IPTK Project Report

Sword Runner
15.07.2021
Group L: Mofan Guo, Zhuo Yu, Weihao Chen, Jiaqi Liu, Reiber Hannes, Petsch Pascal

TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Motivation

With the development of mobile communications and the unification of the Internet to mobile terminals, the network and users have increasingly higher requirements for mobile terminals. With the emergence of smart phone terminals, more rapid development platforms have been provided for video games, especially the appearance of iPhone smart phones and Android smart phones, which provides a more complete development environment. Nowadays, mobile video games have also become an indispensable part of people's lives. And at present, the market for the Android platform has leapt to number one in the world, and more people have joined the Android platform application development platform. Google launched a free online map service in 2005. Google also provides a basic technology API interface to facilitate more users to develop map services on their own. As map technology matures, user-based mobile games have also sprung up and appeared at the same time. Such as Pokémon GO is an augmented reality (AR) mobile game based on Location-based service.

After 2020, the Covid-19 has swept the world. The lifestyle of the people that followed have significant changes. Not only Germany, many countries in the world have promulgated a series of policies to deal with the Covid-19, and Lockdown is also one of them. The company started to let people start Home Office, and more and more people began to change habits and customs, stop going out, and began to get used to ordering takeaways and unhealthy lives, even if people can go out of the house and do jogging. Sword Runner, represented in this paper, is a based-on Location-based service exergame in Android designed for motivating people to go out and run more through the use of an original, user-friendly game story. The user needs to run daily to complete the goals set in the game. The game will record the user's running path and total distance, and then display it to the user on the smartphone screen. This paper will introduce more features about this game.

2. Technical documentation

2.1 Concepts

Sword Runner is an Exergame in a post apocalyptic setting. The players get motivated to go running on a daily bases, to keep their villages alive. The goal for the players is, to stay alive as many days as possible in a row. Through being able to set the daily running distance on different amounts, we on the one hand target people that go running on a regular basis, but also people that want to get started with running and need some help to get motivated.
In times of lockdowns many aren't able to practice their team sports or go to the gym, motivate them to go running can help to prevent health problems. In this project, we six members as a team finished an android based game which use osmdroid, nodejs,

mongodb. It is a front-end separation project. It contains features including secure login, profile system, friend system, single-player and multiple-players system. Furthermore, we designed a comment system to let users appraise each other. Also, players will earn coins with each finished run. The coins can be spent to skip a day of running and still keep the village alive.

