KM	0:55:25	14.03	78 m	98 bpm
Oct 17 2022 Melbourne Walking WALKING	2.34 KM	0:30:24	12.97	60 m	98 bpm
Oct 17 2022 Melbourne Walking	1.70 KM	0:21:09	12.44	34 m	106 bpm

CM Redback

Home / Activities

Dashboard Activities

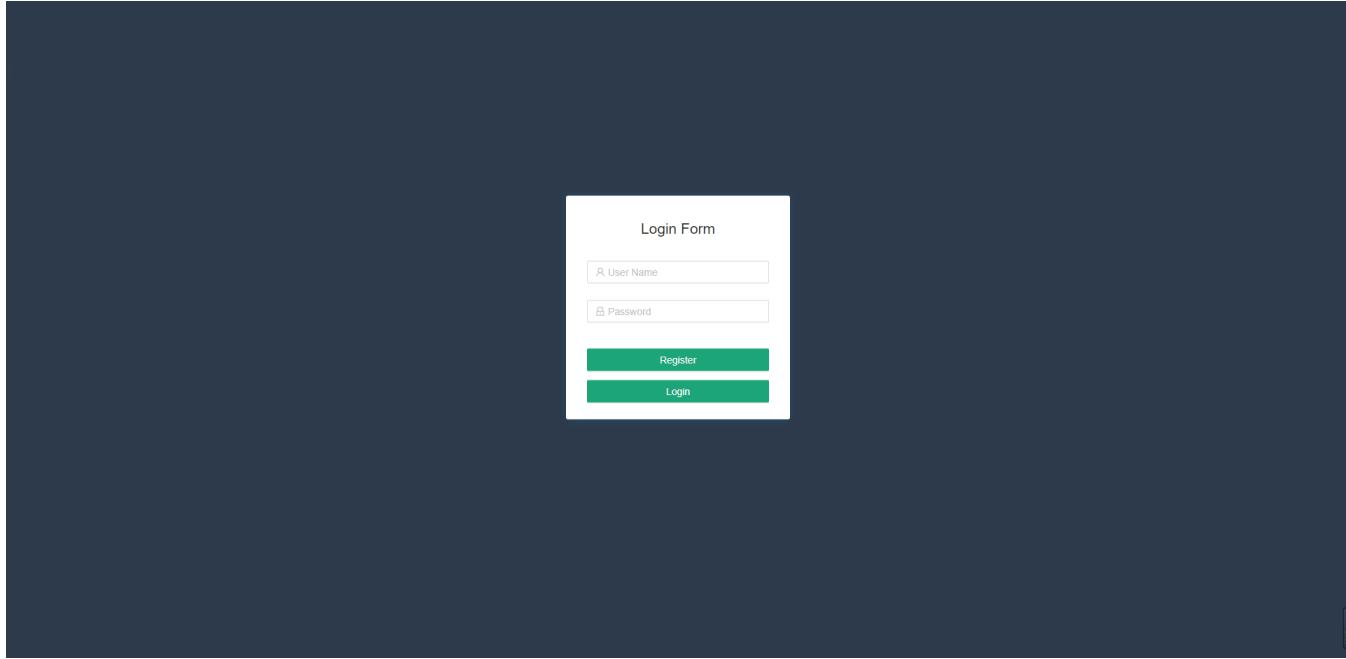
Dashboard

All	Run	Swim	Cycling		
Activity	Distance	Time	Avg Pace	Total Ascent	Avg Heart Rate
Oct 16 2022 Melbourne Cycling CYCLING	59.28 KM	2:28:42	2.51	188 m	145 bpm
Oct 8 2022 Melbourne Cycling CYCLING	5.04 KM	1:15:52	15.07	70 m	128 bpm

Project Product Demo

Url: <https://ga-box-jelly-ddbh.vercel.app/#/>

Example page:



Demo Account:

0428

Demo Password:

AAaa11..

Garmin account page:

Activities

+ Manual Activity Import Export CSV

Activity	Date	Type	Distance	Time	Avg Pace	Total Ascent	Avg HR
	Apr 29 2023	Act5 RUNNING	31.07 mi DISTANCE	1:30:00 TIME	2:54 /mi AVG PACE	100 ft TOTAL ASCENT	30 bpm AVG HR
	Apr 29 2023	Act4 INDOOR RUNNING	62.14 mi DISTANCE	30:00 TIME	0:29 /mi AVG PACE	100 ft TOTAL ASCENT	30 bpm AVG HR
	Apr 28 2023	Act3 BMX	9.32 mi DISTANCE	1:00:00 TIME	9.3 mph AVG SPEED	-- TOTAL ASCENT	0 Watts AVG POWER
	Apr 28 2023	Act2 INDOOR RUNNING	3.11 mi DISTANCE	1:00:00 TIME	19:19 /mi AVG PACE	-- TOTAL ASCENT	0 bpm AVG HR

CoachingMate activity page:

Dashboard Activities X

All						
Activity		Distance	Time	Avg Pace	Total Ascent	Avg Heart Rate
Apr 2023 30	Act5 RUNNING	50.00 KM	1:30:00	1.80	30.48 m	30 bpm
Apr 2023 30	Act4 INDOOR_RUNNING	100.00 KM	0:30:00	0.30	30.48 m	30 bpm
Apr 2023 28	Act3 BMX	15.00 KM	1:00:00	4.00	0 m	0 bpm
Apr 2023 28	Act2 INDOOR_RUNNING	5.00 KM	1:00:00	12.00	0 m	0 bpm
Apr 2022 17	CYCLING CYCLING	0.00 KM	0:00:10	71.53	0 m	92 bpm

Requirements

[Add Product requirements](#)

—
—

Requirements 1(Code cleaning)

Goals

- Complete the code cleaning and specification comments for the backend code of the entire project.

Background and strategic fit

Since this project has undergone many iterations and been handled by multiple teams, the code is very messy and bloated. Deployed and usable code needs to be reserved in a timely manner, and undeployed code and useless content should be cleaned up. Thereby maintaining the readability and stability of the entire project code.

Assumptions

- Subsequent technologists want a well-structured project and stability.

Requirements

#	Title	User Story	Importance	Notes
1	Code cleaning	As a technical maintainer, I want a well-structured project code that remains highly readable.	Must have	
2				

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
How can we solve the problem of code redundancy and low readability without affecting existing functionality?	Test the original functionality after the code is cleaned up.

Not Doing

Requirements 2(Initialize frontend & Backend)

Goals

- Run the project code locally

Requirements

#	Title	User Story	Importance	Notes
1	Run the project code locally		Must have	
2				

Not Doing

Requirements 3(Solve previous project left bugs)

Goals

- Fix the need bug left by the previous team

Background and strategic fit

The code left by the previous team has many component version incompatibility issues, so bugs need to be fixed

Assumptions

- As a user, I hope that every function displayed can be used smoothly.

Requirements

#	Title	User Story	Importance	Notes
1	• Fix the need bug left by the previous team		Must have	
2				

Requirements 4Solve previous project left security problems

Goals

- Fix security issues left over from the previous team designing the project.

Background and strategic fit

Since the project is iterated many times, and multiple teams simply add new functions, because the framework used by the project is too old, many security holes remain. So they need to be found and fixed.

Assumptions

- As a user of the project, I hope my system is sufficiently secure.

Requirements

#	Title	User Story	Importance	Notes
1	Fix security issues left over from the previous team designing the project.		Must have	
2				

Requirements 5 (Test in sprint 2)

Goals

Test the existing code API function because the previous team didn't write any test function.

- test the ActivityDataRetrieveController API
- test GarminPushController
- test LoginController
- test OAuthController
- test RegistryController
- test ActivityDaolmp
- test DashboardStatisticsDao
- test generateRequestTokenSecret
- test TokenDao
- test UserDaoTest
- test ActivityServiceImpl
- test UserServiceImpl

Background and strategic fit

Since the project is iterated many times, and multiple teams simply add new functions, we need to ensure all API works well.

Assumptions

- As a user of the project, I hope my all API works well.

Requirements:

#	Title	User Story	Importance	Notes
1	Test ActivityDataRetrieveController	As a technical maintainer, I want ActivityDataRetrieve api works well	Must have	
2	test GarminPushController	As a technical maintainer, I want GarminPush api to works well	Must have	
3	test LoginController	As a technical maintainer, I want Login api to works well	Must have	
4	test OAuthController	As a technical maintainer, I want OAuth api to works well	Must have	
5	test RegistryController	As a technical maintainer, I want registry api to works well	Must have	
6	test ActivityDaolmp	As a technical maintainer, I want activity data operation to works well	Must have	
7	test DashboardStatisticsDao	As a technical maintainer, I want DashboardStatistics data operation to works well	Must have	
8	test generateRequestTokenSecret	As a technical maintainer, I want RequestToken data operation to works well	Must have	
9	test TokenDao	As a technical maintainer, I want Token data operation to works well	Must have	
10	test UserDaoTest	As a technical maintainer, I want User data operation to works well	Must have	
11	test ActivityServiceImpl	As a technical maintainer, I want activity service functions to works well	Must have	
12	test UserServiceImpl	As a technical maintainer, I want user service functions to works well	Must have	

Requirements 6 (Looking for possible security issues)

Goals

- Ensure the security of the back-end system and maintain user data security

Background and strategic fit

Since the project has been taken over and modified by multiple project teams, and the functions have been added many times, there may be great flaws and loopholes in the security of each link. These flaws and loopholes will compromise the security and stability of the system. So it needs to be discovered and recorded in time.

Assumptions

- Project client want to use the system safely and worry-free.

Requirements

#	Title	User Story	Importance	Notes
1	Security Check	As a CoachingMate owner, I want to be able to use the system safely and without worry.	Must Have	
2				

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

Requirements 7(Ensure accessibility)

Goals

- Connect to Garmin Connect, establish data communication channel and fetch data from Garmin.

Background and strategic fit

As the document of Garmin API suggests, the data can only be received with a https connection. Thus, to make sure exercise data is synchronized, the backend needs to run on a server with SSL certificate. And the endpoints on Garmin API should be configured.

Assumptions

- All configurations are correctly done and there is data in Garmin account waiting to be accessed.

Requirements

#	Title	User Story	Importance	Notes
1	Ensure accessibility	As a web application user, I would like to see the real-time data from Garmin using this application.	Must have	
2				

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
How to make sure the data is correct and integrated?	Compare data received that of the Garmin account and make sure they are identical.

Not Doing

Requirements 8(Disconnect Garmin Connect)

Goals

- Complete the deployment of the add the function of disconnect Garmin Connect API code on the backend.
- Improve user experience and control over their data sharing.

Background and strategic fit

To enhance user control and privacy, we need to provide an option for users to disconnect their Garmin Connect account from the application. This will enable them to stop sharing their data or switch to another account if needed.

Assumptions

1. Users have previously connected their Garmin Connect account to the application.
2. Users are aware of the consequences of disconnecting their Garmin Connect account.
3. The backend infrastructure can support the disconnect functionality.
4. Users are aware that they care about their Garmin connect authentication.

Requirements

#	Title	User Story	Importance	Notes
1	Disconnect Garmin Connect	As a user, I want to be able to disconnect my Garmin Connect account from the application, so that I can stop sharing my data or switch to another account if needed. This can be achieved by adding a button or option in the settings page that allows me to disconnect my Garmin Connect account and remove any related data from the application. Additionally, there should be a confirmation message or prompt to ensure that I am aware of the consequences of this action and that I want to proceed.	should have	
2				

Requirements 9(Change Garmin connect account)

Goals

- Enable users to easily disconnect their current Garmin Connect account and connect to another one.
- Improve user experience by providing flexibility to switch between Garmin Connect accounts.

Background and strategic fit

Users may have multiple Garmin Connect accounts or may want to switch to another account for various reasons. To cater to such scenarios, we need to provide a seamless experience for users to disconnect their current account and connect a new one.

Assumptions

- Users have a valid Garmin Connect account that they want to connect to.
- The application can handle multiple Garmin Connect accounts.
- Users understand the implications of disconnecting and connecting different Garmin Connect accounts.

Requirements

#	Title	User Story	Importance	Notes
1	Change Garmin connect account	As a user, I want to be able to disconnect my current Garmin Connect account and connect to a different one, so that I can use my preferred Garmin Connect account to track my activities on the platform. The new feature should allow me to easily disconnect my current account and provide a straightforward way to connect to another Garmin Connect account.	Should have	
2				

Requirements 10(Reset password function)

Goals

- Provide a secure and easy-to-use reset password function for users.
- Improve user experience and trust in the software application by ensuring account security.
- Increase user satisfaction with the application's functionality.

Background and strategic fit

A reset password function is essential for any software application that requires user authentication. By providing this feature, we can ensure that users can easily recover access to their accounts in case they forget their passwords or need to update them for security reasons.

Assumptions

- Users have access to the email address associated with their account.
- The application's backend infrastructure can support the password reset process.
- Proper security measures are in place to protect user data during the password reset process.

Requirements

#	Title	User Story	Importance	Notes
1	Reset password function	As a user, I want to be able to reset my password in case I forget it or need to update it for security reasons. By adding a reset password function, I can easily recover access to my account and ensure the security of my personal information. This will improve the user experience and increase trust in the software application. This will contribute to the success of the project by ensuring that users have a positive experience with the application and are satisfied with its functionality.	Must have	
2				

Requirements 11(Cloud front-end deployment)

Goals

- Deploy the application's front-end on a cloud platform to ensure quick and secure access.
- Enable seamless scalability and performance improvements for the application.

Background and strategic fit

Cloud front-end deployment is essential for modern applications to provide users with a fast, secure, and reliable experience. By leveraging a cloud platform's global infrastructure, scalability, and performance improvements, the application can handle a growing user base and increased demand, ensuring its long-term success.

Assumptions

- The application's front-end is compatible with the chosen cloud platform.
- The cloud platform provides adequate security, performance, and scalability features.
- The development team has the necessary skills and knowledge to perform the cloud front-end deployment.

Requirements

#	Title	User Story	Importance	Notes
1	Cloud front-end deployment	As a user, I want to access the application quickly and securely through a cloud-based front-end deployment so that I can enjoy a reliable and responsive experience regardless of my location. By deploying the application's front-end on a cloud platform, I can benefit from its global infrastructure, scalability, and performance improvements. This will contribute to the success of the project by providing a seamless user experience and ensuring that the application can handle a growing user base and increased demand.	Must have	
2				

Requirements 12(Fix transferring_issue)

Goals

- Avoid putting account information in url request parameter.

Background and strategic fit

The url request shouldn't show user information. We need to put these information in requestbody to avoid security issue.

Assumptions

- All users can register and login successfully.

Requirements

#	Title	User Story	Importance	Notes
1	Ensure security	As a web application user, I don't want my username and password exposed on url.	Must have	
2				

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
How to make sure the data is correct and integrated?	Compare data received that of the Garmin account and make sure they are identical.

Not Doing

Requirements 13(Add SMTP email server to the backend)

Goals

Background and strategic fit

Assumptions

Requirements

#	Title	User Story	Importance	Notes
1	Ensure security	As a web application user, I don't want my username and password exposed on url.	Must have	
2				

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
How to make sure the data is correct and integrated?	Compare data received that of the Garmin account and make sure they are identical.

Not Doing

Requirements 14(Integrationw with SpringSecurity)

Goals

- Use Spring Security for authentication

Background and strategic fit

Use the Spring Security and JWT to manage the authentication of user. When a user login, the browser will be assign a token(with a expire date). Then they can access APIs by carrying the validated token in header.

The token is generated by JWT library.

The login function and accessibility to API are integrated with Spring Security. We need to customize and config it according to our project.

Assumptions

- With validated token, user can access the backend api. Otherwise, they can not access the backend api.

Requirements

#	Title	User Story	Importance	Notes
1	Ensure security	As a web application user, I don't want others can access my account information.	Must have	
2				

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
How to make sure the security is correct and integrated?	By Postman, we can try to use different way to check the accessibility to API

Not Doing

Token should be stored in Redis.

Different user may have different role to have different authorization to be able to access different API.

Project Planning

Sprint 1 Plan: The main plan of sprint1 is to contact the client and supervisor to establish a stable communication channel. Then understand the client's expectations and actual needs for the project through a meeting with the client. And build a plan based on these user stories.

Sprint 2 Plan: According to the information obtained from the meeting with Hanna, the back-end code of the project is sorted out and cleaned, and the potential security problems in the current back-end, as well as the existing obvious bugs are initially repaired and test and document records are set.

Sprint 3 Plan: According to the needs of customers, further develop the functions of the back-end API. It is planned to solve the task last sprint left especially receiving the activity data from Garmin connect (Major task) and add the session security oauth feature into backend if time permitted.

Sprint 4 Plan: According to the feedback of the final product presentation, further polish product and complete the project handover process (only for minor adjustments and not including any feature adjustments).

Sprint 1 Plan

Sprint 1

Goal:

Establish a stable communication channel with the client and supervisor, understand the client's expectations and needs through a meeting, and create a project plan based on the gathered user stories.

Approach:

- Hold a kick-off meeting with the client
- Check the project code handed over
- Run the test and record the problems that need to work on in the future.

Deliverable:

- Hold the kick-off meeting (with Eduardo)
- Hold technical presentation discussion meeting with Hanna
- Successfully run the handover project code in the local environment.
- Complete all project documentation listed on the sprint 1 checklist.
- Document the problems encountered when deploying the handed-off project code locally and the problems encountered when running the tests.

Sprint 2 Plan

Sprint 2

Goal:

Based on the information and requirements gathered from the kick-off meeting with Eduardo and the technical briefing meeting with Hanna, organize the requirements and implement security improvements and necessary fixes to the backend code.

Approach:

- Perform code cleaning on all the main function codes of the backend to remove unused redundant codes, remove non-standard non-English comments, and add missing comments according to the java development specification. Thereby refactoring the structure of the code, increasing the readability and stability of the code. Prepare for subsequent programming.
- Solve the security problem of storing user passwords in clear text in the database.
- Solve system security problems caused by users who may use weak passwords.
- Solve the bug that an error will occur when users register with the same email address.
- Solve the problem that the system cannot display the path of the heat map.
- Solve the problem that the system cannot receive the activity data pushed by Garmin connect.
- Addresses functionality issues where users were unable to reset their passwords.

Deliverable:

- The back-end and front-end programs have been successfully deployed on the server and can be initially run some test.
- The code can successfully pass the functional test to ensure that the normal functions implemented by the back-end code will not be affected by the progress of code cleaning.
- Implemented a secure method for storing user passwords in the database, eliminating the security issue of storing passwords in clear text.
- Passes tests ensuring user only can set the strong password.
- Fixed the bug that caused errors when users register with the same email address. Passes tests which verifying the process of duplicate email addresses during registration.
- Resolved the issue where the system could not display the path of the heat map. Pass the displaying heat map paths test.
- Ensured that the system can receive activity data pushed by Garmin Connect. Pass the receiving activity data test.
- Solved functionality issues where users were unable to reset their passwords. Pass the tests allowing users to reset their passwords successfully.

Sprint 3 Plan

Sprint 3

Goal:

According to the needs of customers, further develop the functions of the back-end API. It is planned to solve the task last sprint left especially receiving the activity data from Garmin connect (Major task) and add the session security oauth feature into backend if time permitted.

Approach:

1. Sprint Planning: Gather the development team and relevant stakeholders to discuss the sprint goal and prioritize tasks. In this sprint, focus primarily on receiving activity data from Garmin Connect and then adding the session security OAuth feature into the backend.
2. Task Breakdown: Break down the sprint goal into smaller, manageable tasks. For this sprint, tasks may include:
 - a. Test and validate the received data to ensure accuracy and completeness.
 - b. Implement the code fix to PushFile method in the backend program.
 - c. Test and validate the received data to ensure accuracy and completeness.
 - d. Add error handling and logging.
 - e. Finish the last sprint left task-Add the function of disconnect Garmin Connect.
 - f. Test and validate the function of disconnect Garmin Connect.
 - g. Add error handling and logging.
 - h. Finish the last sprint left task-Make users be able to change to another connect account
 - i. Test and validate the function of users change to another connect account.
 - j. Add error handling and logging.
 - k. Finish the last sprint left task-Add reset password function
 - l. Test and validate the function of reset password function.
 - m. Add error handling and logging.
 - n. Study OAuth implementation best practices and select the most suitable library or tool for the backend.
 - o. Implement the session security OAuth feature in the backend.
 - p. Test and verify the OAuth functionality to ensure secure and reliable sessions.
 - q. Update documentation and API usage guidelines for the new features.
3. Daily Stand-ups: Conduct daily stand-up meetings to discuss progress, blockers, and any necessary adjustments to the sprint plan. This will help the team stay on track and address any issues promptly.
4. Collaboration and Knowledge Sharing: Encourage team members to collaborate, share knowledge, and support each other throughout the sprint. Set up communication channels and provide resources for technical assistance.
5. Code Review and Quality Assurance: Perform regular code reviews and maintain coding standards to ensure the quality of the backend API features. Implement automated testing and continuous integration to catch any errors or issues early in the development process.
6. Sprint Review and Retrospective: At the end of the sprint, conduct a sprint review to demonstrate the completed features to stakeholders and gather feedback. Additionally, hold a retrospective meeting with the development team to discuss the sprint's successes and areas for improvement. Use this feedback to inform and optimize future sprints.

Deliverables:

- Completed receiving activity data from Garmin Connect: A fully functioning and tested data receiving function with Garmin Connect, enabling the backend to receive activity data and providing users with the ability to disconnect from Garmin Connect, change accounts, and reset passwords. This includes:
 1. Source code for the updated and fixed PushFile method, along with other implemented features.
 2. Test cases and test results, demonstrating the correctness and reliability of the main function, receiving activity data from Garmin Connect.
 3. Error handling and logging mechanisms to ensure smooth operation and ease of troubleshooting.
- Session Security OAuth Feature: A secure and reliable OAuth implementation for session management in the backend. This includes:
 1. Source code for the OAuth feature, adhering to best practices and coding standards.
 2. Test cases and test results, verifying the security and functionality of the OAuth implementation.
 3. Updated API usage guidelines, including instructions for using the OAuth feature.
- Sprint Documentation: Comprehensive documentation for the entire sprint, detailing the approach, progress, and outcomes. This includes:
 1. Sprint plan, outlining the prioritized tasks and timeline.
 2. Meeting notes and action items from daily stand-ups, sprint review, and retrospective.
 3. Collaboration artifacts, such as shared resources, communication channel logs, and knowledge-sharing materials.
- Quality Assurance Reports: Reports and analysis of the code quality, test, and overall performance of sprint 3. This includes:
 1. Code review sheet, highlighting areas of improvement and adherence to coding standards.
 2. Test coverage specification, identifying potential gaps in testing and areas requiring additional validation.
 3. Performance metrics, demonstrating the efficiency and scalability of the implemented features.

Sprint 4 Plan

Sprint 4

Goal:

According to the feedback of the final product presentation, further polish product and complete the project handover process (only for minor adjustments and not including any feature adjustments).