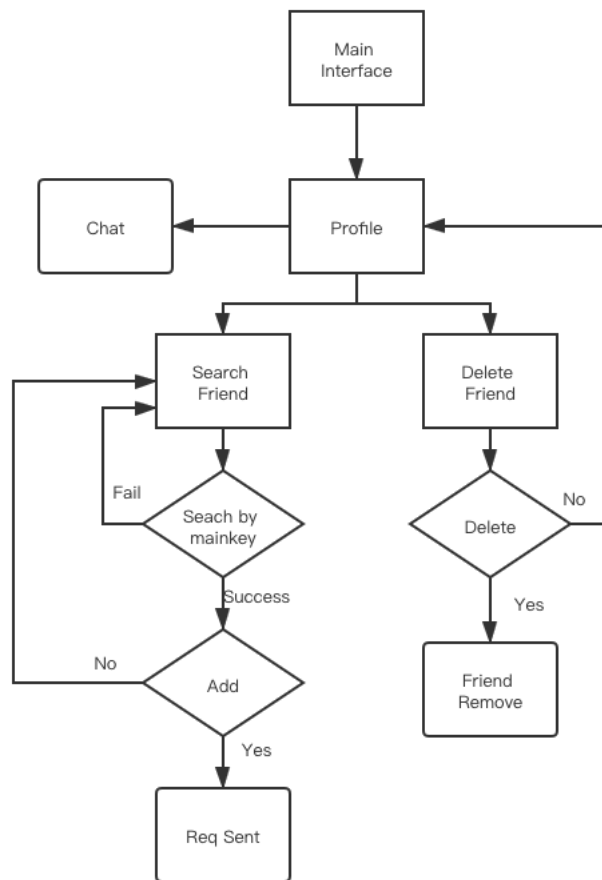## 2.2 Architectural Overviews

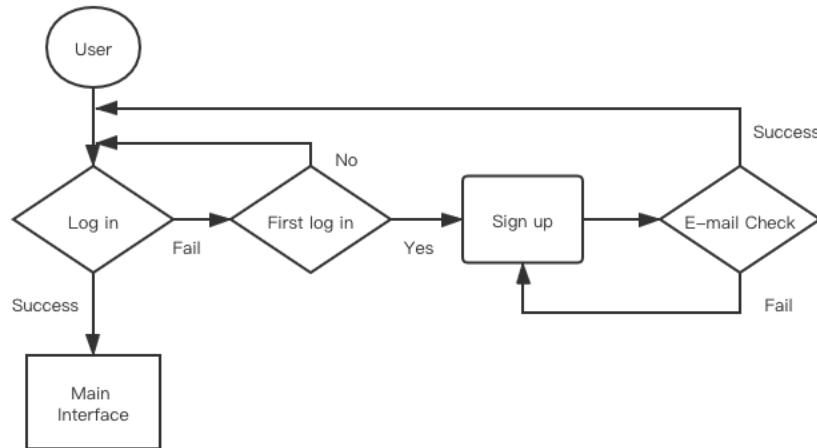2.2.1 Use logic, flow diagram

Figure 2.2.1.1 Friend Flow Diagram
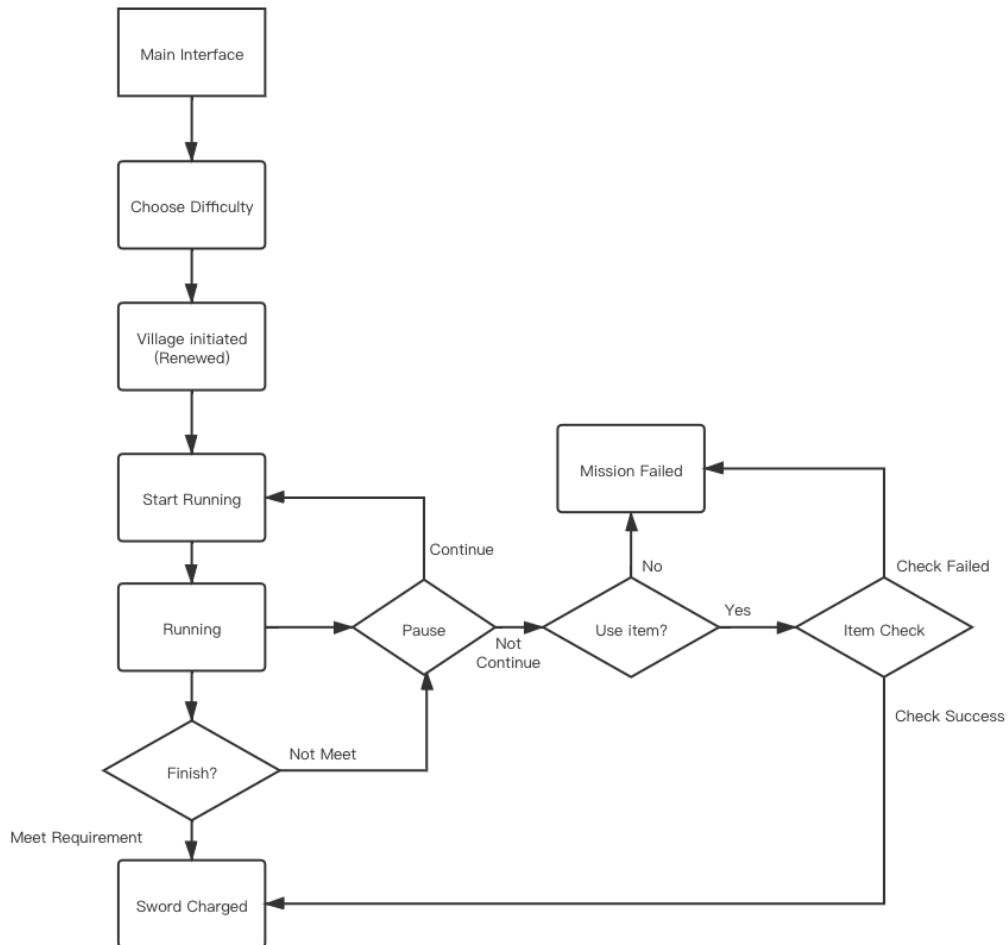
Figure 2.2.1.2 Log in Flow Diagram