Approach:

1. Complete the further polish product based on the feedback of the final product presentation
2. Complete the project handover process
 - a. Complete and documented user guide
 - b. Complete and documented project handover confluence page

Deliverables:

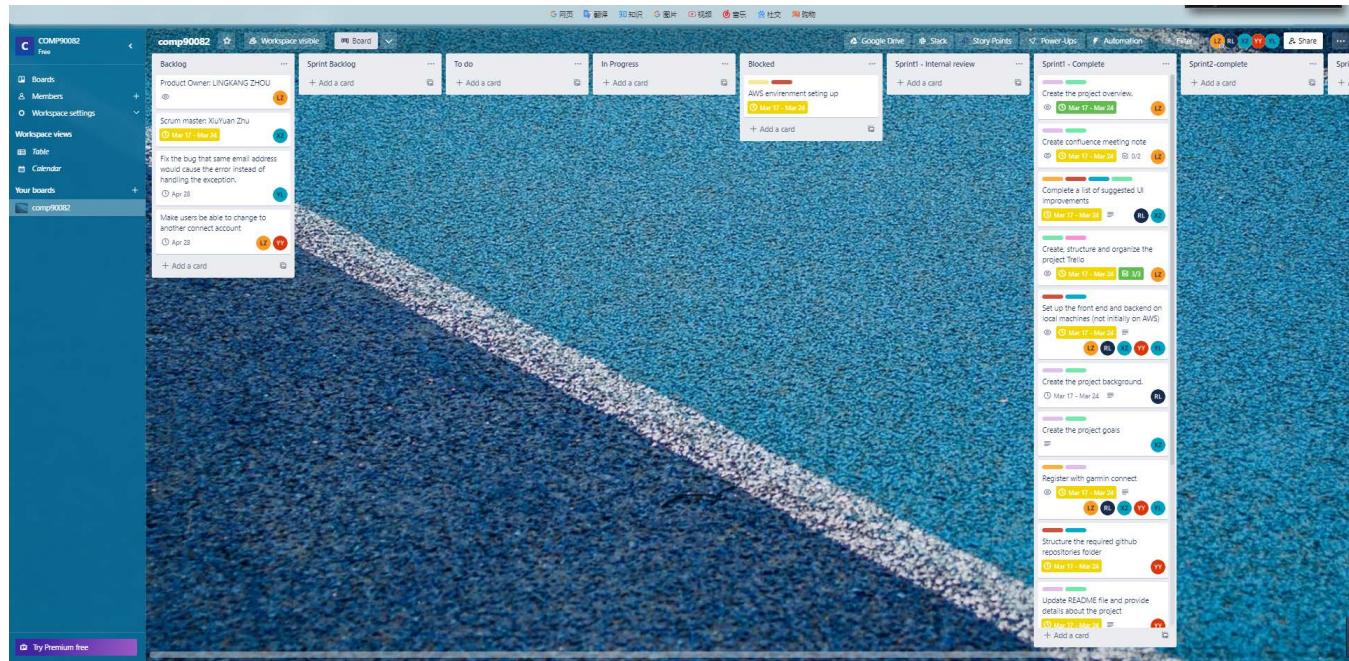
- Complete deployment guide
- Complete user guide page
- Complete project handover confluence page

Trello redirect page

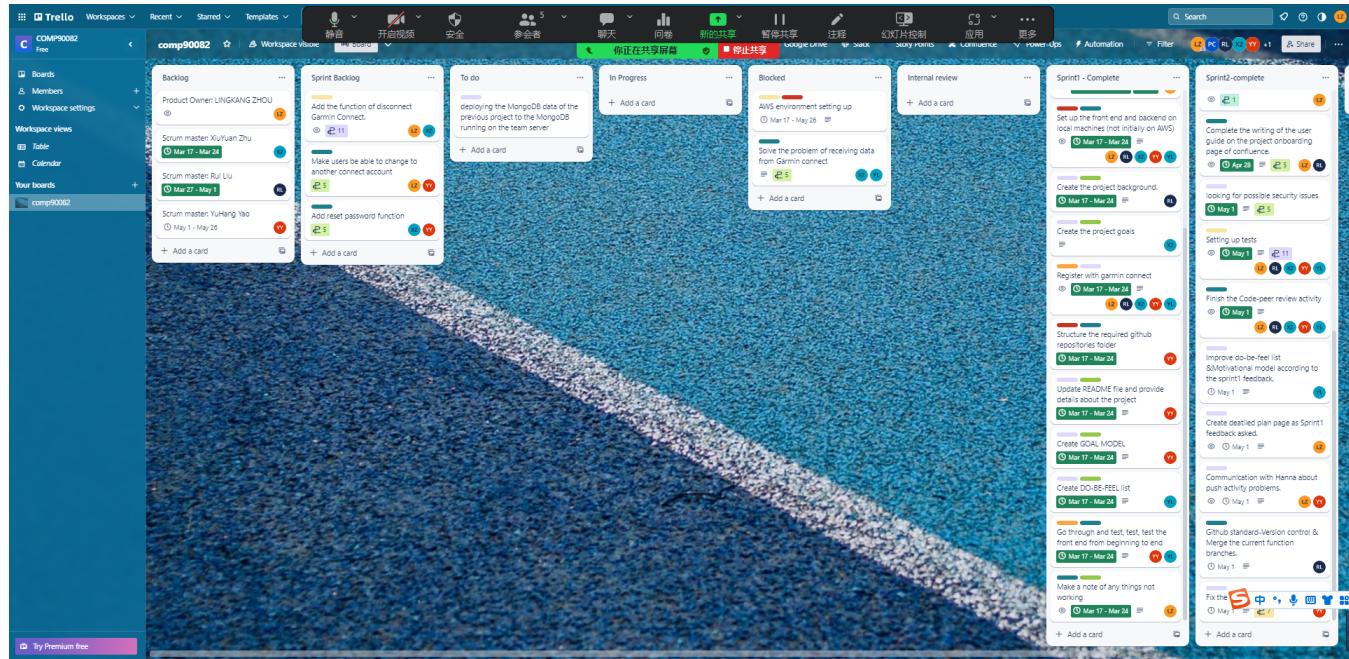
Link

<https://trello.com/invite/b/3wup1hgd/ATTI33455e94acf843ca04f6038d8971483002FBBBDE/comp90082>

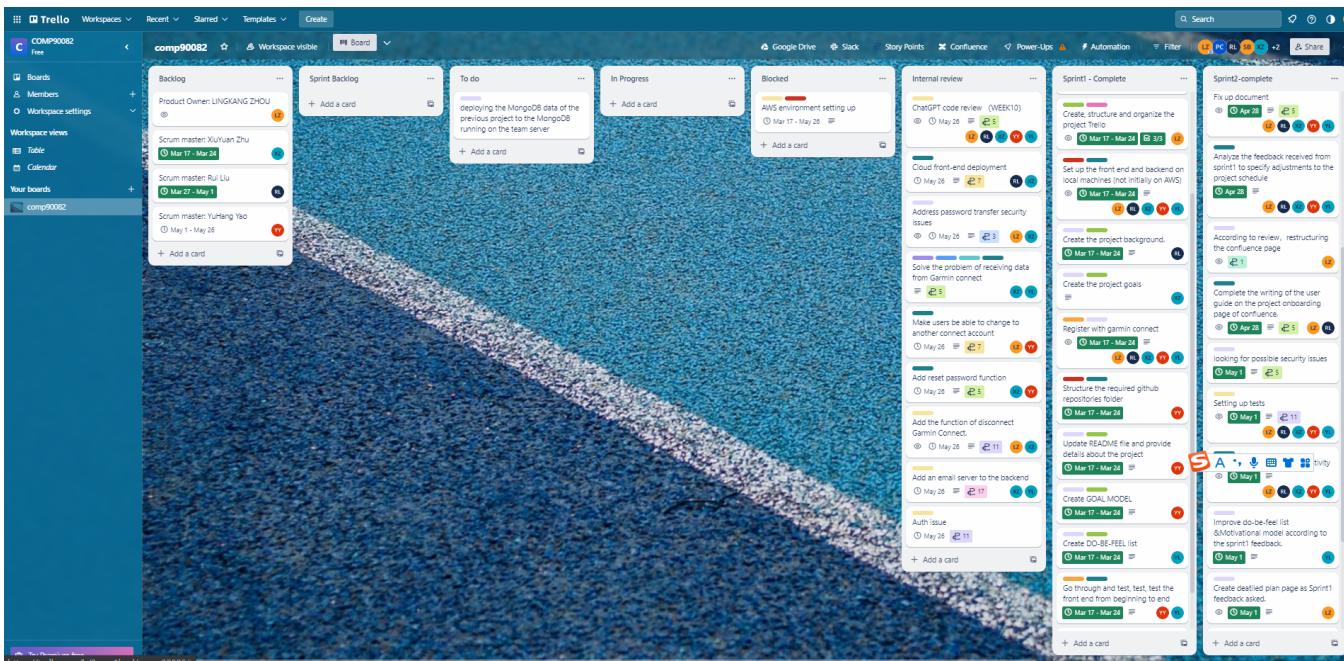
Sprint1



Sprint 2:



Sprint 3:



Sprint 4:

User Story

Sprint 1:

[Sprint 1 related User Story](#)

Sprint 2:

[Sprint 2 related User Story](#)

Sprint 3:

[Sprint 3 related User Story](#)

Sprint 4:

Sprint 1 related User Story

Sprint 1

Epic ID	Epic	UserStory ID	User	Story	Moscow Priority	Acceptance criteria
1	Integration of data from Garmin watches to a servere	1a	CoachingMate owner	I want to securely connect to exerciser's Garmin watches, so that I can receive their workout data in real-time.	must	The user can securely connect to exerciser's Garmin watches.
		2a	exerciser	I want to easily connect my watch to the CoachingMate system, so that my workout data is available for my coach to review.	must	The data is synced to the coaching mate system
		3a	CoachingMate owner	I want to store the collected data from Garmin watches in an organized and secure manner, so that I can access it for analysis and guidance.	should	The data is well-displayed in the dashboard so that the user can drag or analysis data easily
		4a	exerciser	I want to have control over which data is shared with the CoachingMate system, so that I can maintain my privacy.	must	The user can control which data is allowed to be shared to others.
2	Display of collected data for better workout guidance	1b	CoachingMate owner	I want to easily visualize and analyze the collected data from exercisers, so that I can provide personalized workout guidance.	should	The user can visualize and analyze the collected data from exercisers.
		2b	exerciser	I want to view my workout data in the CoachingMate system, so that I can track my progress and achievements.	must	The user can view the workout data in the CoachingMate system.
		3b	CoachingMate owner	I want to set up alerts and notifications based on exerciser's data, so that I can provide timely advice and support during their workout sessions.	should	Exerciser data can be configured for alerts and notifications.
		4b	exerciser	I want to receive personalized feedback from my coach based on my workout data, so that I can improve my performance and reach my fitness goals.	should	The user can receive personalized feedback from the coach based on the workout data.

Sprint 2 related User Story

2023

Sprint goals:

According to the information obtained from the meeting with Hanna, the back-end code of the project is sorted out and cleaned, and the potential security problems in the current back-end, as well as the existing obvious bugs are initially repaired and test and document records are set.

Completed Work:

Task 1

Analyze the feedback received from sprint1 to specify adjustments to the project schedule

User story:

As a project manager, I want to analyze the feedback received from sprint 1 in order to identify areas for improvement and adjust the project schedule accordingly. By carefully reviewing the feedback and making necessary adjustments, I can ensure that the project stays on track and that team members are able to work effectively and efficiently. This will help to ensure that the project is completed on time and within budget, while also ensuring that the team is able to deliver a high-quality product that meets the needs of stakeholders.

Task 2

Complete the writing of the user guide on the project onboarding page of confluence.

User story:

As a technical writer, I want to complete the writing of the user guide for the project onboarding page in Confluence so that team members can easily understand and navigate the project. By providing clear and concise instructions on how to use the project's features and tools, I can help team members quickly get up to speed and contribute to the project's success.

Completion status:

Finished

[Project Development Environment Setting Guide](#)

Task 3

looking for possible security issues

User story:

As a software developer, I want to proactively search for and identify any potential security issues in the project so that I can ensure that the code is secure and protected against potential threats. By conducting thorough security reviews and identifying vulnerabilities in the code, I can take appropriate measures to address them and prevent security breaches from occurring. This will help to safeguard the project and protect sensitive data and information from being compromised.

Task 4

Setting up tests

User story:

As a software developer, I want to be able to set up automated tests for the project so that I can ensure that the code is functioning correctly and catch any errors or bugs early in the development process. This will help me to improve the quality of the code and reduce the likelihood of issues arising in the future.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass [Test ActivityDataRetrieveController](#)

Track :[Progres-Sprint 2 Fix API Test](#)

Task 5

Finish the Code-peer review activity

User story:

As a software developer, I want to complete the code peer review activity for the project so that I can ensure that the code meets the required standards and is free of errors or bugs. By reviewing my colleagues' code and having my own code reviewed, I can identify areas for improvement and learn from others in the team. This will help me to improve the quality of the code and contribute to the overall success of the project.

Acceptance criteria and completion status:

Finished

[Peer-to-peer code review](#)

Task 6

Communication with Hanna about push activity problems.

User story:

As a software developer, I want to communicate with Hanna in order to solve any issues related to the push activity. By understanding the solution to specific problems from Hanna, I can ensure that the program runs smoothly and effectively for all users. This will improve the user experience and satisfaction with the program, ultimately contributing to the success of the project.

Task 7

Improve do-be-feel list &Motivational model according to the sprint1 feedback.

User story:

As a UX designer, I want to improve the do-be-feel list and motivational model based on the feedback received during sprint 1. By carefully reviewing the feedback and making necessary adjustments, I can ensure that the user experience is optimized and that users are motivated to engage with the product. This will improve user satisfaction and ultimately contribute to the success of the project.

Acceptance criteria and completion status:

Finished

Task 8

Github standard-Version control & Merge the current function branches.

User story:

As a software developer, I want to ensure that the project is following the standard GitHub version control practices and merge the current function branches. By following best practices for version control, such as maintaining a main branch for stable releases and using feature branches for development work, I can ensure that the project is well-organized and easy to manage. Additionally, by merging the current function branches, I can ensure that the latest code is integrated and that any conflicts or issues are resolved. This will improve the overall stability and reliability of the codebase, ultimately contributing to the success of the project.

Acceptance criteria and completion status:

Finished

[Project Github Repositories](#)

Task 9

Create deatiled plan page as Sprint1 feedback asked.

User story:

As a project manager, I want to create a detailed plan page as requested by the Sprint 1 feedback. By creating a detailed plan page, I can ensure that all team members have a clear understanding of project goals, timelines, and tasks. This will improve team communication and collaboration, and help to ensure that the project stays on track and within budget. This will contribute to the success of the project and ensure that all stakeholders are satisfied with the outcome.

Acceptance criteria and completion status:

Finished

[Project Planning](#)

Task 10

Fix the backend API test case

User story:

As a software developer, I want to fix the backend API test case to ensure that the API functions correctly and as expected. By fixing the test case, I can ensure that any errors or issues are identified and resolved before the software is deployed to production. This will improve the overall quality and reliability of the software application and reduce the risk of bugs or errors causing issues for users. This will contribute to the success of the project by ensuring that the software application meets the needs and expectations of users and stakeholders.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test ActivityDataRetrieveController](#)

Track: [Progess-Sprint 2 Fix API Test](#)

Task 11

Fix up document

User story:

As a team member, I want to fix up the document to ensure that it is accurate, up-to-date, and reflects the current state of the project. By fixing up the document, I can ensure that all stakeholders have access to the information they need to make informed decisions and contribute to the success of the project. This will improve communication and collaboration among team members, reduce the risk of misunderstandings or confusion, and ensure that everyone is working towards the same goals.

Acceptance criteria and completion status:

Finished

Task 12

Code cleaning & code restructuring

User story:

As a software developer, I want to clean up and restructure the code to improve its readability, maintainability, and scalability. By cleaning up and restructuring the code, I can ensure that it is well-organized, easy to understand, and follows best practices and coding standards. This will make it easier to make changes to the code in the future, reduce the risk of bugs or errors, and improve the overall quality and reliability of the software application.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test of Code Cleaning](#)

Track: [Progess-Sprint 2 Fix API Test](#)

Task 13

Add constraint rules to password setting process

User story:

As a user, I want to be able to set a strong and secure password for my account. Therefore, I want the password setting process to include constraint rules that ensure my password meets certain requirements, such as minimum length, including a mix of uppercase and lowercase letters, numbers, and special characters. By including these constraint rules, I can be confident that my account is secure and protected from potential unauthorized access. This will help to improve the overall security and integrity of the software application, and ultimately enhance the user experience and satisfaction.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test of Login/Register page](#)

Track: [Add password constraints in frontend](#)

Task 14

Solving the problem of using plain text in the user login backend

User story:

As a user, I want my login credentials to be stored in a secure and encrypted format in the backend system. Therefore, I expect the backend system to use industry-standard encryption algorithms to securely store my password in a hashed format, rather than storing it in plain text. By doing so, I can be confident that my login credentials are safe and secure, and that my personal information is protected from potential security breaches. This will help to enhance the overall security and trustworthiness of the software application, and ultimately improve the user experience and satisfaction.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the

Track: [Progress- Encrypt password](#)

Task 15

Fix the bug that same email address would cause the error instead of handling the exception.

User story:

As a user, I want the system to handle exceptions properly when the same email address is used to prevent errors, so that I can use the application without any issues.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test of Login/Register page](#)

Track: [Check Availability for Username and Email](#)

Task 16

Solve the problem of map in the activity page cannot work.

User story:

As a user, I want the map in the activity page to be fixed so that I can view my workout route and track my progress accurately.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the

Track: [Fix the map in frontend.](#)

Acceptance criteria and completion status:

Total number of completed tasks: 18

Incomplete Work:

Task 1

Add the function of disconnect Garmin Connect.

User story:

As a user, I want to be able to disconnect my Garmin Connect account from the application, so that I can stop sharing my data or switch to another account if needed. This can be achieved by adding a button or option in the settings page that allows me to disconnect my Garmin Connect account and remove any related data from the application. Additionally, there should be a confirmation message or prompt to ensure that I am aware of the consequences of this action and that I want to proceed.

Task 2

Make users be able to change to another connect account

User story:

As a user, I want to be able to disconnect my current Garmin Connect account and connect to a different one, so that I can use my preferred Garmin Connect account to track my activities on the platform. The new feature should allow me to easily disconnect my current account and provide a straightforward way to connect to another Garmin Connect account.

Task 3

Solve the problem of receiving data from Garmin connect

User story:

As a user, I want to be able to receive data from Garmin Connect so that I can easily access and analyze my fitness data without having to manually input it into the system.

Track :

[Progress-Receiving data from Garmin connect](#)

Task 4

Add reset password function

User story:

As a user, I want to be able to reset my password in case I forget it or need to update it for security reasons. By adding a reset password function, I can easily recover access to my account and ensure the security of my personal information. This will improve the user experience and increase trust in the software application. This will contribute to the success of the project by ensuring that users have a positive experience with the application and are satisfied with its functionality.

Sprint 3 related User Story

Sprint goals:

According to the needs of customers, further develop the functions of the back-end API. It is planned to solve the task last sprint left especially receiving the activity data from Garmin connect (Major task) and add the session security oauth feature into backend if time permitted.

Work:

Task 1

Add the function of disconnect Garmin Connect.

User story:

As a user, I want to be able to disconnect my Garmin Connect account from the application, so that I can stop sharing my data or switch to another account if needed. This can be achieved by adding a button or option in the settings page that allows me to disconnect my Garmin Connect account and remove any related data from the application. Additionally, there should be a confirmation message or prompt to ensure that I am aware of the consequences of this action and that I want to proceed.

Task 2

Make users be able to change to another connect account

User story:

As a user, I want to be able to disconnect my current Garmin Connect account and connect to a different one, so that I can use my preferred Garmin Connect account to track my activities on the platform. The new feature should allow me to easily disconnect my current account and provide a straightforward way to connect to another Garmin Connect account.

Task 3

Solve the problem of receiving data from Garmin connect

User story:

As a user, I want to be able to receive data from Garmin Connect so that I can easily access and analyze my fitness data without having to manually input it into the system.

Track :

[Progress-Receiving data from Garmin connect](#)

Task 4

Add reset password function

User story:

As a user, I want to be able to reset my password in case I forget it or need to update it for security reasons. By adding a reset password function, I can easily recover access to my account and ensure the security of my personal information. This will improve the user experience and increase trust in the software application. This will contribute to the success of the project by ensuring that users have a positive experience with the application and are satisfied with its functionality.

Task 5

Cloud front-end deployment

User story:

As a user, I want to access the application quickly and securely through a cloud-based front-end deployment so that I can enjoy a reliable and responsive experience regardless of my location. By deploying the application's front-end on a cloud platform, I can benefit from its global infrastructure, scalability, and performance improvements. This will contribute to the success of the project by providing a seamless user experience and ensuring that the application can handle a growing user base and increased demand.

Task 6

Address password transfer security issues

User story:

As a user, I want to ensure that my password is securely transferred between the client and the server so that my sensitive information remains protected. By addressing the password transfer security issues, I can ensure that my personal data is not compromised and that I can trust the application with my sensitive information. This will contribute to the success of the project by enhancing the overall security and reliability of the application, which will attract more users and improve their trust in the product.

Task 7

Add an email server to the backend

User story:

As a system administrator, I want to add an email server to the backend, so that the system can send emails to users for notifications and alerts. This will improve the communication with users and enhance the overall user experience. By integrating an email server into the backend, the application will become more robust and flexible, and enable users to communicate with the system in a more efficient and effective manner. This will contribute to the success of the project by improving the overall user experience and increasing user engagement with the application.

Task 8

ChatGPT code review

User story:

As a developer, I want to have my code reviewed by ChatGPT, so that I can ensure that my code is high quality and meets the company's coding standards. By having a code review process in place, I can identify and fix potential issues early on, avoid technical debt, and improve the overall codebase of the project. This will contribute to the success of the project by increasing its stability, reducing the likelihood of bugs and errors, and enabling faster and more efficient development in the long run.

Task 9

Auth issue

User story:

As a user, I want the application to solve the authentication issue so that I can access it securely and without any hassles. By addressing the authentication issue, I can have peace of mind knowing that my personal information and data are protected. This will contribute to a better user experience by ensuring that only authorized individuals can access the application and its features. It will also enhance the overall security of the system, preventing unauthorized access and potential breaches. Resolving the authentication issue will instill trust and confidence in the application, attracting more users and contributing to its success.

Development

Progress specification

Issue Type	Issue ID	Issue Title
Sprint 1		
Project Inception & Requirement Documentation	N/A	Project Background
	N/A	Personas
	N/A	Create DO-BE-FEEL list
	N/A	Motivational Mode
	N/A	Confluence Setup and Structuring
	N/A	User Stories
	N/A	Product Backlog
	N/A	Create, structure and organize the project Trello
	N/A	Github Setup and Structuring
Sprint 2		
Basic Development Setup	#2	Initialize frontend
	N/A	deploy MongoDB database and connect to our server
	N/A	deploy backend connect to our server
	N/A	Deploy frontend locally and connect to our sever backend
	N/A	Make a note of any things not working
	N/A	Complete the writing of the user usage rules on the project onboarding page on confluence.
Reorganize the project and solve previous project left security problems and bugs	#3	Do code cleaning for entire project
	N/A	Solve the problem of map in the activity page cannot work.
	N/A	Fix the bug that same email address would cause the error instead of handling the exception.
	N/A	Solve the problem of using plain text in the user login backend
	N/A	Solve the problem of activity Url setting
	N/A	Receiving data from Garmin connect
	N/A	Sprint 2 Fix API Test
Functional Development	#4	Change Garmin connect account
	N/A	Add password constraints in frontend
Sprint 3		
Security improvement	#5	Fix transferring security issue
	N/A	Integrate with Spring Security
	N/A	Add SMTP email server to the backend
System fuction improvement	#6	Receiving data from Garmin connect
	N/A	Disconnect Garmin Connect
	N/A	Change Garmin connect account
	N/A	Reset password function
Product depolymtent	#7	Cloud front-end deployment
Sprint 4		

Code cleaning

Progress description:

At the request of the customer, code clean the entire project.

Customer requirements connected:

- Requirements 1(Code cleaning)

Related Test:

- Test of Code Cleaning

Progress time

Sprint 2 week1

Progress action

1. Perform code cleaning operations on the code under the GA-BoxJelly/src/cm-backend/src/main/java/coachingmateanalytics/**coachingmate**/ directory file which include:

- Remove duplicate import statements.
- Remove useless commented code.
- Optimize code structure to make it more readable.
- Add appropriate code comments.