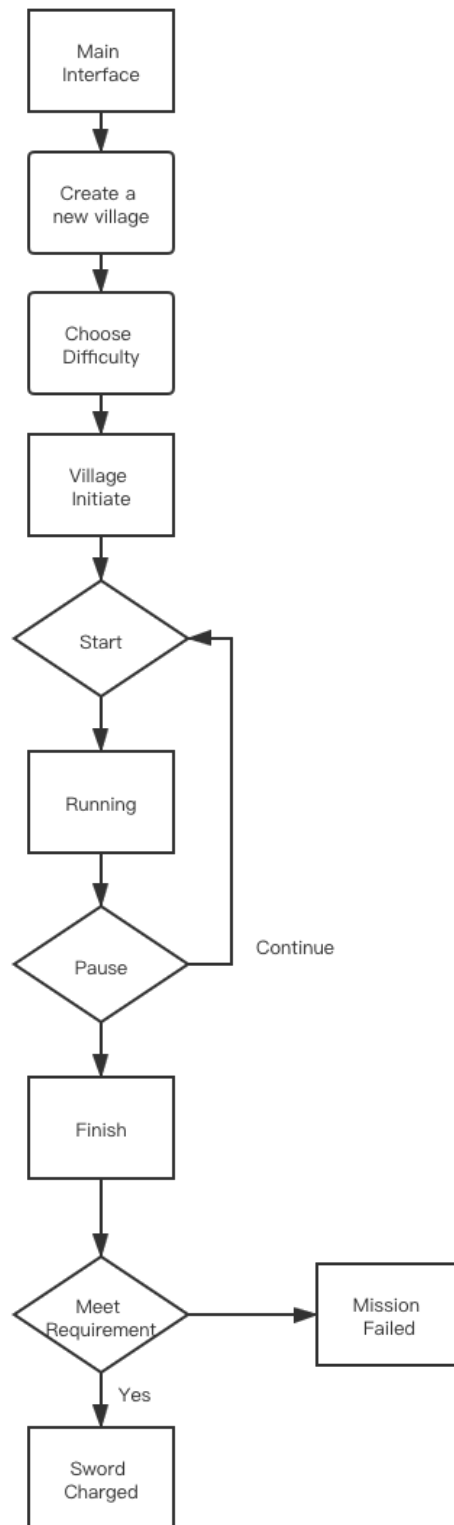Figure 2.2.1.3 Single-Player Flow Diagram

Figure 2.2.1.4 Multi-Player Flow Diagram

2.2.2 Entities

Friend: friendId, userId, friendEmail, userEmail, friendName, friendAvatar, username, userAvatar, lastSentence, countingDays, chattingFile, days.

User: name, email, password, avatar, date, comments, tokens, coins.

Game: userId, period, distance, progress, started_at, status.

Message: senderId, receiverId, totalId, message, time, date.

Map: userId, createdAt, startTime, endTime, startLocation, endLoction, route, distance, duration.

2.3 Used technologies

2.3.1 Design Pattern

In this project, we use MVC Design pattern. The full name of MVC is Model View Controller. What is defined in the MVC framework is that Model is the digitization of the object model in real life, which contains data and data processing methods. The View displays the data in the Model and provides it to the user, and the Controller is the bridge between the two. After accepting the user's input operation, the result is passed to the View after the corresponding operation is completed.
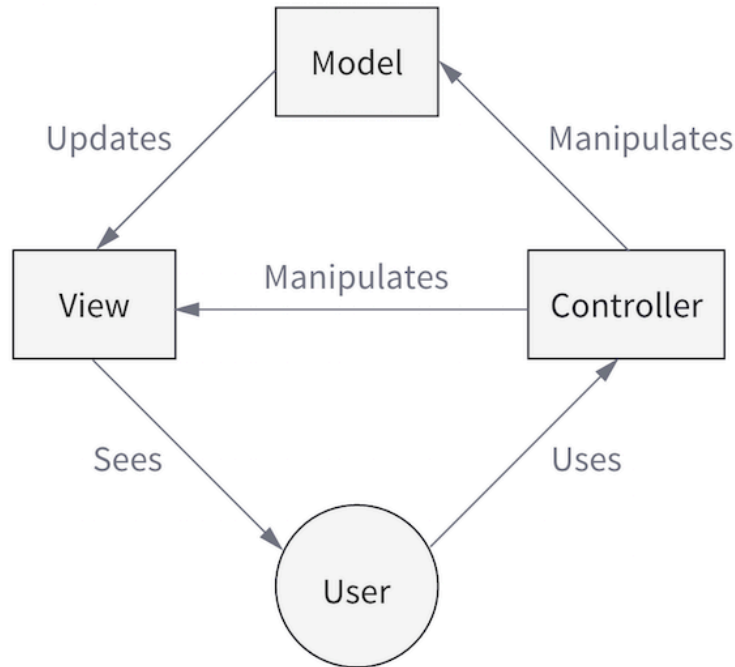
Figure 2.3.1.1 working principle of MVC

Figure 2.3.1.1 shows the working principle of MVC. The advantages of the architecture designed by the MVC model are low coupling and relatively high code reusability. It can group a project and develop it in modules, which can effectively reduce development costs and improve the maintainability and engineering of the software. management. In the Android development process, the MVC framework can also be used to divide the application development into multiple modules, the code reuse rate is high, and the various modules also have the characteristics of low coupling. The corresponding relationship among the Model View Controller in Android development is shown in Figure 2.3.1.2.
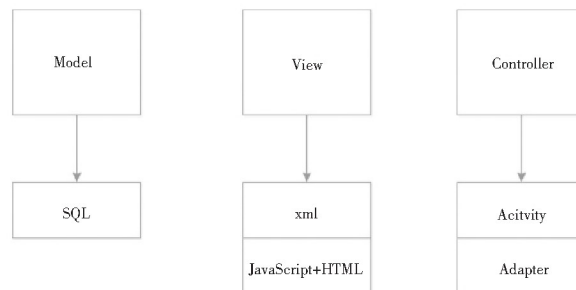

Figure 2.3.1.2 three parts of MVC

It can be seen from Figure 2.3.1.2 that the three parts of MVC have their own corresponding methods to implement. The Model layer provides business support for the

entire Android application and is mainly responsible for database requests, including some SQL operation statements and some Internet network operations.

The view layer (View) generally uses XML Layout in Android development. The author uses Android Studio development tools. It provides a simple XML page layout method. It is very convenient to reference the elements in the back-end development process, and it is visualized. The editing method is very conducive to the improvement of the later pages. When writing, it is necessary to determine whether it is consistent with the ID of the interface, which greatly enhances the maintainability of the code.

The control layer (Controller) is responsible by the Activity. Activity is one of the most basic components in the Android component. The Activity is responsible for coordinating the work of the Model and the View. After receiving the operation from the user, it is passed to the Model layer. After the business operation of the Model layer The result is returned and the View is updated and displayed to the user.

The use of this model of framework in Android development makes the entire system have a clear division of labor and cooperation, and makes code development more standardized and formatted, which provides great convenience for later maintenance, because the Model layer is responsible for the business of this Android application Logical processing, and the View layer is only responsible for the view interface, so the two can be said to be independent of each other and do not interfere with each other, which is convenient for adding new functions later.