Progress result:

..		
common	code clean for AuthCheck.java	now
controller	code clean & restructure for RegistryController.java	last week
dao	code clean for UserDaoImpl.java	5 days ago
entity	code clean for UserPartner.java	5 days ago
service	code clean for UserServiceImpl.java	5 days ago
CoachingmateApplication.java	code clean for CoachingmateApplication.java	3 minutes ago

[Give feedback](#)

..		
ActivityDataRetrieveController.java	Simplification refactoring methods @RequestMapping	last week
GarminPushController.java	code clean for GarminPushController	last week
LoginController.java	code clean for LoginController.java	last week
OAuthController.java	code clean for OAuthController.java	last week
RegistryController.java	code clean & restructure for RegistryController.java	last week

[Give feedback](#)

..		
ActivityDao.java		
ActivityDaoImpl.java		
DashboardStatisticsDao.java		
TokenDao.java		
TokenProviderDao.java		
UserDao.java		
UserDaoImpl.java		

[Give feedback](#)

..		
Activity.java	code clean for Activity.java	5 days ago
ActivityBean.java	code clean for ActivityBean.java	5 days ago
ActivityDetails.java	code clean for ActivityDetails.java	5 days ago
ActivityDetailsBean.java	code clean for ActivityDetailsBean.java	5 days ago
ActivityFile.java	code clean for ActivityFile.java	5 days ago
Athlete.java	code clean for Athlete.java	5 days ago
Badge.java	code clean for Badge.java	5 days ago
Coach.java	code clean for Coach.java	5 days ago
DashboardStastic.java	code clean for DashboardStastic.java	5 days ago
Exercise.java	code clean for Exercise.java	5 days ago
Group.java	code clean for Group.java	5 days ago
Laps.java	code clean for Laps.java	5 days ago
Phase.java	code clean for Phase.java	5 days ago
Planner.java	code clean for Planner.java	5 days ago
Program.java	code clean for Program.java	5 days ago
Program_status.java	code clean for Program_status.java	5 days ago
Record.java	code clean for Record.java	5 days ago
RequestToken.java	code clean for RequestToken.java	5 days ago
Samples.java	code clean for Samples.java	5 days ago
Session.java	code clean for Session.java	5 days ago
Statistic.java	existing code	2 weeks ago
Summary.java	code clean for Summary.java	5 days ago
UserPartner.java	code clean for UserPartner.java	5 days ago

[Give feedback](#)

..		
serviceImpl	code clean for UserServiceImpl.java	5 days ago
ActivityService.java	code clean for ActivityService.java	5 days ago
EmailService.java	no change for EmailService.java resolve it later	5 days ago
HttpRequestAdapter.java	code clean for HttpRequestAdapter.java	5 days ago
OAuthService.java	code clean for OAuthService.java	5 days ago
OauthConfiguration.java	code clean for OauthConfiguration.java	5 days ago
UserService.java	code clean for UserService.java	5 days ago

[Give feedback](#)

	ActivityServiceImpl.java	code clean for ActivityServiceImpl.java	5 days ago
	UserServiceImpl.java	code clean for UserServiceImpl.java	5 days ago

[Give feedback](#)

Code cleaning work Sample

Showing 1 changed file with 28 additions and 28 deletions.

Split Unified

```

35      private static final Logger logger = LoggerFactory.getLogger(LoginController.class);
36
37      @@ -40,42 +38,44 @@ public class LoginController {
38          @Resource
39          ActivityServiceImpl activityService;
40
41          /**
42          * User login
43          *
44          * @param username User's username
45          * @param password User's password
46          * @return UserPartner object
47          */
48          @RequestMapping(value = "/login", method = RequestMethod.POST, produces = "application/json")
49          @ApiOperation(value = "login interface", notes = "user login")
50          public UserPartner login(@RequestParam(required = true, type = "String") String username,
51                                  @RequestParam(required = true, type = "String") String password) {
52              UserPartner userPartner = userService.loginCheck(username, password);
53              if(userPartner == null) throw new BusinessException(ResponseCode.USER_IS_NOT_EXISTED);
54              userPartner.setPassword("");
55              if(userPartner.getUserAccessToken() != null)
56                  activityService.statisticsActivities(userPartner.getUserAccessToken());
57
58              return userPartner;
59          }
60          /**
61          * @param param
62          * @return
63          * @Description: get user by token
64          */
65          /**
66          * Get user by token
67          */
68          @RequestMapping(value = "/user", method = RequestMethod.GET, produces = "application/json")
69          @ApiOperation(value = "Get user by token", notes = "user login")
70          public UserPartner getUser(@RequestParam Map<String, Object> params) {
71              String token = (String) params.get("token");
72              UserPartner userPartner = userService.getUserByToken(token);
73              if(userPartner == null) throw new BusinessException(ResponseCode.USER_IS_NOT_EXISTED);
74              userPartner.setPassword("");
75
76              return userPartner;
77          }
78      }

```

Check Availability for Username and Email

Progress description:

At the request of the customer, CoachingMate system needs to alert users if the username and the email they use to register are available.

Customer requirements connected:

- Requirements 3 (Solve previous project left bugs)

Related Test:

[Test of Login/Register page](#)

Progress time

Sprint 2 week2

Progress action

1. Usernames and Emails that new users use to register will be inserted into the database. If any of them already exists in the database, the backend will throw an exception and pass it to the frontend. Instead of displaying a new page with error, as the original code does, we capture the exception and pops up the alert to inform new users the reason why they failed to register.

- a. To do this, we simply capture the exception in the fronted (src/cm-frontend-react/src/views/register/index.jsx):

```
30     register(params).then((data) => {
31       message.success("Register Success!");
32       setLoading(false);
33       history.push("/login")
34     })
35     .catch((error) => {
36       setLoading(false);
37       //           message.error(error);
38       message.error("Username or Email already exists!");
39     });
40   };
41 }
```

Progress result:

 Username or Email already exists!

User Register

 example

 example

 example@qq.com

Register

Back

Receiving data from Garmin connect

Progress description:

At the request of the customer, CoachingMate system needs to be able to receive user activity data from Garmin connect.

Customer requirements connected:

- Requirements 7(Ensure accessibility)

Related Test:

[Test Receiving Data from Garmin](#)

Progress time

Sprint 2 week2

Progress action

1. Deploy backend remotely and configure a https hostname in order to receive data from Gramin connect:
 - a. Run backend on a cloud server and bind 0.0.0.0
 - b. Reversely proxied using web sever tool Apache
 - c. Configure DNS settings
2. Configure endpoints with the domain and paths
3. Connect to a Garmin user and try to fetch data

Progress Track

Currently, this task is blocked and cannot be completed.

So research on this issue:

1. Using the postman to simulate the data receiving process locally to test the system code.
 - ***No problem found.***
1. Already contact supervisor Paul to check the Garmin connect developer API management console ensure developer API works.
 - ***Paul replied no log found in Garmin connect developer API management console.***

As Edurado suggested, for the receiving the Garmin connect pushed activcty data problems, we should contact Hanna for furthur information.

LINGKANG is trying to communicate with Hanna about push activcty problems.

Potential causes of this problem will be discussed by the team and Hanna at the client meeting.

Current stage:

The team has discussed possible potential problems with Hanna in the client meeting based on the information of the sever log. Will be further resolved in Sprint 3.

Progress result:

Add password constraints in frontend

Progress description:

- Requirements 4Solve previous project left security problems

Related Test:

Test of Login/Register page

Progress time

Sprint 2 week1

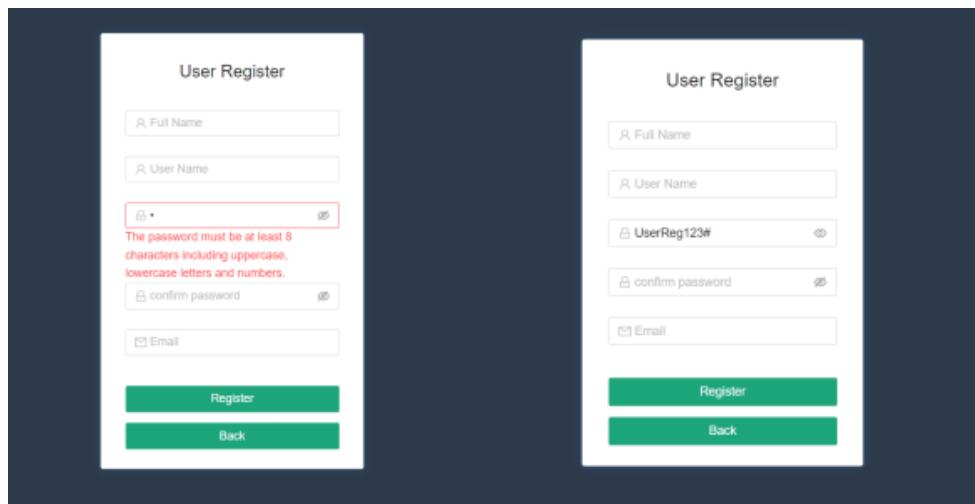
Progress action

Add validation function in validation.js

```
export function passwordValidate(str){  
    let reg = /^(?=.*\d)(?=.*[A-Z])(?=.*[a-z])(?=.*[\x21-\x2f\x3a-\x40\x5b-\x60\x7b-\x7f])[\da-zA-Z\x21-\x2f\x3a-\x40\x5b-\x60\x7b-\x7f]*$/;  
    return reg.test(str)  
}
```

Import the function into the register page, and validate the text from the password field, return invalid message until the password is ok to be used

```
<Form.Item name="password" rules={[{  
    required: true,  
    whitespace: true,  
    message: "Please input password.",  
,  
    ({ getFieldValue }) => ({  
        validator(_, value) {  
            if (!passwordValidate(getFieldValue('password'))) {return Promise.resolve();}  
            return Promise.reject(new Error('The password must be at least 8 characters including uppercase, lowercase letters and numbers.'));  
        },  
    }  
    >  
    <Input.Password prefix={  
        <LockOutlined style={{ color: "rgba(0,0,0,.25)" }} />  
    }/></Form.Item>
```



Encrypt password

Progress description:

The previous code didn't encrypt the password which may lead a security issue. Hence, we use spring security to encrypt the password to cipher text.

Customer requirements connected:

[Requirements 4](#)Solve previous project left security problems

Related Test:

[Test LoginController](#)

Progress time

Sprint 2 week1

Progress action

Use Sprint security crypto library to encrypt the password.

1. add library into pom file

```
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-crypto</artifactId>
    <version>5.7.3</version>
</dependency>
```

The newer version may not match the project jdk or other library.

2. In RegistryController, use library to encode the password so that the encryption password could be inserted into the database.

```
BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
UserPartner user = userService.register(fullname,username, passwordEncoder.encode(password), email);
```

main steps in source code:

```

private static String hashpw(byte[] passwordb, String salt, boolean for_check) {
    char minor = 0;
    StringBuilder rs = new StringBuilder();
    if (salt == null) {
        throw new IllegalArgumentException("salt cannot be null");
    } else {
        int saltLength = salt.length();
        if (saltLength < 28) {
            throw new IllegalArgumentException("Invalid salt");
        } else if (salt.charAt(0) == '$' && salt.charAt(1) == '2') {
            byte off;
            if (salt.charAt(2) == '$') {
                off = 3;
            } else {
                minor = salt.charAt(2);
                if (minor != 'a' && minor != 'x' && minor != 'y' && minor != 'b' || salt.charAt(3) != '$') {
                    throw new IllegalArgumentException("Invalid salt revision");
                }
            }
            off = 4;
        }

        if (salt.charAt(off + 2) > '$') {
            throw new IllegalArgumentException("Missing salt rounds");
        } else if (off == 4 && saltLength < 29) {
            throw new IllegalArgumentException("Invalid salt");
        } else {
            int rounds = Integer.parseInt(salt.substring(off, off + 2));
            String real_salt = salt.substring(off + 3, off + 25);

            String real_salt = salt.substring(off + 3, off + 25);
            byte[] saltb = decode_base64(real_salt, maxlen: 16);
            if (minor >= 'a') {
                passwordb = Arrays.copyOf(passwordb, newLength: passwordb.length + 1);
            }

            BCrypt B = new BCrypt();
            byte[] hashed = B.crypt_raw(passwordb, saltb, rounds, sign_ext_bug: minor == 'x', minor == 'a' ? 65536 : 0, t);
            rs.append("$2");
            if (minor >= 'a') {
                rs.append(minor);
            }

            rs.append("$");
            if (rounds < 10) {
                rs.append("0");
            }

            rs.append(rounds);
            rs.append("$");
            encode_base64(saltb, saltb.length, rs);
            encode_base64(hashed, len: bf_crypt_ciphertext.length * 4 - 1, rs);
            return rs.toString();
        }
    } else {
        throw new IllegalArgumentException("Invalid salt version");
    }
}

```

3. In LoginController, the password user inputted should be compared with the encryption password. The Algorithm would first get the salt from the hashed password. The salt is part of hashed value. The encryption round is also from the hashed password. When we have the parameter and raw password, we can encrypt it again. Then we can compare the 'new' encrypted password with the encryption in database to login.

Fix the map in frontend.

Progress description:

Customer requirements connected:

- Requirements 3(Solve previous project left bugs)

Related Test:

[Test ActivityDaoImpl](#)

Progress time

Sprint 2 week1

Progress action

- At first, the map cannot show in each activity. After we debug, we find that the first thing we should do is change the google-map-react API. So we register a new google map api and change the API key in the previous code.

The screenshot shows the Google Cloud Platform API credentials dashboard. Under the '凭据' tab, it displays an OAuth 2.0 Client ID and an API key. The API key is highlighted with a yellow box. The URL for the screenshot is <https://console.cloud.google.com/apis/dashboard>.

(<https://console.cloud.google.com/apis/dashboard>)

- We also find the given data don't have the location information. (Using the Garmin Api Data generator). After we check code again, we found we should add the latitudeInDegreee and longitudeInDegree to each samples in ActivityDetail.

- After we update the config, we cannot still open the map. The we found perhaps the loading part has problem. When we change come loading

The screenshot shows a code editor with a portion of a React component. It includes a loading state and a GoogleMapReact component. The code is as follows:

```
143 // Important: Always set the container height explicitly
144 <div>
145   <div style={{ height: '600px', width: '100%', display: this.state.loading ? 'block' : 'none' }}>
146     <LoadingOutlined className="loading-style" spin />
147   </div>
148   <div style={{ height: '600px', width: '100%', display: this.state.loading ? 'none' : 'block' }}>
149     <GoogleMapReact
150       bootstrapURLKeys={{ key: "AIzaSyAAityg-sL70gj5yrBTxdLJvNhFrWY_6H0" }}
151       defaultCenter={{
152         lat: -37.813,
153         lng: 144.96
154       }}
155       defaultZoom={14}
156       center={this.state.center}
157       yesIWantToUseGoogleMapApiInternals
158       onGoogleApiLoaded={({ map, maps }) => apiIsLoaded(map, maps)}
159     >
160     </GoogleMapReact>
161   </div>
162   <img src={ img } style={{ marginLeft: '38%', display: this.state.loading ? 'none' : 'block' }} />
</div>
```

code, the map shows.
lines.

delete these 3

Data samples(json): (userAccessToken and activityId and userId need to change)

```
[  
{  
"summary": {  
"durationInSeconds": 10,  
"averageSpeedInMetersPerSecond": 1.362,  
"averageHeartRateInBeatsPerMinute": 88,  
"distanceInMeters": 13.39,  
"activityName": "Running",  
"deviceName": "forerunner935",  
"steps": 4,  
"averageRunCadenceInStepsPerMinute": 25.09375,  
"averagePaceInMinutesPerKilometer": 12.236906,  
"activityId": 10,  
"startTimelnSeconds": 1650195177,  
"startTimeOffsetInSeconds": 36000,  
"maxPaceInMinutesPerKilometer": 4.8676014,  
"maxHeartRateInBeatsPerMinute": 93,  
"maxRunCadenceInStepsPerMinute": 138,  
"maxSpeedInMetersPerSecond": 3.424,  
"activityType": "RUNNING"  
},  
"activityId": 10,  
"userAccessToken": "bd5b13a9-9eed-4204-a807-94bf0eb06356",  
"summaryId": "10-detail",  
"laps": [  
{  
"startTimelnSeconds": 1650195177  
}  
],  
"userId": "898329394",  
"samples": [  
{  
"startTimelnSeconds": 1650195177,  
"timerDurationInSeconds": 0,  
"movingDurationInSeconds": 0,  
"heartRate": 84,  
"elevationInMeters": 33.20000076293945,  
"speedMetersPerSecond": 0,  
"clockDurationInSeconds": 0,  
"stepsPerMinute": 0,  
"airTemperatureCelcius": 32,  
"totalDistanceInMeters": 0,  
"latitudeInDegree": -37.813,  
"longitudeInDegree": 144.96  
},  
{  
"startTimelnSeconds": 1650195178,  
"timerDurationInSeconds": 1,  
"movingDurationInSeconds": 0,  
"heartRate": 86,  
"elevationInMeters": 33.20000076293945,  
"speedMetersPerSecond": 0,  
"clockDurationInSeconds": 1,  
"stepsPerMinute": 0,  
"airTemperatureCelcius": 32,  
"totalDistanceInMeters": 0,  
"latitudeInDegree": -37.815,  
"longitudeInDegree": 144.95  
},  
{  
"startTimelnSeconds": 1650195182,  
"timerDurationInSeconds": 5,  
"movingDurationInSeconds": 0,  
"heartRate": 129,  
"elevationInMeters": 31.799999237060547,  
"speedMetersPerSecond": 0,  
"clockDurationInSeconds": 5,  
"stepsPerMinute": 0,  
"airTemperatureCelcius": 32,  
"totalDistanceInMeters": 0,  
"latitudeInDegree": -37.816,  
"longitudeInDegree": 144.94  
},  
{  
"startTimelnSeconds": 1650195185,  
"timerDurationInSeconds": 8,  
"movingDurationInSeconds": 3,  
"heartRate": 129,  
"elevationInMeters": 31,
```

```
"speedMetersPerSecond": 3.4240000247955322,  
"clockDurationInSeconds": 8,  
"stepsPerMinute": 0,  
"airTemperatureCelcius": 32,  
"totalDistanceInMeters": 6.539999961853027,  
"latitudeInDegree": -37.817,  
"longitudeInDegree":144.93  
},  
{  
"startTimelnSeconds": 1650195187,  
"timerDurationInSeconds": 10,  
"movingDurationInSeconds": 5,  
"heartRate": 112,  
"elevationInMeters": 31,  
"speedMetersPerSecond": 3.4240000247955322,  
"clockDurationInSeconds": 10,  
"stepsPerMinute": 0,  
"airTemperatureCelcius": 32,  
"totalDistanceInMeters": 13.390000343322754,  
"latitudeInDegree": -37.818,  
"longitudeInDegree":144.92  
}  
]  
}  
]
```

Sprint 2 Fix API Test

Progress descriptionL

As a project developers, all APIs should works well.

Requirements connected:

[Requirments 5 \(Test in sprint 2\)](#)

Progress time:

Complete all by Sprint 2.

Progress action:

Write all API Junit test case in test folder.

- [Test ActivityDaolmpl](#)
- [Test ActivityDataRetrieveController](#)
- [Test DashboardStatisticsDao](#)
- [Test generateRequestTokenSecret](#)
- [Test ActivityServiceImpl](#)
- [Test UserServicempl](#)
- [Test LoginController](#)
- [Test of Code Cleaning](#)
- [Test of Login/Register page](#)
- [Test RegistryController](#)
- [Test TokenDao](#)
- [Test UserDao](#)

Fix transferring security issue

Progress description:

The user account information shouldn't be exposed in url otherwise others can easily get his private information, so we fix this security issue.

Customer requirements connected:

Requirements 12(Fix transferring_issue)

Related Test:

Test fixingt ransferring issue

Progress time

Sprint 3 week1

Progress action

Before fixing: Others can easily see user's username and password once he register and login

The screenshot shows a NetworkMiner capture window. In the left pane, there are several network requests listed. One request, highlighted in blue, is a POST request to the URL `https://coach.myyylxy.com/register?fullname=0505&username=0505&password=AAaa11..`. In the right pane, under the 'General' tab, the 'Request URL' field displays the full URL including the password `password=AAaa11..`. Other details shown include the Request Method (POST), Status Code (200), and Remote Address (104.21.15.96:443). Under the 'Response Headers' tab, various CORS-related headers are listed, such as `access-control-allow-credentials: True`, `access-control-allow-headers: Origin, X-Requested-With, Content-Type, Accept, x-token, User, Us, token, Authorization`, and `access-control-allow-methods: *`.

This screenshot shows the same NetworkMiner capture window after the fix has been applied. The registered user's credentials are no longer visible in the URL. The highlighted POST request now shows the URL as `https://coach.myyylxy.com/login?username=0505&password=AAaa11..`. The 'General' tab in the right pane still shows the original URL with the password, but this is likely a cached or displayed state rather than the actual transmitted URL. The rest of the details (Request Method, Status Code, Headers) remain the same as in the previous screenshot.

After fixing: the information will not show in url

register

register

General

Request URL: http://localhost:8080/register
Request Method: POST
Status Code: 200
Remote Address: [::1]:8080
Referrer Policy: strict-origin-when-cross-origin

Response Headers

View source

login

login

getUserByToken

getUserByToken

getUserByToken

getUserByToken

getUserByToken

0e03b7da-db9e-4819-ba10-9016ddfdad3

getUserByToken

General

Request URL: http://localhost:8080/login
Request Method: OPTIONS
Status Code: 200
Remote Address: [::1]:8080
Referrer Policy: strict-origin-when-cross-origin

Response Headers

View source

Cloud front-end deployment

Progress description:

At the request of the customer, deploy the front-end on Vercel

Customer requirements connected:

[Requirements 11\(Cloud front-end deployment\)](#)

Related Test:

[Test of Cloud front-end deployment](#)

Progress time

Sprint 3 week1

Progress action

Import front-end code into Vercel from Github Repositories

Let's build something new.

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

The screenshot shows the Vercel dashboard. On the left, the 'Import Git Repository' section displays a list of repositories from the user 'KevinZhouk'. One repository, 'COMP90082-2023-SM1', is selected and highlighted with a green border. To the right of the list are four 'Import' buttons. Below the list is a link to 'Import Third-Party Git Repository'. On the right side of the dashboard, the 'Clone Template' section is visible, featuring cards for various front-end frameworks: NEXT.js, Next.js, SvelteKit, Nuxt.js, and Vite. Each card includes a preview image and a brief description.

Select right Github Repositories

Import Git Repository

COMP90082-2023-SM1

- GA-BoxJelly · 14d ago Import
- PR-Bluering · 10d ago Import
- comp90082-2023-sm1-resources · 50d ago Import
- FA-Web-App · 48d ago Import

Import Third-Party Git Repository →

Clone Template

- Next.js
- SvelteKit
- Nuxt.js
- Vite

Browse All Templates →

Select the source code folder

The screenshot shows a project configuration interface. On the left, there's a sidebar with 'Configure Project' selected. Below it, under 'GIT REPOSITORY', is a list of repositories: 'COMP90082-2023-SM1/GA-BoxJelly' (selected), 'main', and a placeholder '...' for a local directory. At the bottom of this sidebar are links to 'Import a different Git Repository' and 'Browse Templates'.

The main area is titled 'Configure Project'. A modal window is open, titled 'Root Directory'. It contains a tree view of the repository structure:

```

    Root Directory
    └── GA-BoxJelly
        ├── data_samples
        ├── docs
        ├── prototypes
        └── src
            ├── cm-backend
            └── cm-frontend-react

```

The 'cm-frontend-react' folder is highlighted with a red box. At the bottom of the modal are 'Cancel' and 'Continue' buttons. In the background, a progress bar indicates 'Preparing deployment...' with steps 'Building' and 'Deployment Summary'.

Change the build command to `npm run build`

Change the install Command to `npm install`

Project Name
ga-box-jelly-zub9

Framework Preset
Create React App

Root Directory
src/cm-frontend-react Edit

Build and Output Settings

Build Command ?
npm run build OVERRIDE

Output Directory ?
build OVERRIDE

Install Command ?
npm install OVERRIDE

> Environment Variables

Deploy

The screenshot shows a deployment configuration interface. It includes fields for Project Name (ga-box-jelly-zub9), Framework Preset (Create React App), and Root Directory (src/cm-frontend-react). A 'Build and Output Settings' section is expanded, showing 'Build Command' (npm run build) with an 'OVERRIDE' toggle set to 'ON'. Below it is 'Output Directory' (build) with an 'OVERRIDE' toggle set to 'OFF'. Another expanded section shows 'Install Command' (npm install) with an 'OVERRIDE' toggle set to 'ON'. At the bottom is a large black 'Deploy' button.

Click the Deploy button.

Progress output:

Project Product Demo

Demo Url: <https://ga-box-jelly-ddbh.vercel.app/#/>

Example page:

Login Form

User Name

Password

Reset password function

Progress description:

At the request of the customer, realize the function that the user can change the password independently.

Customer requirements connected:

[Requirements 10\(Reset password function\)](#)

Related Test:

[Test Reset Password Function](#)

Progress time

Sprint 3 week1

Progress action

1. Add email template for resetting password.

```
public void sendResetEmail(String toEmail) throws UnsupportedEncodingException {
    String from = "g.lee@yqfqzmy.monster";
    SimpleMailMessage message = new SimpleMailMessage();
    message.setFrom(from);
    message.setTo(toEmail);
    message.setSubject("CoachingMate: Reset Password");
    String codeString = Vercode.generateString();
    VercodeTime.stringCodeMap.put(toEmail, codeString);
    System.out.println(codeString);
    VercodeTime.currentTimeMapString.put(toEmail, new Date().getTime());

    String url = Consts.FRONT_URL+ "requestReset/reset/?";
    String para = "email="+toEmail+"&security="+codeString;
    String encodedQuery = Base64.getEncoder().encodeToString(para.getBytes());
    String encodedQuery = URLEncoder.encode(para, StandardCharsets.UTF_8.toString());
    String fullurl = url+encodedQuery+"#/reset";

    message.setText("You've requested to reset your password:\n"
        + "\nTo continue please click this link:\n"+ fullurl
        + "\nIf you did not make this request, please disregard this email.\n"+ "(This is a test message from CoachingMate.)");

    mailSender.send(message);
}
```

2. Add maps to temporarily store generated string, time, and emails.

```
2 usages
public static HashMap<String,Long> currentTimeMapString = new HashMap<>();

3 usages
public static HashMap <String, String> stringCodeMap = new HashMap<>();
```

3. Add a function to secure the reset password process.

```

public static final String generateString(){
    int leftLimit = 97;
    int rightLimit = 122;
    int targetStringLength = 10;
    Random random = new Random();
    StringBuilder buffer = new StringBuilder(targetStringLength);
    for (int i = 0; i < targetStringLength; i++) {
        int randomLimitedInt = leftLimit + (int)
            (random.nextFloat() * (rightLimit - leftLimit + 1));
        buffer.append((char) randomLimitedInt);
    }
    String generatedString = buffer.toString();
    return generatedString;
}

```

4. Map requests to functions.

```

@RequestMapping(value = "/requestReset/reset", method = RequestMethod.POST)
@ApiOperation(value = "reset password", notes = "reset password")
public ResponseEntity reset(@ApiParam(required = true, type = "String") String email, @ApiParam(required = true, type = "String")
    if (Objects.equals(oldpassword, newPassword)) return ResponseEntity.ok("Same password error");
    UserPartner userObject = JSONObject.parseObject(email, UserPartner.class);
    email = userObject.getEmail();

    Date current = new Date();
    boolean a = (Objects.equals(String.valueOf(VercodeTime.stringCodeMap.get(email)), security.toString()));
    boolean b = (VercodeTime.currentTimeMapString.get(email) - current.getTime() <= 1800000);
    System.out.println(VercodeTime.stringCodeMap.get(email).toString());
    System.out.println(security.toString());
    System.out.println(a);
    System.out.println(b);

    if (a&&b){
        UserPartner userPartner = userService.getUserByEmail(email);
        UserPartner userObject1 = JSONObject.parseObject(password, UserPartner.class);
        password = userObject1.getPassword();
        System.out.println(password);
        String username = userPartner.getUsername();
        userPartner = userService.loginCheck(username, oldpassword);
        if (userPartner == null) return ResponseEntity.ok("Wrong password provided");
        Query query = new Query(Criteria.where("username").is(username));
        BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
        Update update = Update.update("password", passwordEncoder.encode(password));
        UpdateResult result = mongoTemplate.updateMulti(query, update, UserPartner.class, Consts.MONGODB_USER_COLLECTIN_NAME);
        System.out.println(result);
        return ResponseEntity.ok("Request success");
    }
    else return ResponseEntity.status(HttpStatus.CONFLICT).body("security code expired or incorrect");
}

```

Users can now click reset password in the frontend and use the emails they receive to reset the password.

Change Garmin connect account

Progress description:

At the request of the customer, realize the function that users can autonomously switch the bound Garmin Connect account.

Customer requirements connected:

[Requirements 9\(Change Garmin connect account\)](#)

Related Test:

[Test Disconnect and Change Garmin Account](#)

Progress time

Sprint 3 week1

Progress action

As per [Disconnect Garmin Connect](#), use another account to link to coachingmate to switch account after disconnecting from coachingmate using the current account.

Disconnect Garmin Connect

Progress description:

At the request of the customer, the garmin account can be disconnect from coachingmate.

Customer requirements connected:

[Requirements 8\(Disconnect Garmin Connect\)](#)

Related Test:

[Test Disconnect and Change Garmin Account](#)

Progress time

Sprint 3 week1

Progress action

1. Add deleteAccessToken in TokenDao.java

```
3 usages ▲ Geralt1209
public void deleteAccessToken(String username) {
    Query query = Query.query(Criteria.where("username").is(username));
    Update update = new Update().unset("userAccessToken").unset("userAccessSecret");
    mongoTemplate.updateMulti(query, update, UserPartner.class, Consts.MONGODB_USER_COLLECTIN_NAME);
}
```

2. Add revokeAccessToken in OAuthService.java

```
1 usage ▲ Geralt1209
public void revokeAccessToken(String accessToken) {
    RequestToken token = tokenDao.findByToken(accessToken); // find access token by token
    if (token != null) { // if token exists
        tokenDao.deleteAccessToken(token.getUsername()); // delete access token by username
    } else {
        logger.error("Access token not found: {}", accessToken);
    }
}
```

3. Map requests to functions

```
@PostMapping("/disconnect")
@ApiOperation(value = "Revoke access token and disconnect from Garmin Connect", notes = "Revokes the access token for the user and disconnects them from the system")
public ResponseEntity<String> disconnect(@ApiParam(required = true, type = "String") @RequestParam String username) {
    UserDaoImpl userdaoimpl = new UserDaoImpl();
    UserPartner userPartner = userService.getUser(username);
    if (userPartner.getUserAccessToken() == null) {
        return ResponseEntity.badRequest().body("User " + username + " has not been authorized.");
    } else {
        OAuthService oAuthService = new OAuthService();
        if (userService.tokenCheck(userPartner.getUserAccessToken())) {
            oAuthService.revokeAccessToken(userPartner.getUserAccessToken());
            userdaoimpl.selectUserByToken(null);
            userdaoimpl.saveUser(userPartner);
        }
        else {
            tokenDao.deleteAccessToken(username);
        }
        return ResponseEntity.ok().body("User " + username + " has been disconnected.");
    }
}
```

Users can click disconnect their garmin connect accounts from coachingmate account by clicking the disconnect button in the user interface.

Add SMTP email server to the backend

Progress description:

At the request of the customer, deploy the SMTP email server to prepare for the implementation of subsequent user validation purpose.

Customer requirements connected:

[Requirements 13 \(Add SMTP email server to the backend\)](#)

Related Test:

[Test SMTP Service](#)

Progress time

Sprint 3 week1

Progress action

1. Add spring-boot-starter-mail and javax.mail and remove jakarta.mail into pom file to enable mail service as well as resolving conflicts.

```
119 -      <dependency>
120 -        <groupId>com.sun.mail</groupId>
121 -        <artifactId>jakarta.mail</artifactId>
122 -        <version>2.0.1</version>
123 -      </dependency>
124 119      <dependency>
125 120        <groupId>org.springframework.boot</groupId>
126 121        <artifactId>spring-boot-starter-test</artifactId>
193 188      <!--          <version>7.0</version>-->
194 189      <!--          </dependency>-->
195 190
191 +      <dependency>
192 +        <groupId>org.springframework.boot</groupId>
193 +        <artifactId>spring-boot-starter-mail</artifactId>
194 +      </dependency>
195 +      <dependency>
196 +        <groupId>javax.mail</groupId>
197 +        <artifactId>mail</artifactId>
198 +        <version>1.4.7</version>
199 +      </dependency>
200 +
```

2. Configure SMTP service in properties file.

```

spring.mail.host=smtp.mail.ru
spring.mail.port=465
spring.mail.username=g.lee@yqfqzmy.monster
spring.mail.password=DKyCLaw0trnScsi7u3cY
spring.mail.default-encoding=UTF-8
spring.mail.properties.mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
spring.mail.properties.mail.debug=true

```

3. Add email template.

```

public void sendResetEmail(String toEmail) throws UnsupportedEncodingException {
    String from = "g.lee@yqfqzmy.monster";
    SimpleMailMessage message = new SimpleMailMessage();
    message.setFrom(from);
    message.setTo(toEmail);
    message.setSubject("CoachingMate: Reset Password");
    String codeString = Vercode.generateString();

```

4. Map requests to functions.

```

@RequestMapping(value = "/requestReset/reset", method = RequestMethod.POST)
@ApiOperation(value = "reset password", notes = "reset password")
public ResponseEntity reset(@ApiParam(required = true, type = "String") String email, @ApiParam(required = true, type = "String") String oldpassword, @ApiParam(required = true, type = "String") String newPassword) {
    if (Objects.equals(oldpassword, newPassword)) return ResponseEntity.ok("Same password error");
    UserPartner userObject = JSONObject.parseObject(email, UserPartner.class);
    email = userObject.getEmail();

    Date current = new Date();
    boolean a = (Objects.equals(String.valueOf(VercodeTime.stringCodeMap.get(email)), security.toString()));
    boolean b = (VercodeTime.currentTimeMapString.get(email) - current.getTime() <= 18000000);
    System.out.println(VercodeTime.stringCodeMap.get(email).toString());
    System.out.println(security.toString());
    System.out.println(a);
    System.out.println(b);

    if (a&&b){
        UserPartner userPartner = userService.getUserByEmail(email);
        UserPartner userObject1 = JSONObject.parseObject(password, UserPartner.class);
        password = userObject1.getPassword();
        System.out.println(password);
        String username = userPartner.getUsername();
        userPartner = userService.loginCheck(username, oldpassword);
        if (userPartner == null) return ResponseEntity.ok("Wrong password provided");
        Query query = new Query(Criteria.where("username").is(username));
        BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
        Update update = Update.update("password", passwordEncoder.encode(password));
        UpdateResult result = mongoTemplate.updateMulti(query, update, UserPartner.class, Consts.MONGODB_USER_COLLECTION_NAME);
        System.out.println(result);
        return ResponseEntity.ok("Request success");
    }
    else return ResponseEntity.status(HttpStatus.CONFLICT).body("security code expired or incorrect");
}

```

5. Add verification method

a. Create maps to temporarily store verification code, verification time, and emails used.

```
public static HashMap<String, Long> currentTimeMap = new HashMap<>();  
3 usages  
public static HashMap <String, String> verCodeMap = new HashMap<>();
```

b. Create verification method.

```
public static final String generateVerCode(){  
    char [] numbers = new char[6];  
    for (int i=0;i<numbers.length;i++){  
        numbers[i] = SYMBOLS.charAt(RANDOM.nextInt(SYMBOLS.length()));  
    }  
    return new String(numbers);  
}
```

The email service is then set up and ready to use. Newly registered users can receive emails to verify their email addresses.

Integrate with Spring Security

Progress description:

The previous code allow everyone to use backend API. In real situation, if someone want to get the information from website, he must login. We use Spring Security and JWT to implement this functionality.

Customer requirements connected:

[Requirements 14\(Integrationw with SpringSecurity\)](#)

Related Test:

[Test Spring Security Integration](#)

Progress time

Sprint 3 week3

Progress action

Pervious: When we have accessToken or other parameter, we can call backend API to get private information without anything.

POST http://localhost:8080/activity/getActivityByAccessToken?accessToken=2d7c25b2-27f5-4974-a82f-2a9e41502b9f

Key	Value	Description
Key	Value	Description

```
1 "statusCode": 200,
2 "message": "Success",
3 "data": [
4   {
5     "parentSummaryId": null,
6     "userId": "898329394",
7     "userAccessToken": "2d7c25b2-27f5-4974-a82f-2a9e41502b9f",
8     "summaryId": "8654408419",
9     "activityId": 10,
10    "activityName": "Running",
11    "durationInSeconds": 10,
12    "startTimeInSeconds": 1650191523,
13    "startTimeOffsetInSeconds": 36000,
14    "activityType": "RUNNING",
15    "averageHeartRateInBeatsPerMinute": 92,
```

Now: After Backend assign user a JWT and this JWT doesn't expire validated date, then user have ability to get information by backend API. Otherwise, it will show no authority.

Before login: no JWT header

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

[Create Collection](#) or [Use Collection Template](#)

After Login and JWT doesn't expire validated date:

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

[Create Collection](#) or [Use Collection Template](#)

JWT expire validated date:

The screenshot shows the Postman application interface. At the top, there are navigation links: Home, Workspaces, API Network, and Explore. A search bar says "Search Postman". On the right, there are buttons for Invite, Settings, Notifications, and Upgrade.

The main area is titled "My Workspace". It shows a sidebar with "My first collection" containing "First folder inside collection" (with two GET requests) and "Second folder inside collection" (with two GET requests). Below this, there's a section titled "Create a collection for your requests" with options to "Create Collection" or "Use Collection Template".

A central panel displays a POST request to "https://coach.myyhhixy.com/activity/getActivityByAccessToken?accessToken=2d7c25b2-27f5-4974-a...". The "Headers" tab is selected, showing a table with one row: "X-Token" with value "eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJ1c2VyS...". Other tabs include Params, Authorization, Body, Pre-request Script, Tests, and Settings. The "Cookies" tab is also visible.

Below the request details, a "Body" tab shows the response: "Pretty" view of JSON data with "message": "Need Auth" and "statusCode": 2003. Other tabs include Raw, Preview, and Visualize. The response status is 500 Internal Server Error, with 660 ms and 963 B details. Buttons for Save as Example and more options are available.

At the bottom, there are links for Online, Find and Replace, Console, Cookies, Capture requests, Runner, Trash, and Help.

Project Github Repositories

Github link:

<https://github.com/COMP90082-2023-SM1/GA-BoxJelly>

Github Standard:

Branch management standards

In order to standardize development, maintain commit history, and facilitate later maintenance, Git branch management is an essential work.

Roles:

Administrator

Developer

Current Branches:

Sprint2	
main branch	<ul style="list-style-type: none">• main
Functional development branch	<ul style="list-style-type: none">• develop-code-cleaning
	<ul style="list-style-type: none">• develop-encryption
	<ul style="list-style-type: none">• develop-front-end
	<ul style="list-style-type: none">• develop-server

Branch specification

Sprint 2

- [Sprint 2 Branch specification](#)

Sprint 3

Sprint 4:

Branch management:

The main branch serves as the primary branch for performing critical operations on the project, such as tagging or releasing versions. It represents the final, stable, and thoroughly tested version of the codebase that can be deployed to the production environment without errors or bugs.

The develop branch serves as the primary development branch that spans the entire lifecycle of the project. It is from this branch that feature branches are created for each iteration, allowing team members to work on specific features or tasks. Each team member maintains their own feature branch, which is checked out from the develop branch. Upon completion of the feature, the branch is merged back into the develop branch. This branch is generally used for developing new features or performing project maintenance tasks.

Under the develop branch, iteration-specific branches are created for developing specific functionality modules. Depending on the project management methodology, team members work on these branches by modifying the corresponding feature branch based on the tasks they have claimed. After completing the development work, the branch is merged into the develop branch, and the code is reviewed by other team members before merging.

The Test branch serves as the branch for testing the codebase. This branch is based on the develop branch and is used for comprehensive testing. If any bugs are found, they can be fixed directly on this branch and committed to it.

Sprint 2 Branch specification

Current Branches:

Sprint2	
main branch	<ul style="list-style-type: none">• main
Functional development branch	<ul style="list-style-type: none">• develop-code-cleaning
	<ul style="list-style-type: none">• develop-encryption
	<ul style="list-style-type: none">• develop-front-end
	<ul style="list-style-type: none">• develop-server

Branch Action

Branch name	status	Action
• develop-code-cleaning	Verified-test pass	Merge into main
• develop-encryption	Verified-test pass	Merge into main
• develop-front-end	Verified-test pass	Merge into main
• develop-server	Verified-test pass	Merge into main

Peer-to-peer code review

Review code :

main ▾ GA-BoxJelly / src / cm-backend / src / test / java / coachingmateanalytics / coachingmate /

Go to file Add file ...

edwinyao121	finish ActivityServiceImplTest and UserServiceImplTest and generateRe...	d0bd0ee 2 days ago	History
..			
controller	finish RegistryControllerTest and LoginControllerTest	4 days ago	
dao	finish ActivityServiceImplTest and UserServiceImplTest and generateRe...	2 days ago	
service	finish ActivityServiceImplTest and UserServiceImplTest and generateRe...	2 days ago	
AppTest.java	existing code	last month	
CoachingmateApplicationTests.java	existing code	last month	

Give feedback

Code Example:

```

1 package coachingmateanalytics.coachingmate.controller;
2
3 import coachingmateanalytics.coachingmate.entity.Activity;
4 import coachingmateanalytics.coachingmate.entity.ActivityDetails;
5 import coachingmateanalytics.coachingmate.entity.DashboardStastic;
6 import coachingmateanalytics.coachingmate.service.ActivityService;
7 import com.alibaba.fastjson.JSON;
8 import com.alibaba.fastjson.JSONArray;
9 import com.alibaba.fastjson.JSONObject;
10 import org.json.JSONObject;
11 import org.junit.jupiter.api.Test;
12 import org.junit.jupiter.api.extension.ExtendWith;
13 import org.junit.runner.RunWith;
14 import org.mockito.junit.jupiter.MockitoExtension;
15 //import org.junit.platform.runner.JUnitPlatform;
16 import org.springframework.beans.factory.annotation.Autowired;
17 import org.springframework.boot.test.context.SpringBootTest;
18 import coachingmateanalytics.coachingmate.controller.ActivityDataRetrieveController;
19 import org.springframework.http.ResponseEntity;
20
21 import javax.annotation.Resource;
22 import java.util.ArrayList;
23 import java.util.Collections;
24 import java.util.List;
25
26 import static org.junit.jupiter.api.Assertions.*;
27
28 @SpringBootTest
29 class ActivityDataRetrieveControllerTest{
30     @Autowired
31     private ActivityDataRetrieveController activityDataRetrieveController;
32     String accessToken="2d7c25b2-27f5-4974-a82f-2a9e41502b9f";
33     @Resource
34     ActivityService activityService;
35
36     @Test
37     void testGetActivityByAccessToken() {
38         ResponseEntity<List<Activity>> activitiesResponse = activityDataRetrieveController.getActivityByAccessToken(accessToken);
39         String jsonStringActivity="[{< n" +
40             " \\"durationInSeconds\\": 10,< n" +
41             " \\"averageSpeedInMetersPerSecond\\": 0.233,< n" +
42             " \\"averageHeartRateInBeatsPerMinute\\": 92,< n" +
43             " \\"distanceInMeters\\": 2.42,< n" +
44             " \\"activityName\\": \"Running\",< n" +
45             " \\"userId\\": \"898329394\",< n" +
46             " \\"deviceName\\": \"forerunner935\",< n" +
47             " \\"averagePaceInMinutesPerKilometer\\": 71.53076,< n" +
48             " \\"activityId\\": 10,< n" +
49             " \\"startTimeInSeconds\\": 1650191523,< n" +
50             " \\"userAccessToken\\": \"2d7c25b2-27f5-4974-a82f-2a9e41502b9f\",< n" +
51             " \\"startTimeOffsetInSeconds\\": 36000,< n" +
52             " \\"maxPaceInMinutesPerKilometer\\": 6.8956003,< n" +
53             " \\"maxHeartRateInBeatsPerMinute\\": 95,< n" +
54             " \\"summaryId\\": \"8654408419\",< n" +

```

Sprint 2 Peer-to-peer code review record:

ORGANISING YOUR CODE REVIEW (A FEW DAYS/HOURS BEFORE MEETING)	
Name and local of artifact (on Github) to be reviewed:	/GA-BoxJelly/src/cm-backend/src/test/java/coachingmateanalytics/coachingmate/controller/
What to be reviewed?	All file under controller & dao filefolder
When's the code review meeting happening:	25/4/2023
Reviewers:	Yuhang Yao, LingKang Zhou,Yukun Li,Rui Liu,Xiuyuan Zhu
CODE REVIEW MEETING (STARTING THE MEETING)	
Team:	GA-BoxJelly
Date:	25/4/2023
Time:	19:00:00
Facilitator:	Yuhang Yao
Reviewers:	LingKang Zhou,Yukun Li,Rui Liu,Xiuyuan Zhu

CODE REVIEW (DURING MEETING TIME)								
	Item Artifact (on GitHub)	Location (where the issue was found in the reviewed artifact?)	Severity	Type	Defects Category	Description	Fixed by the author?	Verified by the Moderator?
1	GA-BoxJelly/src/cm-backend/src/main/java/coachingmateanalytics/coachingmate/controller/OAuthController	function OAuthController ()	Trivial	Improvement	Documentation Defects (Comment)	Didn't delete the useless log information, so we would see messy information in terminal	Yes	Yes
2	/GA-BoxJelly/src/cm-backend/src/test/java/coachingmateanalytics/coachingmate/controller/LoginControllerTest.java	function testGetInfo	High /critical	Improvement	Logic Defects (Compute)	Token cannot be fixed and repeated tests cannot be performed	No	Yes
3	/GA-BoxJelly/src/cm-backend/src/test/java/coachingmateanalytics/coachingmate/controller/DashboardStatisticsDaoTest.java	function testInsertDashboardStatistics	High /critical	Improvement	Logic Defects (Performance)	When we test the data, we didn't remove the test case in database.	Yes	Yes
4	GA-BoxJelly/src/cm-backend/src/test/java/coachingmateanalytics/coachingmate/controller/LoginControllerTest.java	function testLogin()	Medium	Improvement	Documentation Defects (Naming)	Lack of documentation for JAVA programming specification	Yes	Yes
5	GA-BoxJelly/src/cm-backend/src/test/java/coachingmateanalytics/coachingmate/dao/ActivityDaoImplTest.java	function testSelectActivityDetailsByActivityId()	High /critical	Improvement	Structure Defects (Duplication)	While data selection, use wrong format String.	Yes	Yes
6	GA-BoxJelly/src/cm-backend/src/test/java/coachingmateanalytics/coachingmate/dao/UserDaoTest.java	function deleteUserByUsername()	Medium	Issue	Check Defects (Check User Input)	AssertEquals the same object, which would always pass the test but doesn't make sense.	Yes	Yes
7	GA-BoxJelly/src/cm-backend/src/main/java/coachingmateanalytics/coachingmate/controller/GarminPushController	function acceptPushedActivityDetail()	Trivial	Improvement	Documentation Defects (Comment)	There are too many useless comment in the function, which make the code messy.	Yes	Yes
END OF CODE REVIEW MEETING								
Number of severe/critical errors:	3							
Number of medium errors:	2							
Number of trivial errors:	0							
Total inspection time (hs):	2h							

ChatGPT code review

Time:

Sprint 3 week2

Review code :

<src/cm-backend/src/main/java/coachingmateanalytics/coachingmate/controller/LoginController.java>

Code Example:

```
13 import org.slf4j.LoggerFactory;
14 import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
15 + import org.springframework.data.mongodb.core.query.Criteria;
16 + import org.springframework.data.mongodb.core.query.Query;
17 + import org.springframework.http.ResponseEntity;
18 import org.springframework.web.bind.annotation.RequestBody;
19 import org.springframework.web.bind.annotation.RequestMapping;
20 import org.springframework.web.bind.annotation.RequestMethod;
21 import org.springframework.web.bind.annotation.RestController;
22
23 import javax.annotation.Resource;
24 + import java.io.UnsupportedEncodingException;
25 import java.util.Map;
26 // work done:
27 // 1. clean up the code
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88         userPartner.setPassword(""); // set password to empty
89         return userPartner;
90     }
91 + @RequestMapping(value = "/requestReset", method = RequestMethod.POST)
92 + @ApiOperation(value = "request reset password", notes = "request reset password")
93 + public ResponseEntity requestReset(@ApiParam(required = true, type = "String") String email) throws UnsupportedEncodingException {
94 + //     UserPartner userObject = JSONObject.parseObject(email,UserPartner.class);
95 + //     email = userObject.getEmail();
96 +     Query query = Query.query(Criteria.where("email").is(email));
97 +     if (mongoTemplate.exists(query,"user")){
98 +         sendEmail.sendResetEmail(email);
99 +         return ResponseEntity.ok("Request success");
100 +     }
101 +     else {
102 +         System.out.println(email);
103 +         return ResponseEntity.badRequest().body("Email does not exist");
104 +     }
105 + }
106 }
```

Sprint 3 ChatGPT code review record:

```
src/cm-backend/src/main/java/coachingmateanalytics/coachingmate/controller/LoginController.java
```

```
... ... @@ -12,12 +12,16 @@
12 12 import org.slf4j.Logger;
```



github-actions bot 12 minutes ago

ChatGPT review for src/cm-backend/src/main/java/coachingmateanalytics/coachingmate/controller/LoginController.java:
Documentation Defects:

1. Naming: The names of the software elements are generally descriptive and follow standard naming conventions.
2. Comment: The code has some comments, but they are not comprehensive enough to provide a clear understanding of the code's purpose and functionality.

Visual Representation Defects:

1. Bracket Usage: The code has no issues with incorrect or missing brackets.
2. Indentation: The code is properly indented and easy to read.
3. Long Line: There are no long code statements that hinder readability.