2.3.2 NodeJs

NodeJs was born in 2009, it is based on GoogleChromeV8 engine. Javascript runtime is mainly used to create fast and scalable web applications. NodeJs adopts event-driven and non-classical I/O model, has the characteristics of lightweight and high efficiency, and is especially suitable for real-time application systems in data-intensive application environments.
As the operating platform of server-side Javascript, NodeJs retains the characteristics of front-end Javascript. Compared with other server-side languages, NodeJs has the following characteristics:
1. asynchronous I/O
Most operations in NodeJs are called asynchronously. The significance of this is that parallel I/O operations can be performed at the language level. There is no need to wait for the end of the previous I/O call between each call, and the efficiency has been greatly improved
2. event-driven
In the Web2.0 era, Javascript takes on more responsibilities on the front end. NodeJs introduces the widely used event mechanism in the front-end browser to the server, and cooperates with asynchronous I/O to expose event points to business logic. Event-driven programming has the advantages of being lightweight, loosely coupled, and focusing only on transaction points

3. single thread
NodeJs maintains the single-threaded feature of Javascript in the front-end browser.
Compared with multi-threaded programming, NodeJs does not need to consider the
performance of state synchronization, deadlock and context switching in multi-threading
everywhere.
problem.
Because of the event-driven and asynchronous non-blocking I/O characteristics of
NodeJs, NodeJs is particularly suitable for network-oriented parallel I/O processing.

2.3.3 MongoDB

MongoDB is a scalable, high-performance and high-availability document-oriented-open-
source database that can be applied to enterprises, industries, and various applications.
MongoDB can make developing groups more agile and expandable. By using MongoDB
to create new applications, developing groups can achieve the goal of improving work
efficiency, speeding up the time, and reducing their costs. MongoDB has collection-
oriented storage, suitable for storing objects and data in the form of JSON, supports
dynamic query, freedom of mode, complete index support, support for replication and
automatic failure recovery, automatic fragmentation processing, automatic shading,
efficient binary data storage and other functional features Among them, shading
technology, replication and automatic failure recovery are the key technologies for
building high-availability distributed database clusters.

2.3.4 OkHttp3 + Retrofit

OkHttp is an open-source network request library of Square. Using OkHttp3 for network
requests can increase the upload speed of HTTP requests and save bandwidth at the same
time. The efficiency of OkHttp3 is reflected in the following three aspects: (1) Allow all
requests connecting to the same host to share the same socket; (2) Use responsive caching
to avoid repeated requests; (3) When there are multiple IP addresses on the server, If the
connection to the first address fails, OkHttp3 will silently recover by trying an alternate
IP address.
Retrofit2 is the same as OkHttp. Retrofit is also from Square. Retrofit2 can be understood
as an HTTP network request adapter. It expresses an HTTP request through a Java/Kotlin
interface dynamic proxy, and sends HTTP requests through OkHttp3. The relationship
between Retrofit and OkHttp can be summarized as follows: OkHttp is purely an
HTTP/SPDY client; Retrofit is a high-level REST abstraction based on HTTP. The
advantages of the Retrofit framework are: (1) Use clear annotations to simplify the
spelling of URLs to the greatest extent; (2) Large degrees of freedom, support for custom
Converters and other business logic; (3) Support both synchronous execution and
asynchronous Execution; (4) Support a variety of file parsing, such as GSON, JSON and
XML, etc.;

Retrofit2 is a RESTful network request framework based on OkHttp encapsulation. It is currently the lowest network and the most powerful network request framework.

2.3.5 WebSocket+SocketIO

WebSocket makes instant communication more effective. Users can use polling over HTTP, sometimes even streaming, and receive notifications via HTTP. But WebSocket can effectively save network resources, CPU usage and shorten network latency. WebSocket has a significant improvement in performance.
WebSocket makes the communication between client and server on the Web easier. Developers who have used the architecture before WebSocket to build instant messaging applications will know that the instant messaging technology of the HTTP protocol is very complicated. And maintaining communication in stateless requests adds to the complexity. Every attempt to extend HTTP to use cases where it does not apply increases the complexity of the software. The WebSocket communication protocol significantly simplifies connection-oriented communication in instant applications.
SocketIO is a JavaScript library for instant web applications. It makes real-time two-way communication possible between the server and the client. It has two parts: a client-side library that runs in the browser, and a server-side library for NodeJS. Both have almost the same API. Like NodeJS, it is also event-driven.

2.3.6 Osmdroid

Open Street Map (OSM) is an online map collaboration project, OSMDroid is its terminal map applied to the Android operating system. By visiting the website, you can freely draw a map, mark the desired location on the map, add buildings, etc. Based on OSM, users can build their own map server, which can be used to embed in a website or provide external services, such as for mobile devices. OSMDroid maps provide the Marker method to use custom vector graphics as map symbols. Therefore, in the process of customizing map symbols, the mapping relationship between vector graphics primitives and sensor data is established through the relationship between sensor data and symbol visual variables to realize the sensor of custom map symbols. data visualization.

2.3.7 Glide

Pictures are an important part of Android development. They are not only used to set user avatars, icons and other functions, but also display content intuitively. In Android development, the storage and display of pictures consumes a lot of memory. At the same time, the transmission of pictures in network data requests also occupies a lot of resources.

Glide is an efficient and open-source media management framework that supports functions such as loading and decoding pictures and videos, and can be used in almost any network protocol stack

The main features of Glide image loading framework are:

1. Supports multi-threaded asynchronous download of pictures, and supports network communication boxes such as OkHttp, Retrofit2.0, etc.

Frame, suitable for environments with poor network connection conditions, fast image loading speed.

2. The image loading function is rich. Use the drawables object as a placeholder to reset the picture

The width and height. Support loading thumbnails to reduce the time to load multiple pictures in the same View at the same time.

3. Integrated Activity life cycle Glide can intelligently pause and restart according to the status of Activity,

And after the Android device reconnects to the network, the failed request can be automatically restarted.

3. User manual

3.1 Main Feature

3.1.1 Register and Login page

In this page, we offer secure log-in function: if you are a new user, touch register to register your fundamental personal details by e-mail and password. If you have your account, just fill up the e-mail and password blank and log in.
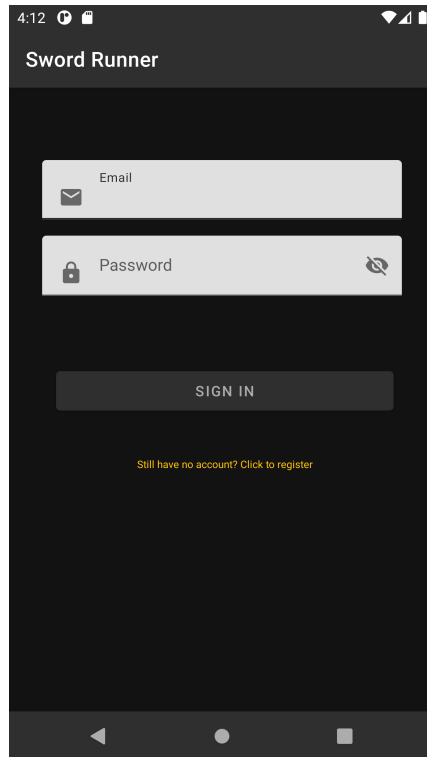
Figure 3.1.1 Log in Page
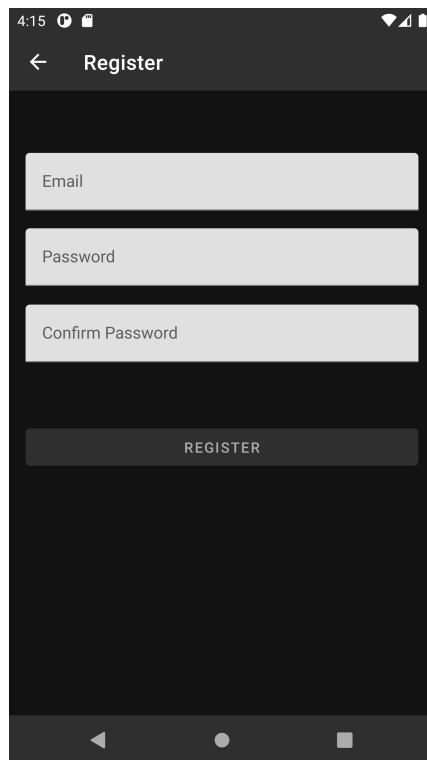
Figure 3.1.2 Register page

3.1.2 Home page

In this page, you can see your surviving days, which means days mission finished. Under that you can see the mission you have, including single-player mission and multi-player mode. Moreover, the schedule of the games will be placed too.
Under the dashboard it's a "start to run" button. Press the button to switch to the game mode.
Under the button there is a high score blank which will show best players among friends. Another important button is the plus button on the top-right of the screen. This button can let you initiate a single mode or multiple mode game. You can't start running when you are not in a game.
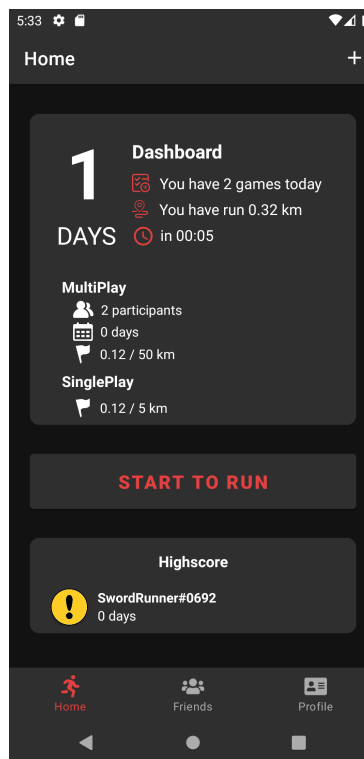
Figure 3.1.3 Home page
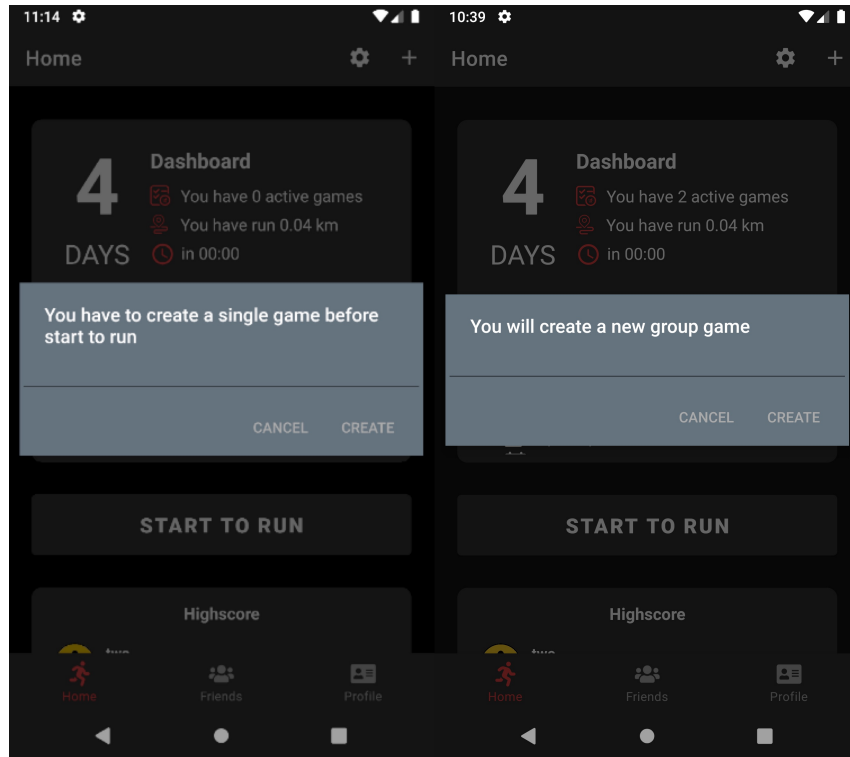
Figure 3.1.4 Can't start to run without a game & Initiate a game (right)