Structure Defects:

1. Dead Code: There is no dead code in the provided code.
2. Duplication: There is no duplicate code in the provided code.

New Functionality:

1. Use Standard Method: No single-purpose code statements are present that require a standardized approach.

Resource Defects:

1. Variable Initialization: All variables are properly initialized.
2. Memory Management: The code does not have any memory management issues.

Check Defects:

1. Check User Input: The code does not have any user input validation or handling.

Interface Defects:

1. Parameter: The code does not have any issues with incorrect or missing parameters when calling functions or libraries.

Logic Defects:

1. Compute: The code does not have any incorrect logic during system execution.
2. Performance: The efficiency of the algorithm used cannot be evaluated without additional context.

Recommended Changes:

1. Add more comprehensive comments to aid in understanding the code's purpose and functionality.
2. Consider adding user input validation and handling to ensure the security and reliability of the application.



```
src/cm-backend/src/main/java/coachingmateanalytics/coachingmate/controller/LoginController.java
```

```
@@ -12,12 +12,16 @@
```

```
import org.slf4j.Logger;
```

```
@github-actions github-actions bot 12 minutes ago
```

ChatGPT review for src/cm-backend/src/main/java/coachingmateanalytics/coachingmate/controller/LoginController.java:
Documentation Defects:

Naming: The names of the software elements are generally descriptive and follow standard naming conventions.

Comment: The code has some comments, but they are not comprehensive enough to provide a clear understanding of the code's purpose and functionality.

Visual Representation Defects:

Bracket Usage: The code has no issues with incorrect or missing brackets.

Indentation: The code is properly indented and easy to read.

Long Line: There are no long code statements that hinder readability.

Structure Defects:

Dead Code: There is no dead code in the provided code.

Duplication: There is no duplicate code in the provided code.

New Functionality:

Use Standard Method: No single-purpose code statements are present that require a standardized approach.

Resource Defects:

Variable Initialization: All variables are properly initialized.

Memory Management: The code does not have any memory management issues.

Check Defects:

Check User Input: The code does not have any user input validation or handling.

Interface Defects:

Parameter: The code does not have any issues with incorrect or missing parameters when calling functions or libraries.

Logic Defects:

Compute: The code does not have any incorrect logic during system execution.

Performance: The efficiency of the algorithm used cannot be evaluated without additional context.

Recommended Changes:

Add more comprehensive comments to aid in understanding the code's purpose and functionality.

Consider adding user input validation and handling to ensure the security and reliability of the application.

Testing

Test

Test ActivityDaoImpl

Aim of the test:

Developers can interact with the activity database.

Progress:

[Progress-Sprint 2 Fix API Test](#)

Requirement:

[GA-BoxJelly requirements 5 \(test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.6 testFindAllByAccessToken	We can find all activity by userAccessToken	1. created data by our own. 2. get the data from api. 3. compare the content.	Data0	the result of API is same as the result get by our creating	Pass
T02	R5.6 testSelectActivityDetailsByAccessToken	We can find all activity details by userAccessToken	1. created data by our own. 2. get the data from api. 3. compare the content.	Data1	the result of API is same as the result get by our creating	Pass
T03	R5.6 testInsertActivity	we can insert the activity to database.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data2	the result of API is same as the result get by our creating	Pass
T04	R5.6 testInsertActivityDetails	We can insert activity details to database.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data3	the result of API is same as the result get by our creating	Pass
T05	R5.6 testSelectActivityDetailsByActivityId	We can get the activity details by activity ID	1. created data by our own. 2. get the data from api. 3. compare the content.	Data1	the result of API is same as the result get by our creating	Pass
T06	R5.6 testSelectActivityDetailsByAccessTokenAndType	We can get the specific activities by userAccessToken and type.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data1	the result of API is same as the result get by our creating	Pass

Test developers:

Data:

data 0:

```
"[{\n" +
" \\"durationInSeconds\\": 10,\n" +
" \\"averageSpeedInMetersPerSecond\\": 0.233,\n" +
" \\"averageHeartRateInBeatsPerMinute\\": 92,\n" +
" \\"distanceInMeters\\": 2.42,\n" +
" \\"activityName\\": \"Running\",\\n" +
" \\"userId\\": \"898329394\",\\n" +
" \\"deviceName\\": \"forerunner935\",\\n" +
" \\"averagePaceInMinutesPerKilometer\\": 71.53076,\n" +
" \\"activityId\\": 10,\n" +
" \\"startTimeInSeconds\\": 1650191523,\n" +
" \\"userAccessToken\\": \"2d7c25b2-27f5-4974-a82f-2a9e41502b9f\",\\n" +
" \\"startOffsetInSeconds\\": 36000,\n" +
" \\"maxPaceInMinutesPerKilometer\\": 6.8956003,\n" +
" \\"maxHeartRateInBeatsPerMinute\\": 95,\n" +
" \\"summaryId\\": \"8654408419\",\\n" +
" \\"maxSpeedInMetersPerSecond\\": 2.417,\n" +
" \\"activityType\\": \"RUNNING\"\n" +
"}]"
```

data 1:

```
" [ \n" +
" {\n" +
" \\"summary\\": {\n" +
" \\"durationInSeconds\\": 10,\n" +
" \\"averageSpeedInMetersPerSecond\\": 1.362,\n" +
" \\"averageHeartRateInBeatsPerMinute\\": 88,\n" +
" \\"distanceInMeters\\": 13.39,\n" +
" \\"activityName\\": \"Running\",\\n" +
" \\"deviceName\\": \"forerunner935\",\\n" +
" \\"steps\\": 4,\n" +
" \\"averageRunCadenceInStepsPerMinute\\": 25.09375,\n" +
" \\"averagePaceInMinutesPerKilometer\\": 12.236906,\n" +
" \\"activityId\\": 10,\n" +
" \\"startTimeInSeconds\\": 1650195177,\n" +
" \\"startOffsetInSeconds\\": 36000,\n" +
" \\"maxPaceInMinutesPerKilometer\\": 4.8676014,\n" +
" \\"maxHeartRateInBeatsPerMinute\\": 93,\n" +
" \\"maxRunCadenceInStepsPerMinute\\": 138,\n" +
" \\"maxSpeedInMetersPerSecond\\": 3.424,\n" +
" \\"activityType\\": \"RUNNING\"\n" +
"},\n" +
" \\"activityId\\": 10,\n" +
" \\"userAccessToken\\": \"2d7c25b2-27f5-4974-a82f-2a9e41502b9f\",\\n" +
" \\"summaryId\\": \"10-detail\",\\n" +
" \\"laps\\": [\n" +
" {\n" +
" \\"startTimeInSeconds\\": 1650195177\n" +
" },\n" +
" ],\n" +
" \\"userId\\": \"898329394\",\\n" +
" \\"samples\\": [\n" +
" {\n" +
" \\"startTimeInSeconds\\": 1650195177,\n" +
" \\"timerDurationInSeconds\\": 0,\n" +
" \\"movingDurationInSeconds\\": 0,\n" +
" \\"heartRate\\": 84,\n" +
" \\"elevationInMeters\\": 33.20000076293945,\n" +
" \\"speedMetersPerSecond\\": 0,\n" +
" \\"clockDurationInSeconds\\": 0,\n" +
" \\"stepsPerMinute\\": 0,\n" +
" \\"airTemperatureCelsius\\": 32,\n" +
" \\"totalDistanceInMeters\\": 0,\n" +
" \\"latitudeInDegree\\": -37.813,\n" +
" \\"longitudeInDegree\\": 144.96\n" +
"},\n" +
" {\n" +
" \\"startTimeInSeconds\\": 1650195178,\n" +
" \\"timerDurationInSeconds\\": 1,\n" +
" \\"movingDurationInSeconds\\": 0,\n" +
" \\"heartRate\\": 86,\n" +
" \\"latitud
```

```

    "elevationInMeters": 33.20000076293945, \n" +
    "speedMetersPerSecond": 0, \n" +
    "clockDurationInSeconds": 1, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 0, \n" +
    "latitudeInDegree": -37.815, \n" +
    "longitudeInDegree": 144.95 \n" +
}, \n" +
{\n" +
    "startTimeInSeconds": 1650195182, \n" +
    "timerDurationInSeconds": 5, \n" +
    "movingDurationInSeconds": 0, \n" +
    "heartRate": 129, \n" +
    "elevationInMeters": 31.799999237060547, \n" +
    "speedMetersPerSecond": 0, \n" +
    "clockDurationInSeconds": 5, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 0, \n" +
    "latitudeInDegree": -37.816, \n" +
    "longitudeInDegree": 144.94 \n" +
}, \n" +
{\n" +
    "startTimeInSeconds": 1650195185, \n" +
    "timerDurationInSeconds": 8, \n" +
    "movingDurationInSeconds": 3, \n" +
    "heartRate": 129, \n" +
    "elevationInMeters": 31, \n" +
    "speedMetersPerSecond": 3.4240000247955322, \n" +
    "clockDurationInSeconds": 8, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 6.539999961853027, \n" +
    "latitudeInDegree": -37.817, \n" +
    "longitudeInDegree": 144.93 \n" +
}, \n" +
{\n" +
    "startTimeInSeconds": 1650195187, \n" +
    "timerDurationInSeconds": 10, \n" +
    "movingDurationInSeconds": 5, \n" +
    "heartRate": 112, \n" +
    "elevationInMeters": 31, \n" +
    "speedMetersPerSecond": 3.4240000247955322, \n" +
    "clockDurationInSeconds": 10, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 13.390000343322754, \n" +
    "latitudeInDegree": -37.818, \n" +
    "longitudeInDegree": 144.92 \n" +
} \n" +
] \n" +
} \n" +
"

```

data 2:

```

" { \n" +
    "durationInSeconds": 10, \n" +
    "averageSpeedInMetersPerSecond": 0.233, \n" +
    "averageHeartRateInBeatsPerMinute": 92, \n" +
    "distanceInMeters": 2.42, \n" +
    "activityName": "Walking", \n" +
    "userId": "898329394", \n" +
    "deviceName": "forerunner935", \n" +
    "averagePaceInMinutesPerKilometer": 71.53076, \n" +
    "activityId": 11, \n" +
    "startTimeInSeconds": 1650191523, \n" +
    "userAccessToken": "7e88865e-eb7d-47c8-8418-80cd24bbd4e6", \n" +
    "startTimeOffsetInSeconds": 36000, \n" +
    "maxPaceInMinutesPerKilometer": 6.8956003, \n" +
    "maxHeartRateInBeatsPerMinute": 95, \n" +
    "summaryId": "8654408419", \n" +
    "maxSpeedInMetersPerSecond": 2.417, \n" +
    "activityType": "WALKING" \n" +
}

```

data 3:

```

" { \n" +
    "summary": { \n" +
        "durationInSeconds": 10, \n" +
        "averageSpeedInMetersPerSecond": 1.362, \n" +
        "averageHeartRateInBeatsPerMinute": 88, \n" +

```

```

    "distanceInMeters": 13.39, \n" +
    "activityName": "Walking", \n" +
    "deviceName": "forerunner935", \n" +
    "steps": 4, \n" +
    "averageRunCadenceInStepsPerMinute": 25.09375, \n" +
    "averagePaceInMinutesPerKilometer": 12.236906, \n" +
    "activityId": 11, \n" +
    "startTimeInSeconds": 1650195177, \n" +
    "startTimeOffsetInSeconds": 36000, \n" +
    "maxPaceInMinutesPerKilometer": 4.8676014, \n" +
    "maxHeartRateInBeatsPerMinute": 93, \n" +
    "maxRunCadenceInStepsPerMinute": 138, \n" +
    "maxSpeedInMetersPerSecond": 3.424, \n" +
    "activityType": "WALKING", \n" +
}, \n" +
"activityId": 11, \n" +
"userAccessToken": "7e88865e-eb7d-47c8-8418-80cd24bbd4e6", \n" +
"summaryId": "11-detail", \n" +
"laps": [ \n" +
{ \n" +
    "startTimeInSeconds": 1650195177 \n" +
}, \n" +
], \n" +
"userId": "898329394", \n" +
"samples": [ \n" +
{ \n" +
    "startTimeInSeconds": 1650195177, \n" +
    "timerDurationInSeconds": 0, \n" +
    "movingDurationInSeconds": 0, \n" +
    "heartRate": 84, \n" +
    "elevationInMeters": 33.20000076293945, \n" +
    "speedMetersPerSecond": 0, \n" +
    "clockDurationInSeconds": 0, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 0, \n" +
    "latitudeInDegree": -37.813, \n" +
    "longitudeInDegree": 144.96 \n" +
}, \n" +
{ \n" +
    "startTimeInSeconds": 1650195178, \n" +
    "timerDurationInSeconds": 1, \n" +
    "movingDurationInSeconds": 0, \n" +
    "heartRate": 86, \n" +
    "elevationInMeters": 33.20000076293945, \n" +
    "speedMetersPerSecond": 0, \n" +
    "clockDurationInSeconds": 1, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 0, \n" +
    "latitudeInDegree": -37.815, \n" +
    "longitudeInDegree": 144.95 \n" +
}, \n" +
{ \n" +
    "startTimeInSeconds": 1650195182, \n" +
    "timerDurationInSeconds": 5, \n" +
    "movingDurationInSeconds": 0, \n" +
    "heartRate": 129, \n" +
    "elevationInMeters": 31.799999237060547, \n" +
    "speedMetersPerSecond": 0, \n" +
    "clockDurationInSeconds": 5, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 0, \n" +
    "latitudeInDegree": -37.816, \n" +
    "longitudeInDegree": 144.94 \n" +
}, \n" +
{ \n" +
    "startTimeInSeconds": 1650195185, \n" +
    "timerDurationInSeconds": 8, \n" +
    "movingDurationInSeconds": 3, \n" +
    "heartRate": 129, \n" +
    "elevationInMeters": 31, \n" +
    "speedMetersPerSecond": 3.4240000247955322, \n" +
    "clockDurationInSeconds": 8, \n" +
    "stepsPerMinute": 0, \n" +
    "airTemperatureCelcius": 32, \n" +
    "totalDistanceInMeters": 6.539999961853027, \n" +
    "latitudeInDegree": -37.817, \n" +
    "longitudeInDegree": 144.93 \n" +
}, \n" +
{ \n" +
    "startTimeInSeconds": 1650195187, \n" +
    "timerDurationInSeconds": 10, \n" +

```

```
"      \"movingDurationInSeconds\": 5,\n" +
"      \"heartRate\": 112,\n" +
"      \"elevationInMeters\": 31,\n" +
"      \"speedMetersPerSecond\": 3.4240000247955322,\n" +
"      \"clockDurationInSeconds\": 10,\n" +
"      \"stepsPerMinute\": 0,\n" +
"      \"airTemperatureCelcius\": 32,\n" +
"      \"totalDistanceInMeters\": 13.390000343322754,\n" +
"      \"latitudeInDegree\": -37.818,\n" +
"      \"longitudeInDegree\": 144.92\n" +
"}\n" +
"]\n" +
"}"
```

Test ActivityDataRetrieveController

Aim of the test:

the activity data can be retrieved.

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.1 testGetActivityByAccessToken	we can use the user accesstoken to get the activity list	1. created data by our own. 2. get the data from api. 3. compare the content.	Data 0	the result of API is same as the result get by our creating	Pass
T02	R5.1 testGetActivityDetailsByAccess Token	we can use the user accesstoken to get the activity details list	1. created data by our own. 2. get the data from api. 3. compare the content.	Data1	the result of API is same as the result get by our creating	Pass
T03	R5.1 testGetActivityDetailsById	we can use the activity id to get the activity details list	1. created data by our own. 2. get the data from api. 3. compare the content.	Data1	the result of API is same as the result get by our creating	Pass
T04	R5.1 testGetActivityByAccessTokenAndType	We can use userAccessToken and its type to find activities.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data0	the result of API is same as the result get by our creating	Pass
T05	R5.1 testGetDashboardStatisticsByAccessToken	We can get the Dashboard statistics data by userAccess Token.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data2	the result of API is same as the result get by our creating	Pass

Test developers:

Yuhang Yao, Yukun Li

Data:

```

Data 0:[{
  "durationInSeconds": 10,
  "averageSpeedInMetersPerSecond": 0.233,
  "averageHeartRateInBeatsPerMinute": 92,
  "distanceInMeters": 2.42,
  "activityName": "Running",
  "userId": "898329394",
  "deviceName": "forerunner935",
  "averagePaceInMinutesPerKilometer": 71.53076,
  "activityId": 10,
  "startTimeInSeconds": 1650191523,
  "userAccessToken": "bd5b13a9-9eed-4204-a807-94bf0eb06356",
  "startTimeOffsetInSeconds": 36000,
  "maxPaceInMinutesPerKilometer": 6.8956003,
  "maxHeartRateInBeatsPerMinute": 95,
  "summaryId": "8654408419",
  "maxSpeedInMetersPerSecond": 2.417,
  "activityType": "RUNNING"
}]

Data 1:{
```

```

  "summary": {
    "durationInSeconds": 10,
    "averageSpeedInMetersPerSecond": 1.362,
    "averageHeartRateInBeatsPerMinute": 88,
    "distanceInMeters": 13.39,
    "activityName": "Running",
    "deviceName": "forerunner935",
    "steps": 4,
    "averageRunCadenceInStepsPerMinute": 25.09375,
    "averagePaceInMinutesPerKilometer": 12.236906,
    "activityId": 10,
    "startTimeInSeconds": 1650195177,
    "startTimeOffsetInSeconds": 36000,
    "maxPaceInMinutesPerKilometer": 4.8676014,
    "maxHeartRateInBeatsPerMinute": 93,
    "maxRunCadenceInStepsPerMinute": 138,
    "maxSpeedInMetersPerSecond": 3.424,
    "activityType": "RUNNING"
  },
  "activityId": 10,
  "userAccessToken": "bd5b13a9-9eed-4204-a807-94bf0eb06356",
  "summaryId": "10-detail",
  "laps": [
    {
      "startTimeInSeconds": 1650195177
    }
  ],
  "userId": "898329394",
  "samples": [
    {
      "startTimeInSeconds": 1650195177,
      "timerDurationInSeconds": 0,
      "movingDurationInSeconds": 0,
      "heartRate": 84,
      "elevationInMeters": 33.20000076293945,
      "speedMetersPerSecond": 0,
      "clockDurationInSeconds": 0,
      "stepsPerMinute": 0,
      "airTemperatureCelsius": 32,
      "totalDistanceInMeters": 0,
      "latitudeInDegree": -37.813,
      "longitudeInDegree": 144.96
    },
    {
      "startTimeInSeconds": 1650195178,
      "timerDurationInSeconds": 1,
      "movingDurationInSeconds": 0,
      "heartRate": 86,
      "elevationInMeters": 33.20000076293945,
      "speedMetersPerSecond": 0,
      "clockDurationInSeconds": 1,
      "stepsPerMinute": 0,
      "airTemperatureCelsius": 32,
      "totalDistanceInMeters": 0,
      "latitudeInDegree": -37.815,
      "longitudeInDegree": 144.95
    },
    {
      "
```

```
"startTimelnSeconds": 1650195182,
"timerDurationlnSeconds": 5,
"movingDurationlnSeconds": 0,
"heartRate": 129,
"elevationInMeters": 31.799999237060547,
"speedMetersPerSecond": 0,
"clockDurationlnSeconds": 5,
"stepsPerMinute": 0,
"airTemperatureCelcius": 32,
"totalDistanceInMeters": 0,
"latitudeInDegree": -37.816,
"longitudeInDegree": 144.94
},
{
"startTimelnSeconds": 1650195185,
"timerDurationlnSeconds": 8,
"movingDurationlnSeconds": 3,
"heartRate": 129,
"elevationInMeters": 31,
"speedMetersPerSecond": 3.4240000247955322,
"clockDurationlnSeconds": 8,
"stepsPerMinute": 0,
"airTemperatureCelcius": 32,
"totalDistanceInMeters": 6.539999961853027,
"latitudeInDegree": -37.817,
"longitudeInDegree": 144.93
},
{
"startTimelnSeconds": 1650195187,
"timerDurationlnSeconds": 10,
"movingDurationlnSeconds": 5,
"heartRate": 112,
"elevationInMeters": 31,
"speedMetersPerSecond": 3.4240000247955322,
"clockDurationlnSeconds": 10,
"stepsPerMinute": 0,
"airTemperatureCelcius": 32,
"totalDistanceInMeters": 13.390000343322754,
"latitudeInDegree": -37.818,
"longitudeInDegree": 144.92
}
]
```

Data2:

```

"          0,\n" +
"          0,\n" +
"          0,\n" +
"          0\n" +
"],\n" +
\"except\" : [\n" +
    90,\n" +
    90\n" +
],\n" +
},\n" +
\"SwimmingTime\" : {\n" +
    \"actual\" : [\n" +
        0,\n" +
        0\n" +
    ],\n" +
    \"except\" : [\n" +
        90,\n" +
        90\n" +
    ],\n" +
},\n" +
\"RunningTime\" : {\n" +
    \"actual\" : [\n" +
        0,\n" +
        0\n" +
    ],\n" +
    \"except\" : [\n" +
        90,\n" +
        90\n" +
    ],\n" +
},\n" +
\"radarActivities\" : {\n" +
    \"RUNNING\" : {\n" +
        \"time\" : 0.0,\n" +
        \"calories\" : 0,\n" +
        \"distance\" : 7.26,\n" +
        \"avgSpeed\" : 0.0,\n" +
        \"peakSpeed\" : 2.417,\n" +
        \"avgHeartRate\" : 0\n" +
    },\n" +
    \"OPEN_WATER_SWIMMING\" : {\n" +
        \"time\" : 0.0,\n" +
        \"calories\" : 0,\n" +
        \"distance\" : 0.0,\n" +
        \"avgSpeed\" : 0.0,\n" +
        \"peakSpeed\" : 0.0,\n" +
        \"avgHeartRate\" : 0\n" +
    },\n" +
    \"ROAD_BIKING\" : {\n" +
        \"time\" : 0.0,\n" +
        \"calories\" : 0,\n" +
        \"distance\" : 0.0,\n" +
        \"avgSpeed\" : 0.0,\n" +
        \"peakSpeed\" : 0.0,\n" +
        \"avgHeartRate\" : 0\n" +
    },\n" +
},\n" +
\"hearRateZones\" : [\n" +
    [\n" +

```


Test DashboardStatisticsDao

Aim of the test:

We can interact with the Dashboard database

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.7 testInsertDashboardStatistics	we can insert Dashboard Statistics to database.	1. check if the specific data is null. 2. created data by our own and insert it. 3. get the data from api. 4. compare the content.	Data 0	the result of database is same as the result get by our creating	Pass
T02	R5.7 testSelectDashboardStatisticsByAccessToken	we can use the user accesstoken to get the dashboard Static data.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data0	the result of API is same as the result get by our creating	Pass

Test developers:

Yuhang Yao, Yukun Li

Data:

Data 0:

```
"{\n" +\n    "userAccessToken\" : \"2d7c25b2-27f5-4974-a82f-2a9e41502b9f\", \n" +\n    "ttlActivityTimes\" : 3, \n" +\n    "ttlRunningTimes\" : 3, \n" +\n    "ttlRiddingTimes\" : 0, \n" +\n    "ttlSwimmingTimes\" : 0, \n" +\n    "ttlActivityTime\" : 0.0, \n" +\n    "ttlRunningTime\" : 0.0, \n" +\n    "ttlRiddingTime\" : 0.0, \n" +\n    "ttlSwimmingTime\" : 0.0, \n" +\n    "activityTimeChartMap\" : {\n" +\n        "RiddingTime\" : {\n" +\n            "actual\" : [\n                0, \n                0, \n                0, \n                0, \n                0, \n                0, \n                0\n            ], \n            \"except\" : [\n                90, \n                90, \n                90, \n                90, \n                90\n            ]\n        }\n    }\n}
```

```

"         90,\n" +
"         90,\n" +
"         90\n" +
"]\n" +
},\n" +
\"AllTime\" : {\n" +
  \"actual\" : [\n" +
    0,\n" +
    0\n" +
],\n" +
  \"except\" : [\n" +
    90,\n" +
    90\n" +
],\n" +
},\n" +
\"SwimmingTime\" : {\n" +
  \"actual\" : [\n" +
    0,\n" +
    0\n" +
],\n" +
  \"except\" : [\n" +
    90,\n" +
    90\n" +
],\n" +
},\n" +
\"RunningTime\" : {\n" +
  \"actual\" : [\n" +
    0,\n" +
    0\n" +
],\n" +
  \"except\" : [\n" +
    90,\n" +
    90\n" +
],\n" +
},\n" +
\"radarActivities\" : {\n" +
  \"RUNNING\" : {\n" +
    \"time\" : 0.0,\n" +
    \"calories\" : 0,\n" +
    \"distance\" : 7.26,\n" +
    \"avgSpeed\" : 0.0,\n" +
    \"peakSpeed\" : 2.417,\n" +
    \"avgHeartRate\" : 0\n" +
  },\n" +
  \"OPEN_WATER_SWIMMING\" : {\n" +
    \"time\" : 0.0,\n" +
    \"calories\" : 0,\n" +
    \"distance\" : 0.0,\n" +
    \"avgSpeed\" : 0.0,\n" +
    \"peakSpeed\" : 0.0,\n" +
    \"avgHeartRate\" : 0\n" +
  },\n" +
  \"ROAD_BIKING\" : {\n" +

```


Test generateRequestTokenSecret

Aim of the test:

We can generate token and its secret and then store to database

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.8 testGenerateRequestTokenSecret	we can generate the token and secret and then store to database.	1. created data by our own. 2. get the data from api. 3. compare the content.	it is randomly generated	generate successfully	Pass

Test developers:

Yuhang Yao, Yukun Li

Test ActivityServiceImpl

Aim of the test:

Activities, activity details can be inserted and be searched with enough information.

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.11 selectActivityByAccessToken()	Test functionality and make sure activities can be obtained using accessToken.	1. Copy activities from database 2. Run the function using the accessToken. 3. compare the result of activity lists.	Data 0	The activities obtained using the function is the same as those in the database.	Pass
T02	R5.11 selectActivityDetailsByAccessToken()	Test functionality and make sure activity details can be obtained using accessToken.	1. Copy activity details from database 2. Run the function using the accessToken. 3. compare the result of activity detail lists.	Data 1	The activity details obtained using the function is the same as those in the database.	Pass
T03	R5.11 selectActivityDetailsById()	Test functionality and make sure activities can be obtained using activityID.	1. Copy activities from database 2. Run the function using the activityID. 3. compare the result of activity lists.	Data 0	The activities obtained using the function is the same as those in the database.	Pass
T04	R5.11 insertActivity()	Test functionality and make sure new activities can be inserted.	1. Make sure the activity to be inserted is not in the database. 2. Run the function to insert. 3. Check if the activity can be found in the database.	new Activity()	The inserted activity is in the database.	Pass
T05	R5.11 insertActivityDetails()	Test functionality and make sure new activity details can be inserted.	1. Make sure the activity detail to be inserted is not in the database. 2. Run the function to insert. 3. Check if the activity details can be found in the database.	new ActivityDetails()	The inserted activity details are in the database.	Pass

Test developers:

Yuhang Yao, Yukun Li

Data:

Data 0:

```

" { \n" +
    "   \"durationInSeconds\": 10,\n" +
    "   \"averageSpeedInMetersPerSecond\": 0.233,\n" +
    "   \"averageHeartRateInBeatsPerMinute\": 92,\n" +
    "   \"distanceInMeters\": 2.42,\n" +
    "   \"activityName\": \"Running\", \n" +
    "   \"userId\": \"898329394\", \n" +
    "   \"deviceName\": \"forerunner935\", \n" +
    "   \"averagePaceInMinutesPerKilometer\": 71.53076,\n" +
    "   \"activityId\": 10,\n" +
    "   \"startTimeInSeconds\": 1650191523,\n" +
    "   \"userAccessToken\": \"2d7c25b2-27f5-4974-a82f-2a9e41502b9f\", \n" +
    "   \"startOffsetInSeconds\": 36000,\n" +
    "   \"maxPaceInMinutesPerKilometer\": 6.8956003,\n" +
    "   \"maxHeartRateInBeatsPerMinute\": 95,\n" +
    "   \"summaryId\": \"8654408419\", \n" +
    "   \"maxSpeedInMetersPerSecond\": 2.417,\n" +
    "   \"activityType\": \"RUNNING\"\n" +
  " } "

```

Data 1:

```

" { \n" +
    "   \"summary\": {\n" +
        "     \"durationInSeconds\": 10,\n" +
        "     \"averageSpeedInMetersPerSecond\": 1.362,\n" +
        "     \"averageHeartRateInBeatsPerMinute\": 88,\n" +
        "     \"distanceInMeters\": 13.39,\n" +
        "     \"activityName\": \"Running\", \n" +
        "     \"deviceName\": \"forerunner935\", \n" +
        "     \"steps\": 4,\n" +
        "     \"averageRunCadenceInStepsPerMinute\": 25.09375,\n" +
        "     \"averagePaceInMinutesPerKilometer\": 12.236906,\n" +
        "     \"activityId\": 10,\n" +
        "     \"startTimeInSeconds\": 1650195177,\n" +
        "     \"startOffsetInSeconds\": 36000,\n" +
        "     \"maxPaceInMinutesPerKilometer\": 4.8676014,\n" +
        "     \"maxHeartRateInBeatsPerMinute\": 93,\n" +
        "     \"maxRunCadenceInStepsPerMinute\": 138,\n" +
        "     \"maxSpeedInMetersPerSecond\": 3.424,\n" +
        "     \"activityType\": \"RUNNING\"\n" +
    },\n" +
    "   \"activityId\": 10,\n" +
    "   \"userAccessToken\": \"2d7c25b2-27f5-4974-a82f-2a9e41502b9f\", \n" +
    "   \"summaryId\": \"10-detail\", \n" +
    "   \"laps\": [\n" +
        "     {\n" +
            "       \"startTimeInSeconds\": 1650195177\n" +
        "     }\n" +
    ],\n" +
    "   \"userId\": \"898329394\", \n" +
    "   \"samples\": [\n" +
        "     {\n" +
            "       \"startTimeInSeconds\": 1650195177,\n" +
            "       \"timerDurationInSeconds\": 0,\n" +
            "       \"movingDurationInSeconds\": 0,\n" +
            "       \"heartRate\": 84,\n" +
            "       \"elevationInMeters\": 33.20000076293945,\n" +
            "       \"speedMetersPerSecond\": 0,\n" +
            "       \"clockDurationInSeconds\": 0,\n" +
            "       \"stepsPerMinute\": 0,\n" +
            "       \"airTemperatureCelcius\": 32,\n" +
            "       \"totalDistanceInMeters\": 0,\n" +
            "       \"latitudeInDegree\": -37.813,\n" +
            "       \"longitudeInDegree\": 144.96\n" +
        },\n" +
        "     {\n" +
            "       \"startTimeInSeconds\": 1650195178,\n" +
            "       \"timerDurationInSeconds\": 1,\n" +
            "       \"movingDurationInSeconds\": 0,\n" +
            "       \"heartRate\": 86,\n" +
            "       \"elevationInMeters\": 33.20000076293945,\n" +
            "       \"speedMetersPerSecond\": 0,\n" +
            "       \"clockDurationInSeconds\": 1,\n" +
            "       \"stepsPerMinute\": 0,\n" +
            "       \"airTemperatureCelcius\": 32,\n" +
            "       \"totalDistanceInMeters\": 0,\n" +
            "       \"latitudeInDegree\": -37.815,\n" +
            "       \"longitudeInDegree\": 144.95\n" +
        },\n" +
        "     {\n" +
            "       \"startTimeInSeconds\": 1650195182,\n" +
            "       \"timerDurationInSeconds\": 5,\n" +
            "       \"movingDurationInSeconds\": 0,\n" +

```

```

    "heartRate": 129, \n +
    "elevationInMeters": 31.799999237060547, \n +
    "speedMetersPerSecond": 0, \n +
    "clockDurationInSeconds": 5, \n +
    "stepsPerMinute": 0, \n +
    "airTemperatureCelsius": 32, \n +
    "totalDistanceInMeters": 0, \n +
    "latitudeInDegree": -37.816, \n +
    "longitudeInDegree": 144.94\n +
}, \n +
{\n +
    "startTimeInSeconds": 1650195185, \n +
    "timerDurationInSeconds": 8, \n +
    "movingDurationInSeconds": 3, \n +
    "heartRate": 129, \n +
    "elevationInMeters": 31, \n +
    "speedMetersPerSecond": 3.4240000247955322, \n +
    "clockDurationInSeconds": 8, \n +
    "stepsPerMinute": 0, \n +
    "airTemperatureCelsius": 32, \n +
    "totalDistanceInMeters": 6.539999961853027, \n +
    "latitudeInDegree": -37.817, \n +
    "longitudeInDegree": 144.93\n +
}, \n +
{\n +
    "startTimeInSeconds": 1650195187, \n +
    "timerDurationInSeconds": 10, \n +
    "movingDurationInSeconds": 5, \n +
    "heartRate": 112, \n +
    "elevationInMeters": 31, \n +
    "speedMetersPerSecond": 3.4240000247955322, \n +
    "clockDurationInSeconds": 10, \n +
    "stepsPerMinute": 0, \n +
    "airTemperatureCelsius": 32, \n +
    "totalDistanceInMeters": 13.390000343322754, \n +
    "latitudeInDegree": -37.818, \n +
    "longitudeInDegree": 144.92\n +
}\n +
]\n +
}\n ;

```

Test UserServiceImpl

Aim of the test:

Make sure user services function normally.

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.12 testLoginCheck()	Test functionality and make sure newly registered user can log in.	1. Register a new user. 2. Make sure the user can log in. 3. Remove the user	Data 0	The contents of testPartner and the userPartner are identical using register and logincheck.	Pass
T02	R5.12 test GetUser()	Test functionality and make sure newly registered user can be obtained.	1. Register a new user. 2. Make sure the user can be found.	Data 0	The contents of testPartner and the userPartner are identical using register and getUser.	Pass
T03	R5.12 test GetUserByToken()	Test functionality and make sure the newly registered user can be found using the access Token.	1. Register a new user. 2. Make sure the user can be found using the accessToken.	Data 0	The contents of testPartner and the userPartner are identical using register and getUserByToken.	Pass
T04	R5.12 testRegister()	Test functionality and make sure newly registered user can log in.	1. Register a new user. 2. Make sure the user can log in. 3. Remove the user	Data 0	The contents of testPartner and the userPartner are identical using register and logincheck.	Pass

Test developers:

Yuhang Yao, Yukun Li

Data:

Data 0:

```
testUser = new UserPartner();
testUser.setUsername("testUser");
testUser.setEmail("testUser@gmail.com");
testUser.setPassword("password");
testUser.setToken("token");
testUser.setTokenDate(new Date());
```

Test LoginController

Aim of the test:

Existing users can log in.

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.3 testLogin()	Test functionality and make sure existing users can log in.	1. Copy user data from database 2. Log in using the function given username and password. 3. compare the result.	Data 0	the result of API is same as the data in database	Pass
T02	R5.3 testGetUserInfo()	Test functionality and make sure user info can be accessed given the Token	1. Copy user data from database 2. Get user info using the given token. 3. compare the result.	Data0	the result of API is same as the user info in database	Pass

Test developers:

Yuhang Yao, Yukun Li

Data:

Data 0:

```
"{\n" +\n    "userId": "898329415",\n    "username": "0424",\n    "fullname": "0424",\n    "email": "cxvjier@gmail.com",\n    "password": "$2a$10$mxHz85Hl5eBUU.9T/6ZeTyMsi4a7CyvMSM9C2hXGe/A0BS3686e",\n    "token": "523ed521-7e7d-48ed-a08d-28fb12e757b8",\n    "tokenDate": "2023-04-24T04:00:37.361+0000",\n    "userAccessSecret": "gHRxkZnGWB6WQKXY115pMoG4e2uBkr65cYe",\n    "userAccessToken": "2d7c25b2-27f5-4974-a82f-2a9e41502b9f"\n}\"\n";
```

Test of Code Cleaning

Aim of the test:

Ensure that the implemented normal functions of the back-end code are not affected by the progress of code cleaning.

Progress

[Code cleaning](#)

Requirement:

[Requirements 1 \(Code cleaning\)](#)

Test specification

Test Case ID	Requirement	Test Case	Test Steps	Test Data	Expected Result	Pass/Fail
T01	R1	Log in Testing Account to view Activity List and then view the activity type with the icon, activity name and activity date for this activity.	1) Log in to the testing account. 2) Click the Activities button on the left side. 3) View the activity list.	User ID: 0402 Password AAAAaa1111	<ul style="list-style-type: none">The activity list can be shown on the right side.The event icon, event type, activity name and activity date were loaded successfully.	Pass
T02	R1	View the total distance for this type of activity.	1) View the total distance from the activity list. 2) Click one of the activities, and then view the total distance on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The total distance (km) could be viewed from the activity list.The total distance (km) could be viewed from the activity detailed page.	Pass
T03	R1	View the total time for this type of activity.	1) View the total time from the activity list. 2) Click one of the activities, and then view the total time on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The total time (xx:yy:zz) could be viewed from the activity list.The total time (xx:yy:zz) could be viewed from the activity detailed page.	Pass
T04	R1	View the average speed for this type of activity.	1) View the average speed from the activity list. 2) Click one of the activities, and then view the average speed on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The average speed(km/h) could be viewed from the activity list.The average speed(km/h) could be viewed from the activity detailed page.	Pass
T05	R1	View the ascent or descent distance for this type of activity.	1) View the ascent or descent distance from the activity list. 2) Click one of the activities, and then view the ascent or descent distance on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The ascent or descent distance (m) could be viewed from the activity list.The ascent or descent distance(m) could be viewed from the activity detailed page.	Pass
T06	R1	View the average heart rate.	1) View the average heart rate from the activity list. 2) Click one of the activities, and then view the average heart rate on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The average heart rate could be viewed from the activity list.The average heart rate could be viewed from the activity detailed page.	Pass
T07	R1	View the activity in descending order	1) Click the activity button on the left side. 2) The activity is in descending order (latest).	Activity Data	<ul style="list-style-type: none">The descending ordering was displayed success.	Pass

T08	R1	View the sports route on the heatmap for this activity.	1) Click one of the activities from the activity list. 2) The heat map with the sports route will be shown on the detailed page.	Activity Data	<ul style="list-style-type: none"> The sports route was loaded successfully The sports route has its starting point and ending point 	Pass
T09	R1	Different levels for heart rate and power with colour and number range.	1) Could be tested with T10 together or tested by console log.	Activity Data	<ul style="list-style-type: none"> The heart rate and power have been set into different levels with colour and range. 	Pass
T10	R1	View the sports heatmap	1)Click one of the activity 2)The heatmap was loaded success	Activity Data	<ul style="list-style-type: none"> The colours are loaded in the heatmap successfully. 	Pass
T11	R1	Apply additional functions (zoom in, zoom out, drag) for the heatmap.	1)Click one of the activity 2)Viewing the heatmap 3) Click the "+" button to zoom in and use the "-" button to zoom out 4) Using the mouse directly to drag the map.	Activity Data	<ul style="list-style-type: none"> The zoom-in function was loaded successfully. The zoom-out function was loaded successfully. The drag function was loaded successfully. 	Pass
T12	R1	View a range of data related to the activity through charts (by time).	1) Click one of the activities from the activity list. 2) Four different related charts(Heart Rate, Temperature, Elevation, and Pace/Speed) will be shown on the detailed page.	Activity Data	<ul style="list-style-type: none"> The Heart Rate Line chart was loaded successfully The Temperature Line chart was loaded successfully The Elevation Line chart was loaded successfully The Pace/Speed Line chart was loaded successfully 	Pass
T13	R1	Chart the graph data by distance.	1) Click the distance/time button on the detailed page 2) View the four different related charts.	Activity Data	<ul style="list-style-type: none"> The Heart Rate Line chart was loaded successfully The Temperature Line chart was loaded successfully The Elevation Line chart was loaded successfully The Pace Line chart was loaded successfully 	Pass
T14	R1	View the average value on each chart.	1) View the average value on each chart	Activity Data	<ul style="list-style-type: none"> The average value can be shown on the graph. 	Pass
T15	R1	View Stroke, Cadence and Power rate in Swiming and Cycling activity	1)Click one of the swimming or cycling activities. 2) View the Stroke, Cadence and Power rate chart.	Activity Data	<ul style="list-style-type: none"> The Stroke Rate Line chart was loaded successfully The Cadence Line chart was loaded successfully The Power Line chart was loaded successfully 	Pass
T16	R1	View all the necessary detailed data at the end of the page.	1) Click one of the activities from the activity list. 2) View the detailed data at the end of the detailed page.	Activity Data	<ul style="list-style-type: none"> The detailed data could be loaded successfully. It includes Distance, Calories, Heart Rate, Timing, Elevation, and Temperature. 	Pass

Test developer

LINGKANG ZHOU

Test of Login/Register page

Aim of the test:

Ensure that the login functionality is working as expected, and that the user can successfully log in to their account using valid credentials; ensure the register process is valid and people will be reminded that they should use valid passwords and email

Progress

[Add password constraints in frontend](#)

Requirement:

[Requirements 6 \(Looking for possible security issues\)](#)

Test specification

Test Case ID	Requirement	Test Case	Test Steps	Test Data	Expected Result	Pass/Fail
T01	R6	Register as a new user with an email address that has already been used	1) click on the register button 2) input user name, email and password	User Name: a Password: 1234aa AA! Email: 121@11 .com	<ul style="list-style-type: none">warning shows on the top of the page -"Username or Email already exists!"user cannot successfully register	Pass
T02	R6	Register as a new user with a non-registered email address	1) click on the register button 2) input user name, email and password	User Name: ab Password: 1234aa AA! Email: 12111@ 11.com	<ul style="list-style-type: none">successfully registered tip shows on the top of the page - "successfully registered"user can successfully register	Pass
T03	R6	Register as a new user with an invalid password	1) click on the register button 2) input user name, email and invalid password	User Name: aba Password: 1234 Email: 12211@ 11.com	<ul style="list-style-type: none">warning shows on the top of the page -'Password must longer than 8 characters'user cannot successfully register	Pass
T04	R6	Register as a new user with an invalid password	1) click on the register button 2) input user name, email and invalid password	User Name: aba Password: 11aA11 11 Email: 12211@11.com	<ul style="list-style-type: none">warning shows on the top of the page -'Password must contain at least one uppercase letter, one lowercase letter, one number, and one special character.'user cannot successfully register	Pass

T05	R6	Register as a new user with a valid password	1) click on the register button 2) input user name, email and valid password	User Name: aba Passwo rd: 11aA@ 1111 Email: 12211@1.com	<ul style="list-style-type: none"> successfully registered tip shows on the top of the page- "successfully registered" user can successfully register 	Pass
T06	R6	Login with user name and a valid password	1) input user name and valid password	User Name: ab Passwo rd: 1234aa AA!	<ul style="list-style-type: none"> successfully login tip shows on the top of the page- "Login success!" redirect to the dashboard 	Pass
T07	R6	Login with user name and a invalid password	1) input user name and invalid password	User Name: ab Passwo rd: 1234	<ul style="list-style-type: none"> failed login tip shows on the top of the page- "User is not existed or password is incorrect" 	Pass

Test developer

XiuyuanZhu

Test RegistryController

Aim of the test:

New users can register.

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.5 testRegister()	Test functionality and make sure new users can register	<ol style="list-style-type: none">Create a new user providing username, fullname, password, etc. in ideaRegister using the function and using logincheck function to log in the using the username and password.compare the response.	Data 0	After resigter function runs, the info used to register can be used by logincheck function and they return the same response.	Pass

Test developers:

Yuhang Yao, Yukun Li

Data:

Data 0:

```
String fullName="testCase";
String userName="testYyh";
String password="testYyhPassword";
String email="testYyh@gmail.com";
```

Test TokenDao

Aim of the test:

We can interact with the Token database.

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.9 testSaveRequestToken	we can insert the RequestToken data to database.	1. created data by our own and insert data. 2. get the data from api. 3. compare the content.	Data 0	the result of API is same as the result get by our creating	Pass
T02	R5.9 testFindByToken	we can get the token's information from database.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data 0	the result of API is same as the result get by our creating	Pass
T03	R5.9 testSaveAccessToken	we can insert the userAccessToken to the database.	1. created data by our own and insert it. 2. get the data from debase. 3. compare the content.	Data 1	the result of API is same as the result get by our creating	Pass
T04	R5.9 testDeleteAccessToken	We can use delete the specific userAccessToken from database.	1. created data by our own. 2. delete the data from api. 3. check if the data is null.	Data 1	After we delete, this specific data would be null	Pass

Test developers:

Yuhang Yao, Yukun Li

Data:

data 0: RequestToken reqToken = `new RequestToken("testUser", 123, "testToken", "testSecret", "Not Connected");`

data 1:

```
UserPartner testUser = new UserPartner();
testUser.setUsername("testUser");
testUser.setEmail("testUser@gmail.com");
testUser.setPassword("password");
testUser.setToken("token");
testUser.setTokenDate(new Date());
testUser.setUserAccessToken("userAccessToken");
testUser.setUserAccessSecret("userAccessSecret");
```

Test UserDao

Aim of the test:

We can interact with the User database.