3.2 Game Feature

3.2.1 Single-Player Mode

In this page, you click the start botton to start running. The scale in the left side shows your schedule of play. The route of running will be shown in the screen. After touch stop button, your running distance and coins will be calculated. When you go back to home page, the data will be renewed. After you finished the goal of that day, the surviving days will add one. If when it comes to 24:00 of the day and the goal is not reached, it means you are eaten by zombies thus the surviving days will set to zero.
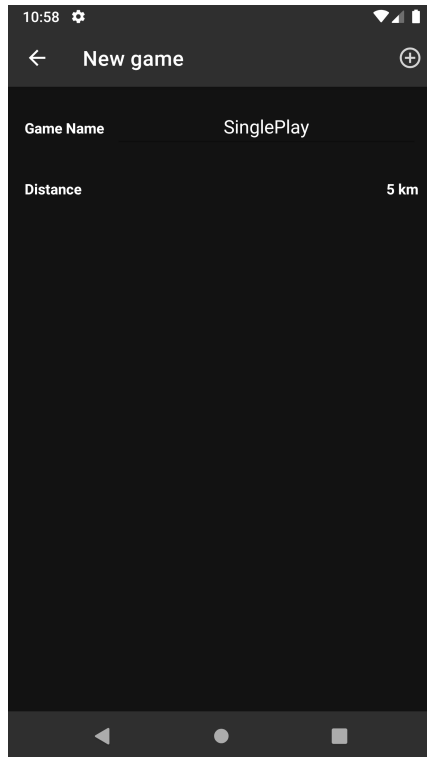
Figure 3.2.1.1 Initiate a Single-Player game
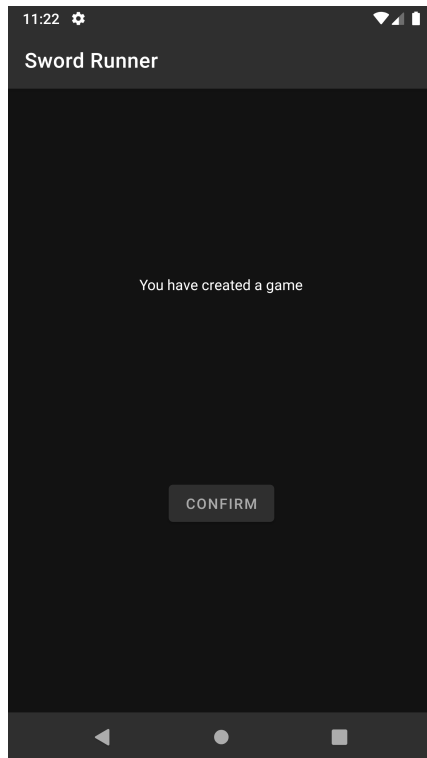
Figure 3.2.1.2 Initiate a Single-Player game success
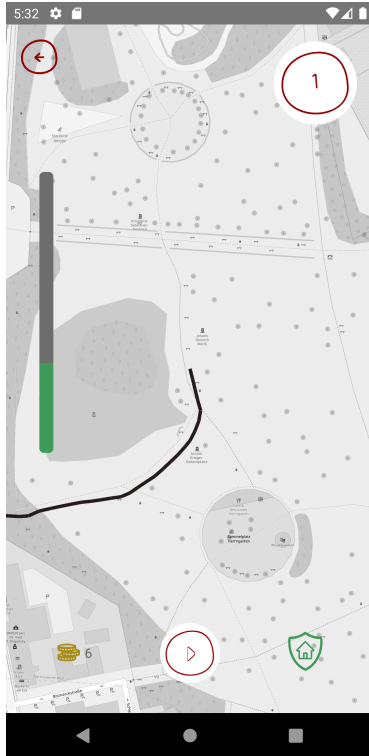
Figure 3.2.1.3 Single-Player Mode
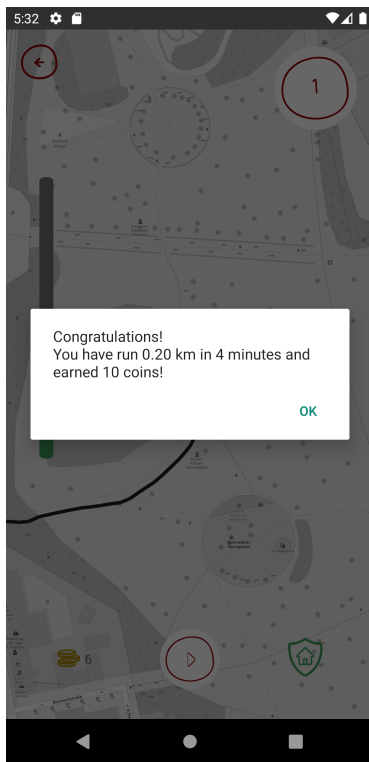
Figure 3.2.1.4 Calculation Page
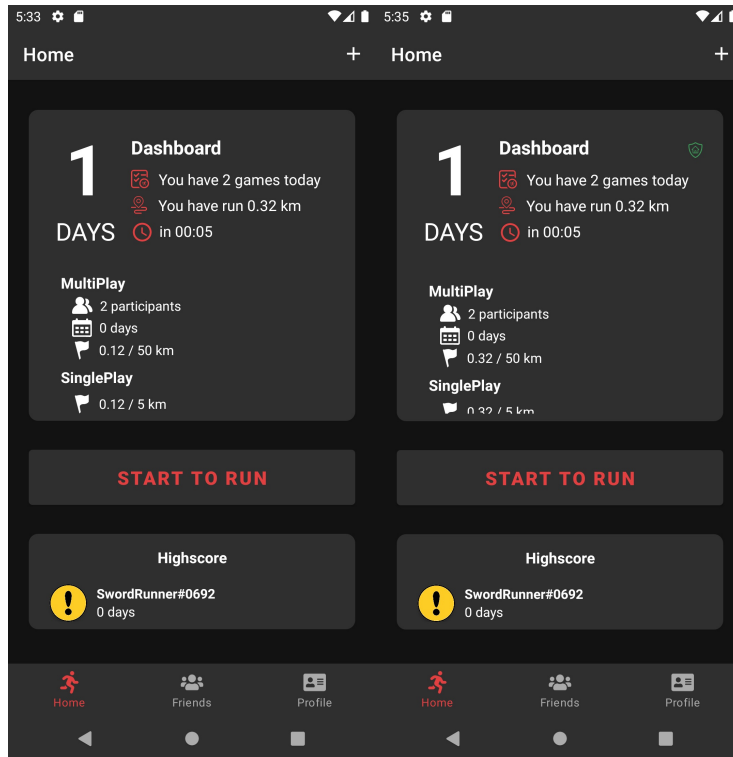
Figure 3.2.1.5 Home page before and after play