Progress:

[Sprint 2 Fix API Test](#)

Requirement:

[Requirements 5 \(Test in sprint 2\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R5.10 testSaveUser	we can insert the user data to database.	1. created data by our own and insert data. 2. get the data from api. 3. compare the content.	Data 0	the result of API is same as the result get by our creating	Pass
T02	R5.10 findUserByUsername	we can get the user information from database by Username.	1. created data by our own. 2. get the data from api. 3. compare the content.	Data 0	the result of API is same as the result get by our creating	Pass
T03	R5.10 findUserByEmail	we can get the user information by his email from database.	1. created data by our own and insert it. 2. get the data from debase. 3. compare the content.	Data 0	the result of API is same as the result get by our creating	Pass
T04	R5.10 updateUser	We can update the user to database.	1. created data by our own. 2. delete the data from api. 3. check if the data is null.	Data 1	After we update, the user would be a different user.	Pass
	R5.10 deleteUserByUsername	we can delete user by specific username	1. create a new user. 2. check if the user has been inserted. 3. delete the user 4. check if the user has been deleted from database	Data 0	After we delete, this specific data would be null	Pass
	R5.10 testSelectUserByToken	we can get the user information by his token.	1. created data by our own and insert it. 2. get the data from debase. 3. compare the content.	Data 0	the result of API is same as the result get by our creating	Pass

Test developers:

Yuhang Yao, Yukun Li

Data :

data 0:

```
testUser = new UserPartner();
testUser.setUsername("testUser");
testUser.setEmail("testUser@gmail.com");
testUser.setPassword("password");
testUser.setToken("token");
testUser.setTokenDate(new Date());
testUser.setUserAccessToken("userAccessToken");
testUser.setUserAccessSecret("userAccessSecret");
```

data 1:

```
testUser = new UserPartner();
testUser.setUsername("testUser");
testUser.setEmail("testUser@gmail.com");
testUser.setPassword("password");
testUser.setToken("token");
testUser.setTokenDate(new Date());

testUser.setUserAccessToken("new access token");
testUser.setUserAccessSecret("new access token secret");
```

Test of Cloud front-end deployment

Aim of the test:

Ensure that the implemented Cloud front-end web works well

Progress

[Cloud front-end deployment](#)

Requirement:

[Requirements 11\(Cloud front-end deployment\)](#)

Test specification

Test Case ID	Requirement	Test Case	Test Steps	Test Data	Expected Result	Pass/Fail
T01	R1	Open the demo Url using a browser.	1)Input the demo Url into browser 2) Press Enter	https://gobox-jelly-ddbh.vercel.app/#/login	<ul style="list-style-type: none">Coaching Mate Login Form should load successfully	Pass
T02	R1	Log in Testing Account to view Activity List and then view the activity type with the icon, activity name and activity date for this activity.	1) Log in to the testing account. 2) Click the Activities button on the left side. 3) View the activity list.	User ID: 0402 Password AAAAaaa 1111	<ul style="list-style-type: none">The activity list can be shown on the right side.The event icon, event type, activity name and activity date were loaded successfully.	Pass
T03	R1	View the total distance for this type of activity.	1) View the total distance from the activity list. 2) Click one of the activities, and then view the total distance on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The total distance (km) could be viewed from the activity list.The total distance (km) could be viewed from the activity detailed page.	Pass
T04	R1	View the total time for this type of activity.	1) View the total time from the activity list. 2) Click one of the activities, and then view the total time on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The total time (xx:yy:zz) could be viewed from the activity list.The total time (xx:yy:zz) could be viewed from the activity detailed page.	Pass
T05	R1	View the average speed for this type of activity.	1) View the average speed from the activity list. 2) Click one of the activities, and then view the average speed on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The average speed(km/h) could be viewed from the activity list.The average speed(km/h) could be viewed from the activity detailed page.	Pass
T06	R1	View the ascent or descent distance for this type of activity.	1) View the ascent or descent distance from the activity list. 2) Click one of the activities, and then view the ascent or descent distance on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The ascent or descent distance (m) could be viewed from the activity list.The ascent or descent distance (m) could be viewed from the activity detailed page.	Pass
T07	R1	View the average heart rate.	1) View the average heart rate from the activity list. 2) Click one of the activities, and then view the average heart rate on the activity detailed page.	Activity Data	<ul style="list-style-type: none">The average heart rate could be viewed from the activity list.The average heart rate could be viewed from the activity detailed page.	Pass

T08	R1	View the activity in descending order	1) Click the activity button on the left side. 2) The activity is in descending order (latest).	Activity Data	<ul style="list-style-type: none"> The descending ordering was displayed success. 	Pass
T09	R1	View the sports route on the heatmap for this activity.	1) Click one of the activities from the activity list. 2) The heat map with the sports route will be shown on the detailed page.	Activity Data	<ul style="list-style-type: none"> The sports route was loaded successfully. The sports route has its starting point and ending point 	Pass
T10	R1	Different levels for heart rate and power with colour and number range.	1) Could be tested with T10 together or tested by console log.	Activity Data	<ul style="list-style-type: none"> The heart rate and power have been set into different levels with colour and range. 	Pass
T11	R1	View the sports heatmap	1)Click one of the activity 2)The heatmap was loaded success	Activity Data	<ul style="list-style-type: none"> The colours are loaded in the heatmap successfully. 	Pass
T12	R1	Apply additional functions (zoom in, zoom out, drag) for the heatmap.	1)Click one of the activity 2)Viewing the heatmap 3) Click the "+" button to zoom in and use the "-" button to zoom out 4) Using the mouse directly to drag the map.	Activity Data	<ul style="list-style-type: none"> The zoom-in function was loaded successfully. The zoom-out function was loaded successfully. The drag function was loaded successfully. 	Pass
T13	R1	View a range of data related to the activity through charts (by time).	1) Click one of the activities from the activity list. 2) Four different related charts(Heart Rate, Temperature, Elevation, and Pace/Speed) will be shown on the detailed page.	Activity Data	<ul style="list-style-type: none"> The Heart Rate Line chart was loaded successfully The Temperature Line chart was loaded successfully The Elevation Line chart was loaded successfully The Pace/Speed Line chart was loaded successfully 	Pass
T14	R1	Chart the graph data by distance.	1) Click the distance/time button on the detailed page 2) View the four different related charts.	Activity Data	<ul style="list-style-type: none"> The Heart Rate Line chart was loaded successfully The Temperature Line chart was loaded successfully The Elevation Line chart was loaded successfully The Pace Line chart was loaded successfully 	Pass
T15	R1	View the average value on each chart.	1) View the average value on each chart	Activity Data	<ul style="list-style-type: none"> The average value can be shown on the graph. 	Pass
T16	R1	View Stroke, Cadence and Power rate in Swimming and Cycling activity	1)Click one of the swimming or cycling activities. 2) View the Stroke, Cadence and Power rate chart.	Activity Data	<ul style="list-style-type: none"> The Stroke Rate Line chart was loaded successfully The Cadence Line chart was loaded successfully The Power Line chart was loaded successfully 	Pass
T17	R1	View all the necessary detailed data at the end of the page.	1) Click one of the activities from the activity list. 2) View the detailed data at the end of the detailed page.	Activity Data	<ul style="list-style-type: none"> The detailed data could be loaded successfully. It includes Distance, Calories, Heart Rate, Timing, Elevation, and Temperature. 	Pass

Test Reset Password Function

Aim of the test:

Users can reset passwords through email addresses.

Progress:

[Reset password function](#)

Requirement:

[Requirements 10\(Reset password function\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R10.1 Reset password function	Test users can receive email and use the link inside to reset password.	<ol style="list-style-type: none">1. Click forget password in the frontend.2. Enter the email that exists in the database3. Enter new password.4. Login Success with new password.	Email address	User can log in with new password and not with the old one.	Pass

Test Disconnect and Change Garmin Account

Aim of the test:

Users can disconnect their Garmin account and change another Garmin account using the same coachingmate account.

Progress:

[Disconnect Garmin Connect](#)

[Change Garmin connect account](#)

Requirement:

[Requirements 8\(Disconnect Garmin Connect\)](#)

[Requirements 9\(Change Garmin connect account\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R8.1 Disconnect Garmin account	Test users can disconnect their Garmin account.	1. Log in an account in the frontend. 2. Click disconnect button in the frontend. 3. Show button where users can connect.	Coachingmate account.	The user cannot see the previous activities and can click connect again.	Pass
T02	R9.1 Change Garmin connect account	Test users can change their Garmin account.	1. Log in an account in the frontend. 2. Click disconnect button in the frontend. 3. Click connect button and authorise again.	Coachingmate account & Garmin account.	The user can connect different account and see activities from it.	Pass

Test SMTP Service

Aim of the test:

Users can receive email.

Progress:

[Add SMTP email server to the backend](#)

Requirement:

[Requirements 13\(Add SMTP email server to the backend\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R13.1 coachingmateanalytics/coachingmate /service/email.java	Test users can receive email with the template.	1. Enter email address 2. Click run	Email address	User can receive email with correct content.	Pass

Test fixngt ransferring issue

Aim of the test:

test if the user can still see the account information in fronted consle when they login or register.

Progesse:

[Fix transferring security issue](#)

Requirment:

[Requirements 12\(Fix transferring_issue\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R13.1 coachingmateanalytics /coachingmate/controller/ LoginController.java	Test users' information should not exposed in frontend consle.	1. Enter username and password and login then 2. check from browser consle.	username: 0426 password: AAaa11 ..	the username and password shouldn't show in consle	Pass
T02	R13.1 coachingmateanalytics /coachingmate/controller/ RegistryController.java	Test users' information should not exposed in frontend consle.	1. Enter all their information. 2. check from browser consle.	username: 0525y password: AAaa11 ..	all information shouldn't show in the frontend.	pass

Test Spring Security Integration

Aim of the test:

test if user can access backend api without JWT.

test if user can get the JWT when they login.

Progress:

[Integrate with Spring Security](#)

Requirement:

[Requirements 14\(Integration with SpringSecurity\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R13.1 coachingmateanalytics /coachingmate /controller/ LoginController. java	User will receive a response that include a token .	1. Enter username and password and login then 2. check from browser to get see if the token is returned in response.	username: 0426 password: AAaa11 ..	The token should be in response.	Pass
T02	R13.1 coachingmateanalytics /coachingmate /controller/ ActivityDataRetri eveController. java	use postman to test if we can call the backend API without JWT.	1. Enter url: https://coach.myyylxy.com/activity/getActivityByAccessToken?accessToken=2d7c25b2-27f5-4974-a82f-2a9e41502b9f with JWT header: JWT 2. check from browser consle.	JWT below	we can get the successful message with the token. And we can get a error message "Need Auth" if we don't have token header.	pass

JWT:(would expire 1 day) eyJhbGciOiJIUzI1NiJ9.
eyJzdWIoIj1c2VySWQiLCJwYXNzd29yZC16liQyYSQxMCRrNGZGdXFmTUNIN2toRUNzUVhOdHVISHVMZWZxTy43Sy53cEZ3NVl3V2hiRWROUzhCe
mwYRylsInVzZXJuYW1lljoiMDQyNilsImhdCI6MTY4NDU2MzE4MCwianRpIjoiZWYwZGUxZjctNDdmNC00MmU4LWEzYWUtM2QzMjdmYThkMmU2liwiZ
XhwIJoxNjg0NjQ5NTgwfQ.vfDISDCFXDhJTAX9ckEf1UctJCEpMkld7alZTDpF8fo

Test Receiving Data from Garmin

Aim of the test:

Test if users can get real-time activity data from garmin accounts.

Progress:

[Receiving data from Garmin connect](#)

Requirement:

[Requirements 7\(Ensure accessibility\)](#)

Test specification:

Test Case ID	Requirement	Test case	Test Steps	Test Data	Expected Result	Pass /Fail
T01	R7.1 Receiving Data from Garmin Connect	Test user can get activity data from garmin connect.	1. Connect garmin from frontend. 2. Check if activity data is synced. 3. Add a new activity.	username: 0428 password: AAaa11 ..	Data is synchronized with garmin and can be added in real time	Pass

Troubleshooting articles

[Add troubleshooting article](#)

Former project problems	Lingkang Zhou	05 , 2023
Goals	Lingkang Zhou	22 , 2023
Background	Lingkang Zhou	22 , 2023

Former project problems

Process1 change the address of the database

Img1

```
GA-RedBack-main - application-dev.properties
Project: GA-RedBack-main
File: application-dev.properties
Line: 58

#oauthAccessToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/access_token
#callBackURL.url=https://coachInn-mate0121.herokuapp.com/auth/accessToken
requestToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/request_token
oauthConfirm.url=http://connect.garmin.com/oauthConfirm
oauthAccessToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/access_token
callBackURL.url=https://ga-boxjelly.herokuapp.com/auth/accessToken
#redis configuration
#spring.redis.database=0
#spring.redis.host=8.129.65.2
#spring.redis.port=6379
#spring.redis.password=717294
#spring.redis.timeout=3000ms

spring.data.mongodb.uri = mongodb://coachingmate:123456@127.0.0.1:27017/coachingmate
#mongodb for local configuration
#spring.data.mongodb.database=coachingmate
#spring.data.mongodb.port=27017

#spring.data.mongodb.uri = mongodb+srv://ga_boxjelly:Ga_boxjelly2022@cluster1.mjmt2.mongodb.net/coachingmate?retryWrites=true&w=majority

Terminal: Local
on [connectionId{localValue:1, serverValue:17}] to 127.0.0.1:27017
2023-03-18 12:04:53.346 INFO [cluster-ClusterId{value='64150e3542f5412b86e90070', description='null'}-127.0.0.1:27017] org.mongodb.driver.cluster.info:71 -Monitor thread successfully connected to server with description ServerDescription{address=127.0.0.1:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentsSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=1690084}
2023-03-18 12:04:53.865 INFO [main] o.s.scheduling.concurrent.ThreadPoolTaskExecutor.initialize:181 - Initializing ExecutorService
58-1 (84 chars) LF ISO-8859-1 4 spaces 2c
```

The location selected in img1 is changed to the address of the database, which is the address on the server.

Process2: backend port optional

```

GA-RedBack-main - application-dev.properties
Project  Current File  Version Control  Run  TODO  Problems  Terminal  Services  Profiler  Dependencies  Build
GA-RedBack-main / src / cm-backend / src / main / resources / application-dev.properties
1 #server.port=8080
2 server.port=${port:8080}
3 #
4 ## the type of the key store (JKS or PKCS12)
5 #server.ssl.key-store-type=pkcs12
6 #
7 ## the path to the key store that contains the SSL certificate.
8 #server.ssl.key-store=classpath:cert2.p12
9 #
10 ## the password used to access the key store
11 #server.ssl.key-store-password=password
12 #
13 ##the password used to access the key in the key store
14 #server.ssl.key-password=password
15 #
16 spring.swagger2.enabled=true
17 #
18 spring.banner.location=banner.txt
19 spring.devtools.restart.trigger-file=trigger.txt
20 #
21 spring.thymeleaf.cache=false
22 #
23 #Consumer Credentials
24 #oauth.consumerKey = d258c644-e0d8-4177-9d0c-b5b4478a2c95
25 #oauth.consumerSecret = 2qzSIibfjvQmY0XA2Uz0837cWE7HLxwoSB
26

Terminal: Local + 
on [connectionId{localValue:1, serverValue:17}] to 127.0.0.1:27017
2023-03-18 12:04:53.346 INFO [cluster-ClusterId{value='64150e3542f5412b86e90070', description='null'}-127.0.0.1:27017] org.mongodb.driver.cluster.info:71 -Monitor thread successfully connected to server with description ServerDescription{address=127.0.0.1:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=1690084}
2023-03-18 12:04:53.865 INFO [main] o.s.scheduling.concurrent.ThreadPoolTaskExecutor.initialize:181 -Initializing ExecutorService

```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (yesterday 9:44 pm) 2:1 (24 chars) LF ISO-8859-1 4 spaces

If the default port 8080 is occupied locally, it needs to be modified.

Img 2

```

GA-RedBack-main - request.js
Project  Current File  Version Control  Run  TODO  Problems  Terminal  Services  Profiler  Dependencies  Build
GA-RedBack-main / src / cm-frontend-react / src / utils / request.js  service
1 import axios from "axios";
2 import store from "@/store";
3 import { Modal } from "antd";
4 import { Form, Icon, Input, Button, message, Spin } from "antd";
5 import { getToken } from "@/utils/auth";
6 import { logout } from "@/store/actions";
7 //
8 // 创建一个axios示例
9 const service = axios.create({
10   // baseURL: process.env.REACT_APP_BASE_API, // api 的 base_url
11   //baseURL: "https://ga-redback-backend.herokuapp.com/", // api 的 base_url
12   baseURL: "http://localhost:8080",
13   timeout: 5000, // request timeout
14 });

15 // 请求拦截器
16 service.interceptors.request.use(
17   config => {
18     // Do something before request is sent
19     if (store.getState().user.token) {
20       // 让每个请求携带token-- ['Authorization']为自定义key 请根据实际情况自行修改
21       config.headers['X-Token'] = getToken()
22     }
23     return config;
24   },
25 );

```

Terminal: Local +

on [connectionId{localValue:1, serverValue:17}] to 127.0.0.1:27017
2023-03-18 12:04:53.346 INFO [cluster-ClusterId{value='64150e3542f5412b86e90070', description='null'}-127.0.0.1:27017] org.mongodb.driver.cluster.info:71 -Monitor thread successfully connected to server with description ServerDescription{address=127.0.0.1:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=1690084}
2023-03-18 12:04:53.865 INFO [main] o.s.scheduling.concurrent.ThreadPoolTaskExecutor.initialize:181 -Initializing ExecutorService

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (yesterday 9:44 pm) 11:2 (112 chars, 1 line break) LF UTF-8 2 spaces*

If the above back-end interface is changed, the front-end listening interface also needs to be changed, and the change location is shown in Img2.

If the local environment used is MAC, there is a problem with node18, and node 16 must be used.

The Pom file may also need to add version

as shown in the picture below

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.3.3.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>coachingmate-analytics</groupId>
  <artifactId>coachingmate</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>coachingmate</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
    <packaging>jar</packaging>
  </properties>

  <dependencies>
    <!--
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-mongodb</artifactId>
    </dependency>

    <dependency>
      <groupId>org.mybatis.spring.boot</groupId>
      <artifactId>mybatis-spring-boot-starter</artifactId>
      <version>2.1.3</version>
    </dependency>

    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <scope>runtime</scope>
    </dependency>
    -->

    <!-- https://mvnrepository.com/artifact/com.alibaba/fastjson -->

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!--
    <dependency>
      <groupId>org.springframework.boot</groupId>-->
      <artifactId>spring-boot-starter-security</artifactId>-->
    </dependency>-->

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>true</optional>
    </dependency>

    <dependency>
      <groupId>oauth.signpost</groupId>
      <artifactId>signpost-core</artifactId>
      <version>1.2.1.1</version>
    </dependency>
  
```

```

</dependency>
<!--
    <dependency>-->
    <groupId>com.google.api</groupId>-->
    <artifactId>api-common</artifactId>-->
    <version>2.1.5</version>-->
<!--
    </dependency>-->
<!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger2 -->
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.9.2</version>
</dependency>
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger-ui</artifactId>
    <version>2.9.2</version>
</dependency>

<!--
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
-->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.62</version>
</dependency>

<dependency>
    <groupId>com.darwinsky</groupId>
    <artifactId>hirondelle-date4j</artifactId>
    <version>1.5.1</version>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-aop</artifactId>
</dependency>

<!-- mongodb -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-mongodb</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
    <groupId>com.sun.mail</groupId>
    <artifactId>jakarta.mail</artifactId>
    <version>2.0.1</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
    <exclusions>
        <exclusion>
            <groupId>org.junit.vintage</groupId>
            <artifactId>junit-vintage-engine</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<!-- swagger API-->
<!--
    <dependency>-->
    <groupId>com.spring4all</groupId>-->
    <artifactId>swagger-spring-boot-starter</artifactId>-->
    <version>1.9.1.RELEASE</version>-->
<!--
    <exclusions>-->
<!--
    <exclusion>-->
    <groupId>io.springfox</groupId>-->
    <artifactId>springfox-swagger-ui</artifactId>-->
<!--
    </exclusion>-->
-->
```

```

<!--      </exclusions>-->
<!--      </dependency>-->

<!--      <dependency>-->
<!--          <groupId>com.github.xiaoymin</groupId>-->
<!--          <artifactId>swagger-bootstrap-ui</artifactId>-->
<!--          <version>1.9.6</version>-->
<!--      </dependency>-->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <scope>test</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->
<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.4</version>
</dependency>

<dependency>
    <groupId>com.xxx.www</groupId>
    <artifactId>out-jar-2</artifactId>
    <version>1.0.0</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/lib/FitReader-1.1.jar</systemPath>
</dependency>
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.20</version>
</dependency>
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.9.0</version>
</dependency>
<!--      <dependency>-->
<!--          <groupId>jakarta.validation</groupId>-->
<!--          <artifactId>jakarta.validation-api</artifactId>-->
<!--          <version>7.0</version>-->
<!--      </dependency>-->

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
                <includeSystemScope>true</includeSystemScope>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>2.19.1</version>
        </plugin>
    </plugins>
</build>

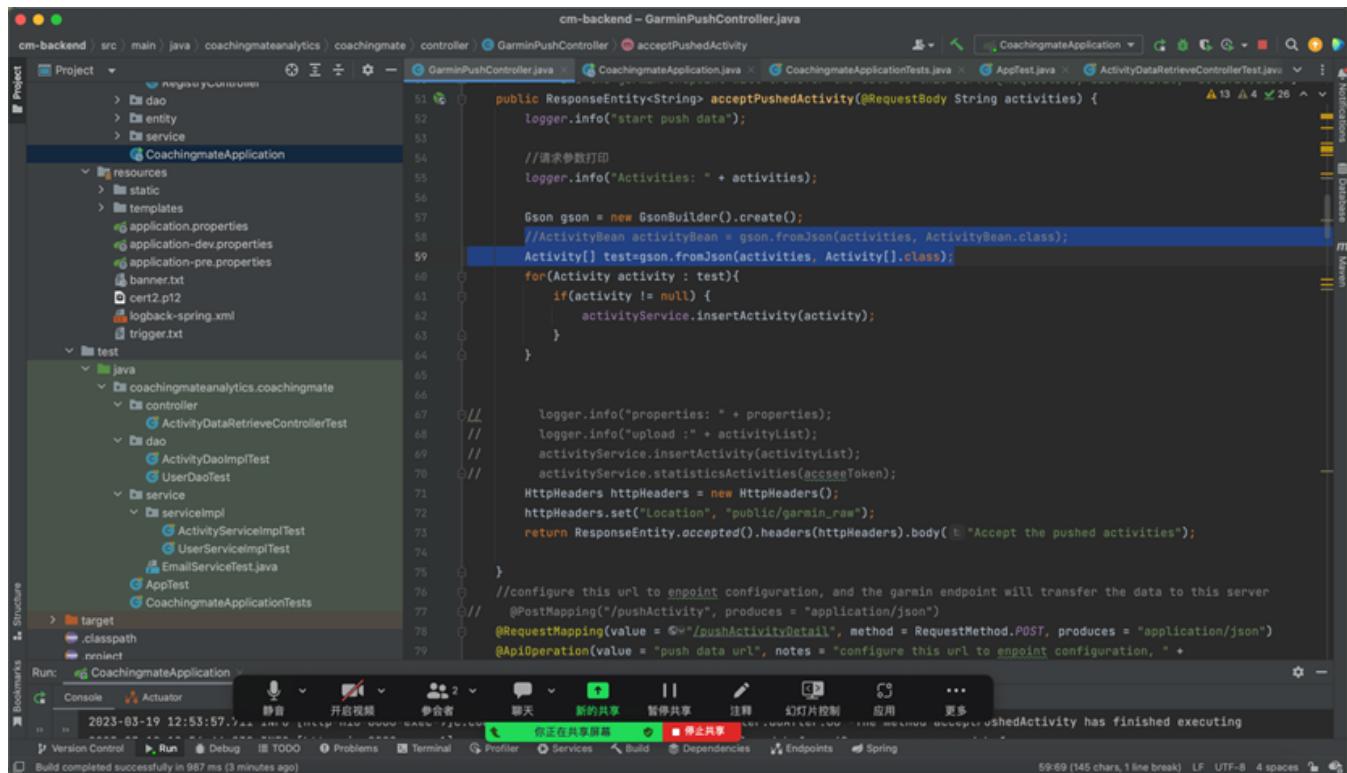
</project>

```

The above is the available pom file, and two versions have been changed, but it may be different due to different local environments.

call_back_url of Dev- property;

Data_samples_backend/backenddata/regisgtry-controller To enter the website here, you need to change the push URL.



```
cm-backend - GarminPushController.java
cm-backend > src > main > java > coachingmateanalytics > coachingmate > controller > GarminPushController > acceptPushedActivity
public ResponseEntity<String> acceptPushedActivity(@RequestBody String activities) {
    logger.info("start push data");

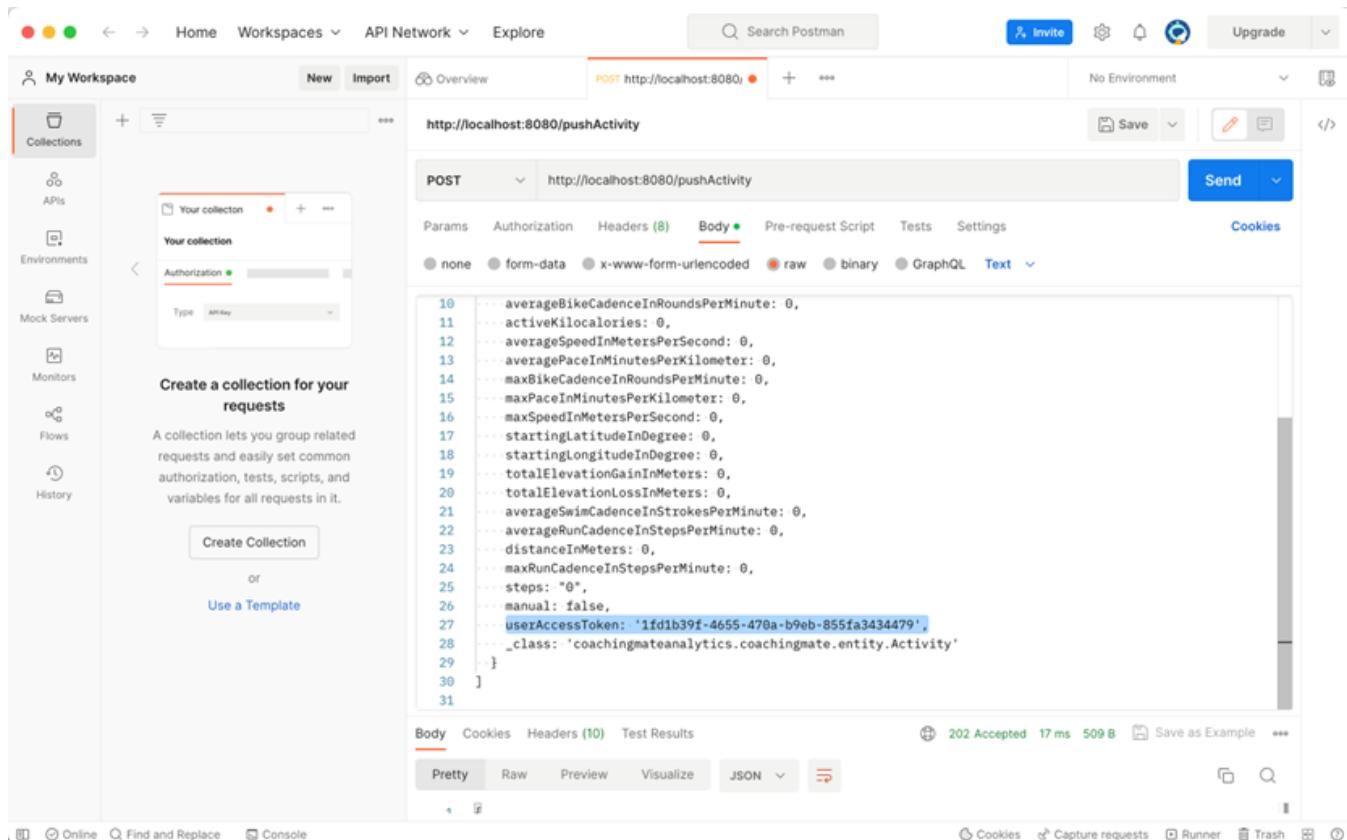
    //请求参数打印
    logger.info("Activities: " + activities);

    Gson gson = new GsonBuilder().create();
    //ActivityBean activityBean = gson.fromJson(activities, ActivityBean.class);
    Activity[] test=gson.fromJson(activities, Activity[].class);
    for(Activity activity : test){
        if(activity != null) {
            activityService.insertActivity(activity);
        }
    }

    logger.info("properties: " + properties);
    logger.info("upload :" + activityList);
    activityService.insertActivity(activityList);
    activityService.statisticsActivities(accesstoken);
    HttpHeaders httpHeaders = new HttpHeaders();
    httpHeaders.set("Location", "public/garmin_raw");
    return ResponseEntity.accepted().headers(httpHeaders).body("Accept the pushed activities");
}

//configure this url to endpoint configuration, and the garmin endpoint will transfer the data to this server
@PostMapping("/pushActivity", produces = "application/json")
@RequestMapping(value = "/pushActivityDetail", method = RequestMethod.POST, produces = "application/json")
@ApiOperation(value = "push data url", notes = "configure this url to endpoint configuration, ")
@ApiOperation(value = "pushedActivity has finished executing")
```

Gson conversion problem



My Workspace

POST http://localhost:8080/pushActivity

Body

```
averageBikeCadenceInRoundsPerMinute: 0,
activeKilocalories: 0,
averageSpeedInMetersPerSecond: 0,
averagePaceInMinutesPerKilometer: 0,
maxBikeCadenceInRoundsPerMinute: 0,
maxPaceInMinutesPerKilometer: 0,
maxSpeedInMetersPerSecond: 0,
startingLatitudeInDegree: 0,
startingLongitudeInDegree: 0,
totalElevationGainInMeters: 0,
totalElevationLossInMeters: 0,
averageSwimCadenceInStrokesPerMinute: 0,
averageRunCadenceInStepsPerMinute: 0,
distanceInMeters: 0,
maxRunCadenceInStepsPerMinute: 0,
steps: "0",
manual: false,
userAccessToken: '1fd1b39f-4655-470a-b9eb-855fa3434479',
_class: 'coachingmateanalytics.coachingmate.entity.Activity'
```


Project review

Sprint 2:

[Sprint2_Review](#)

Sprint 3:

[Sprint3_Review](#)

Sprint2_Review

Review Detail

Date

2023-4-28

Attendees

- Lingkang Zhou
- Rui Liu
- Yuhang YAO
- Yukun Li
- @ Hanna Navissi

Sprint Overview:

Sprint start:

27 Mar 2023

Sprint end:

01 May 2023

Sprint goals:

According to the information obtained from the meeting with Hanna, the back-end code of the project is sorted out and cleaned, and the potential security problems in the current back-end, as well as the existing obvious bugs are initially repaired and test and document records are set.

Completed Work:

Task 1

Analyze the feedback received from sprint1 to specify adjustments to the project schedule

User story:

As a project manager, I want to analyze the feedback received from sprint 1 in order to identify areas for improvement and adjust the project schedule accordingly. By carefully reviewing the feedback and making necessary adjustments, I can ensure that the project stays on track and that team members are able to work effectively and efficiently. This will help to ensure that the project is completed on time and within budget, while also ensuring that the team is able to deliver a high-quality product that meets the needs of stakeholders.

Task 2

Complete the writing of the user guide on the project onboarding page of confluence.

User story:

As a technical writer, I want to complete the writing of the user guide for the project onboarding page in Confluence so that team members can easily understand and navigate the project. By providing clear and concise instructions on how to use the project's features and tools, I can help team members quickly get up to speed and contribute to the project's success.

Completion status:

Finished

[Project Development Environment Setting Guide](#)

Task 3

looking for possible security issues

User story:

As a software developer, I want to proactively search for and identify any potential security issues in the project so that I can ensure that the code is secure and protected against potential threats. By conducting thorough security reviews and identifying vulnerabilities in the code, I can take appropriate measures to address them and prevent security breaches from occurring. This will help to safeguard the project and protect sensitive data and information from being compromised.

Task 4

Setting up tests

User story:

As a software developer, I want to be able to set up automated tests for the project so that I can ensure that the code is functioning correctly and catch any errors or bugs early in the development process. This will help me to improve the quality of the code and reduce the likelihood of issues arising in the future.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass [Test ActivityDataRetrieveController](#)

Track :[Progess-Sprint 2 Fix API Test](#)

Task 5

Finish the Code-peer review activity

User story:

As a software developer, I want to complete the code peer review activity for the project so that I can ensure that the code meets the required standards and is free of errors or bugs. By reviewing my colleagues' code and having my own code reviewed, I can identify areas for improvement and learn from others in the team. This will help me to improve the quality of the code and contribute to the overall success of the project.

Acceptance criteria and completion status:

Finished

[Peer-to-peer code review](#)

Task 6

Communication with Hanna about push activity problems.

User story:

As a software developer, I want to communicate with Hanna in order to solve any issues related to the push activity. By understanding the solution to specific problems from Hanna, I can ensure that the program runs smoothly and effectively for all users. This will improve the user experience and satisfaction with the program, ultimately contributing to the success of the project.

Task 7

Improve do-be-feel list &Motivational model according to the sprint1 feedback.

User story:

As a UX designer, I want to improve the do-be-feel list and motivational model based on the feedback received during sprint 1. By carefully reviewing the feedback and making necessary adjustments, I can ensure that the user experience is optimized and that users are motivated to engage with the product. This will improve user satisfaction and ultimately contribute to the success of the project.

Acceptance criteria and completion status:

Finished

Task 8

Github standard-Version control & Merge the current function branches.

User story:

As a software developer, I want to ensure that the project is following the standard GitHub version control practices and merge the current function branches. By following best practices for version control, such as maintaining a main branch for stable releases and using feature branches for development work, I can ensure that the project is well-organized and easy to manage. Additionally, by merging the current function branches, I can ensure that the latest code is integrated and that any conflicts or issues are resolved. This will improve the overall stability and reliability of the codebase, ultimately contributing to the success of the project.

Acceptance criteria and completion status:

Finished

[Project Github Repositories](#)

Task 9

Create detailed plan page as Sprint1 feedback asked.

User story:

As a project manager, I want to create a detailed plan page as requested by the Sprint 1 feedback. By creating a detailed plan page, I can ensure that all team members have a clear understanding of project goals, timelines, and tasks. This will improve team communication and collaboration, and help to ensure that the project stays on track and within budget. This will contribute to the success of the project and ensure that all stakeholders are satisfied with the outcome.

Acceptance criteria and completion status:

Finished

[Project Planning](#)

Task 10

Fix the backend API test case

User story:

As a software developer, I want to fix the backend API test case to ensure that the API functions correctly and as expected. By fixing the test case, I can ensure that any errors or issues are identified and resolved before the software is deployed to production. This will improve the overall quality and reliability of the software application and reduce the risk of bugs or errors causing issues for users. This will contribute to the success of the project by ensuring that the software application meets the needs and expectations of users and stakeholders.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test ActivityDataRetrieveController](#)

Track: [Sprint 2 Fix API Test](#)

Task 11

Fix up document

User story:

As a team member, I want to fix up the document to ensure that it is accurate, up-to-date, and reflects the current state of the project. By fixing up the document, I can ensure that all stakeholders have access to the information they need to make informed decisions and contribute to the success of the project. This will improve communication and collaboration among team members, reduce the risk of misunderstandings or confusion, and ensure that everyone is working towards the same goals.

Acceptance criteria and completion status:

Finished

Task 12

Code cleaning & code restructuring

User story:

As a software developer, I want to clean up and restructure the code to improve its readability, maintainability, and scalability. By cleaning up and restructuring the code, I can ensure that it is well-organized, easy to understand, and follows best practices and coding standards. This will make it easier to make changes to the code in the future, reduce the risk of bugs or errors, and improve the overall quality and reliability of the software application.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test of Code Cleaning](#)

Track: [Code cleaning](#)

Task 13

Add constraint rules to password setting process

User story:

As a user, I want to be able to set a strong and secure password for my account. Therefore, I want the password setting process to include constraint rules that ensure my password meets certain requirements, such as minimum length, including a mix of uppercase and lowercase letters, numbers, and special characters. By including these constraint rules, I can be confident that my account is secure and protected from potential unauthorized access. This will help to improve the overall security and integrity of the software application, and ultimately enhance the user experience and satisfaction.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test of Login/Register page](#)

Track: [Add password constraints in frontend](#)

Task 14

Solving the problem of using plain text in the user login backend

User story:

As a user, I want my login credentials to be stored in a secure and encrypted format in the backend system. Therefore, I expect the backend system to use industry-standard encryption algorithms to securely store my password in a hashed format, rather than storing it in plain text. By doing so, I can be confident that my login credentials are safe and secure, and that my personal information is protected from potential security breaches. This will help to enhance the overall security and trustworthiness of the software application, and ultimately improve the user experience and satisfaction.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the

Track: [Encrypt password](#)

Task 15

Fix the bug that same email address would cause the error instead of handling the exception.

User story:

As a user, I want the system to handle exceptions properly when the same email address is used to prevent errors, so that I can use the application without any issues.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the [Test of Login/Register page](#)

Track: [Check Availability for Username and Email](#)

Task 16

Solve the problem of map in the activity page cannot work.

User story:

As a user, I want the map in the activity page to be fixed so that I can view my workout route and track my progress accurately.

Acceptance criteria and completion status:

Finished

Acceptance criteria: Pass the

Track: [Fix the map in frontend.](#)

Acceptance criteria and completion status:

Total number of completed tasks: 16

Incomplete Work:

Task 1

Add the function of disconnect Garmin Connect.

User story:

As a user, I want to be able to disconnect my Garmin Connect account from the application, so that I can stop sharing my data or switch to another account if needed. This can be achieved by adding a button or option in the settings page that allows me to disconnect my Garmin Connect account and remove any related data from the application. Additionally, there should be a confirmation message or prompt to ensure that I am aware of the consequences of this action and that I want to proceed.

Task 2

Make users be able to change to another connect account

User story:

As a user, I want to be able to disconnect my current Garmin Connect account and connect to a different one, so that I can use my preferred Garmin Connect account to track my activities on the platform. The new feature should allow me to easily disconnect my current account and provide a straightforward way to connect to another Garmin Connect account.

Task 3

Solve the problem of receiving data from Garmin connect

User story:

As a user, I want to be able to receive data from Garmin Connect so that I can easily access and analyze my fitness data without having to manually input it into the system.

Track :

Progress-Receiving data from Garmin connect

Task 4

Add reset password function

User story:

As a user, I want to be able to reset my password in case I forget it or need to update it for security reasons. By adding a reset password function, I can easily recover access to my account and ensure the security of my personal information. This will improve the user experience and increase trust in the software application. This will contribute to the success of the project by ensuring that users have a positive experience with the application and are satisfied with its functionality.

Total number of incompletely completed tasks: 4

Review Key Metrics for the Sprint:

Comparison of anticipated workload vs. actual completed workload:

87 story points / 61 story points

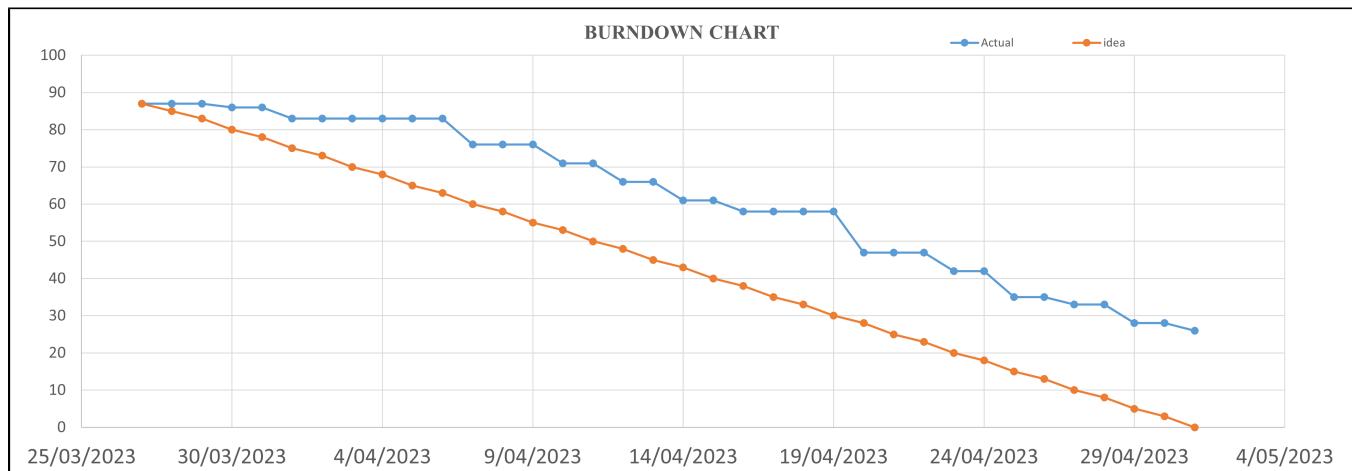
Team velocity (story points completed during the Sprint):

61 story points /sprint

Estimated remaining workload (incomplete story points):

26 story points /sprint

Burndown chart



Team Members' Reflection:

LingkangZhou: Due to the Garmin Connect API issue limited, we have been unable to solve the problem of receiving Garmin connect activity data. This makes our subsequent API development very limited. But this problem was not solved in time, although to a certain extent it was due to the inability to contact the client, but it was more because we did not clarify who the client should be communicated with. So in the next sprint, we will actively negotiate with the client to understand the problem.

Yukun Li: The process of merging branches was tough. There were too many conflicts from different functionalities who were completed in different branches. We should have organised our branches in a way that's easy for us to deal with conflicts. For example, we can inform others that some branch has been updated and encourage peers to merge with their own branches. In this way, it could save much time when we combine all branches together as the default branch and bugs caused by merging can be easily fixed as well.

Rui Liu: During the process of testing, we encountered some issues, such as it may be difficult to reproduce the defects or problems encountered during testing, which may lead to insufficient test coverage, and limited resources such as hardware or personnel can affect testing process and its coverage. Our team had a meeting about this and came up with some workarounds to improve our testing process. First, there needs to be effective communication and documentation to record issues in real-time, helping to reproduce defects and resolve issues. Meanwhile, there must be comprehensive test coverage to identify and solve all problems, and we meet the requirements by enhancing work efficiency and trying more corner case tests to achieve comprehensive testing.

Yuhang Yao: During the sprint, by our good collaboration, we have developed the baseline application on each members' local and a server. The challenge in this sprint is that there is few document and test function on existing code, so we have to spent much time understanding the previous code , write testing function and fixing the bug. Then we don't have much time to implement new features. Hence, for the improvement, we can do more works on new features and enhance the code documentation.

XiuyuanZhu: During this sprint there are some difficulties via get data from backend and properly display it in the page. For example, we cannot get map details showed on page and it is hard for us to debug in the console. I find it really important to document all the progress and details because if we have a proper document from the past team there will not be all the fuss. Hence, to document the progress and all the apis used in code can be important and helpful.

Planning for the Next Sprint:

- Prioritization of incomplete tasks
- [Sprint 3 Plan](#)
- Estimated start and end dates for the next Sprint

Start from 2023-5-2 end at 2023-5-28

Sprint3_Review

Review Detail

Date

2023-5-26

Attendees

- Lingkang Zhou
- Rui Liu
- Yuhang YAO
- Yukun Li
- @ Hanna Navissi

Sprint Overview:

Sprint start:

2023-5-1

Sprint end:

2023-5-26

Sprint goals:

According to the needs of customers, further develop the functions of the back-end API. It is planned to solve the task last sprint left especially receiving the activity data from Garmin connect (Major task) and add the session security oauth feature into backend if time permitted.

Completed Work:

Task 1

Add the function of disconnect Garmin Connect.

User story:

As a user, I want to be able to disconnect my Garmin Connect account from the application, so that I can stop sharing my data or switch to another account if needed. This can be achieved by adding a button or option in the settings page that allows me to disconnect my Garmin Connect account and remove any related data from the application. Additionally, there should be a confirmation message or prompt to ensure that I am aware of the consequences of this action and that I want to proceed.

Track :

[Disconnect Garmin Connect](#)

Task 2

Make users be able to change to another connect account

User story:

As a user, I want to be able to disconnect my current Garmin Connect account and connect to a different one, so that I can use my preferred Garmin Connect account to track my activities on the platform. The new feature should allow me to easily disconnect my current account and provide a straightforward way to connect to another Garmin Connect account.

Track :

[Change Garmin connect account](#)

Task 3

Solve the problem of receiving data from Garmin connect

User story:

As a user, I want to be able to receive data from Garmin Connect so that I can easily access and analyze my fitness data without having to manually input it into the system.

Track :

[Receiving data from Garmin connect](#)

Task 4

Add reset password function

User story:

As a user, I want to be able to reset my password in case I forget it or need to update it for security reasons. By adding a reset password function, I can easily recover access to my account and ensure the security of my personal information. This will improve the user experience and increase trust in the software application. This will contribute to the success of the project by ensuring that users have a positive experience with the application and are satisfied with its functionality.

Track :

[Reset password function](#)

Task 5

Cloud front-end deployment

User story:

As a user, I want to access the application quickly and securely through a cloud-based front-end deployment so that I can enjoy a reliable and responsive experience regardless of my location. By deploying the application's front-end on a cloud platform, I can benefit from its global infrastructure, scalability, and performance improvements. This will contribute to the success of the project by providing a seamless user experience and ensuring that the application can handle a growing user base and increased demand.

Track :

[Cloud front-end deployment](#)

Task 6

Address password transfer security issues

User story:

As a user, I want to ensure that my password is securely transferred between the client and the server so that my sensitive information remains protected. By addressing the password transfer security issues, I can ensure that my personal data is not compromised and that I can trust the application with my sensitive information. This will contribute to the success of the project by enhancing the overall security and reliability of the application, which will attract more users and improve their trust in the product.

Track :

[Fix transferring security issue](#)

Task 7

Add an email server to the backend

User story:

As a system administrator, I want to add an email server to the backend, so that the system can send emails to users for notifications and alerts. This will improve the communication with users and enhance the overall user experience. By integrating an email server into the backend, the application will become more robust and flexible, and enable users to communicate with the system in a more efficient and effective manner. This will contribute to the success of the project by improving the overall user experience and increasing user engagement with the application.

Track :

[Add SMTP email server to the backend](#)

Task 8

ChatGPT code review

User story:

As an developer, I want to have my code reviewed by ChatGPT, so that I can ensure that my code is high quality and meets the company's coding standards. By having a code review process in place, I can identify and fix potential issues early on, avoid technical debt, and improve the overall codebase of the project. This will contribute to the success of the project by increasing its stability, reducing the likelihood of bugs and errors, and enabling faster and more efficient development in the long run.

Track :

[ChatGPT code review](#)

Task 9

Integrate with Spring Security-Auth

User story:

As a user, I want the application to solve the authentication issue so that I can access it securely and without any hassles. By addressing the authentication issue, I can have peace of mind knowing that my personal information and data are protected. This will contribute to a better user experience by ensuring that only authorized individuals can access the application and its features. It will also enhance the overall security of the system, preventing unauthorized access and potential breaches. Resolving the authentication issue will instill trust and confidence in the application, attracting more users and contributing to its success.

Track :

[Integrate with Spring Security](#)

Total number of completed tasks: 9

Incomplete Work:

Null

Total number of incompletely tasks: 0

Review Key Metrics for the Sprint:

Comparison of anticipated workload vs. actual completed workload:

71 story points / 71 story points

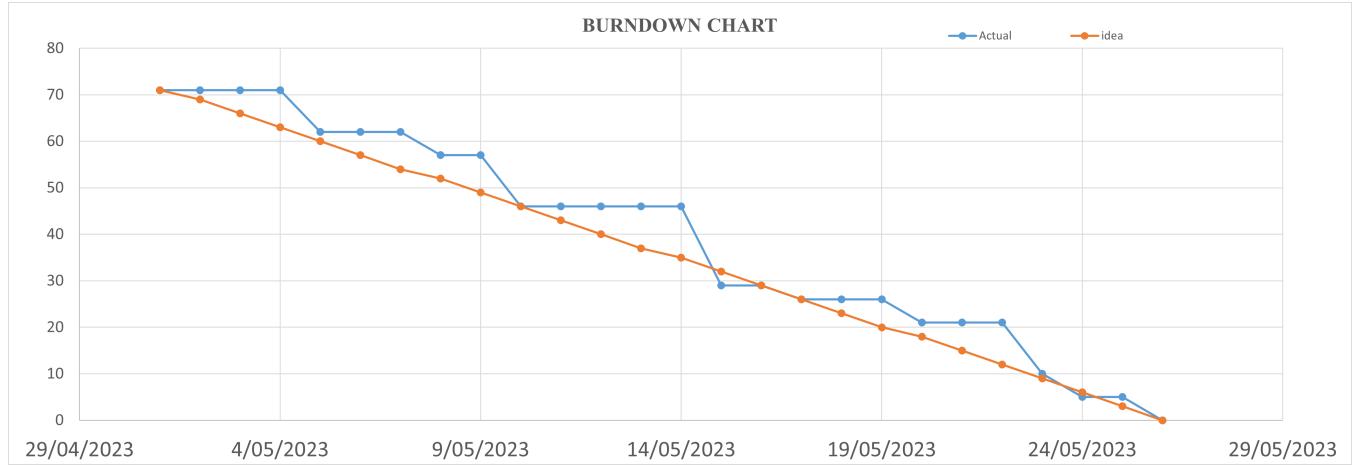
Team velocity (story points completed during the Sprint):

71 story points / sprint

Estimated remaining workload (incomplete story points):

0 story points /sprint

Burndown chart



Team Members Reflection:

Lingkang Zhou: In sprint3, we adjusted the overall status and strategy of the team, giving priority to completing the unfinished user stories of sprint2. And actively communicate with customers and hold technical analysis meetings. So, in sprint3, we completed all planned tasks. The execution of the plan has been greatly improved, so that we can well fulfill the requirements and expectations of customers. and smoother team collaboration. Generally speaking, compared with sprint2, the overall working efficiency of the team has been greatly improved, and I am very satisfied with our current team.

Yukun Li: We have made a progress in the sprint. Pretty much all planned tasks have been well finished and some of them were done in advance. The development team engaged in daily meetings so that everyone can keep up with the rate of progress. The product has been deployed as expected in this sprint and everyone contributed efforts and time to make it happen. We also had more ideas about the project although there might not be enough time. This sprint has been a great success compared to the sprints before and it produced a satisfying outcome as expected.

Rui Liu: During Sprint 3, we prioritized completing unfinished user stories from Sprint 2. We communicated with customers and held technical analysis meetings, resulting in successful task completion, improved execution, and meeting client requirements. Compared to Sprint 2, team efficiency improved, leading to smoother collaboration. The supervisor is satisfied with team performance. Daily meetings ensured progress transparency. The product was deployed as planned, with the team's commitment. Time constraints limited new idea exploration, but the sprint achieved satisfactory results, surpassing previous sprints. In this sprint, we completed main product features. Despite learning Java and Spring Security, they gained valuable web application development experience. In the future, next sprint will focus on product refinement and identifying areas for improvement based on lessons learned.

Yuhang Yao: In this sprint, we have finished the main features of the product. By the experience, although it's hard for us to meet the new tech and apply it, we have learned a lot about the Java, Spring Security and other web application development. Besides, we also have improvement on communication and cooperation. In summary, we work well in the sprint, and we think we can do better in the next final sprint. In the next sprint, we will focus on the polish of the product, and we will also check what we can improve for the project based on this sprint.

Xiuyuan Zhu: During sprint3 we have done a lot more work since the requirements from clients have been more specified. Our goal is to enhance the security and integrity of our code and do more tests after deployment to ensure the quality of the application under different conditions. I personally have learned a lot about how to use the React components and cooperate more with the backend developers. In the next sprint we will focus more on how to deliver the project to the next team better by checking the integrity of documentation.

Planning for the Next Sprint:

- Polish the product
- Finish the final product handover

Start from 2023-5-28 end at 2023-6-9