3.2.2 Multiple-Player Mode

The multiple player mode is like several player make effort achieving the same goal of the game. You can initiate the name, distance goal, participants of the game. Then you can play the multiple game.

Concerned to the ability of human being, the game mode is same to single player game. To achieve the same goal, players can be more motivated in multiple player mode. The data shown on the dashboard is updated in real time.

If the participants finish a run, the status will be synchronized to all members of the game. We believe in this way people will be motivated in running.

If one of the players delete the game, the whole game will be destroyed. And the information will be synchronized to all members of the game.
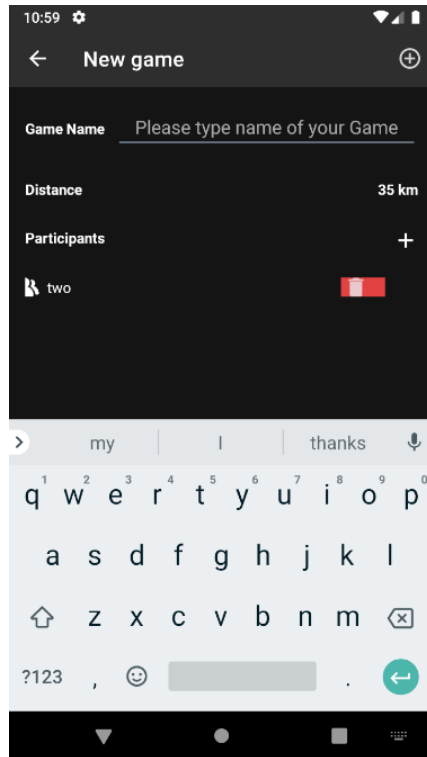
Figure 3.2.2.1 Initiate a multiple player game
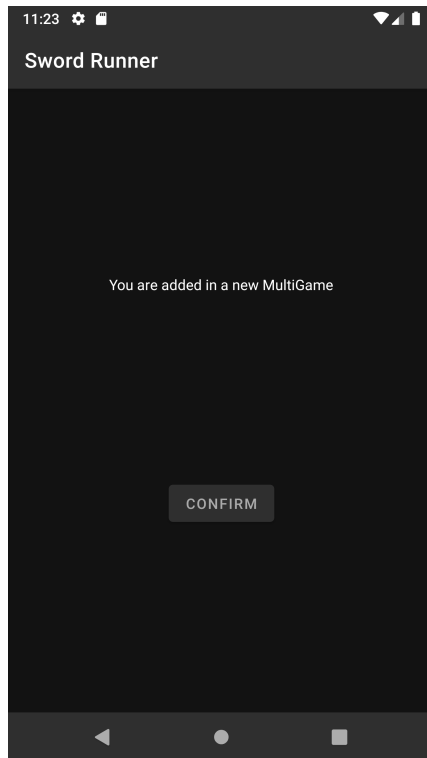
Figure 3.2.2.2 Participants join in a multiple player game
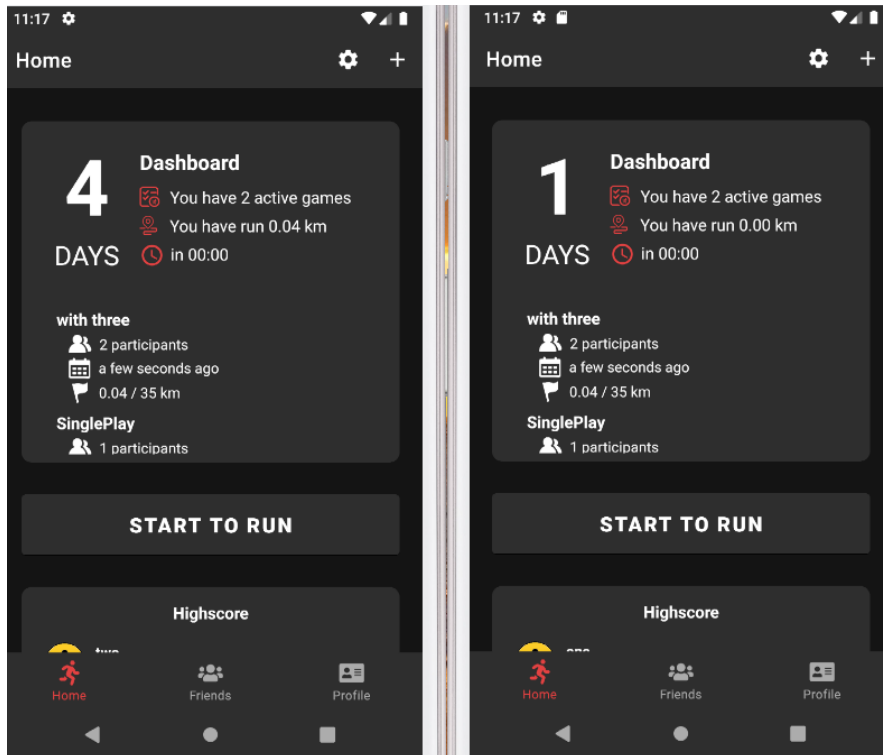
Figure 3.2.2.3 One multiple game shown in both side of players
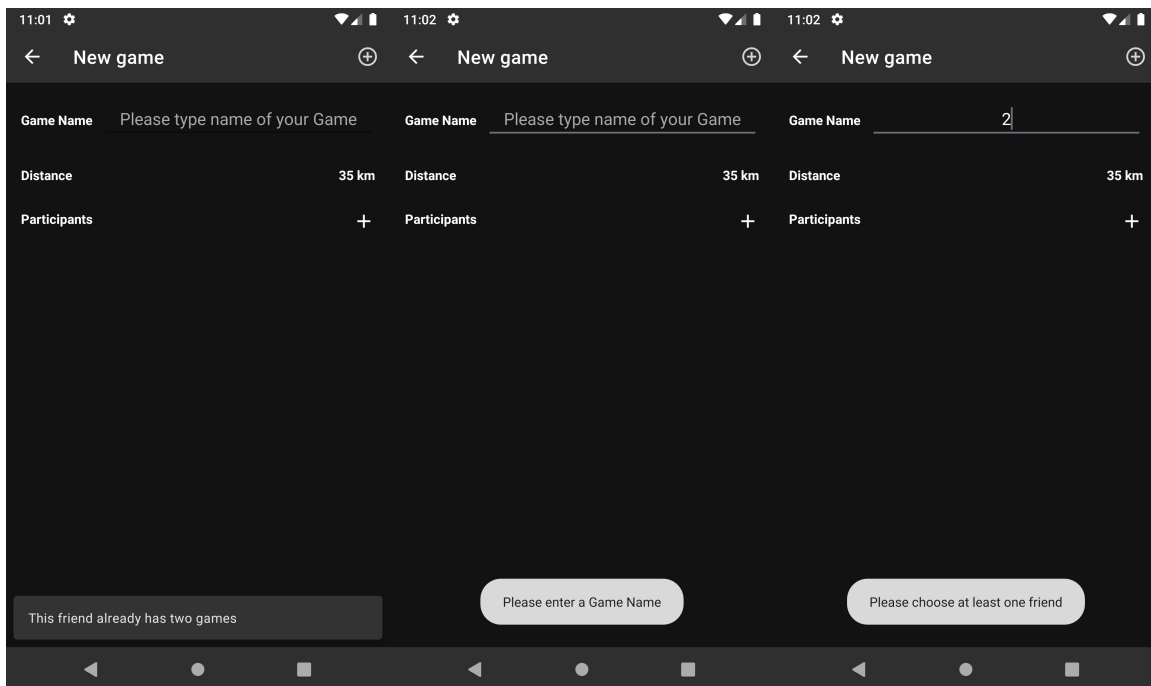
Figure 3.2.2.4 Game logic: (left) can't add same friend in two different games; (middle) must have a game name; (right) must have at least one friend
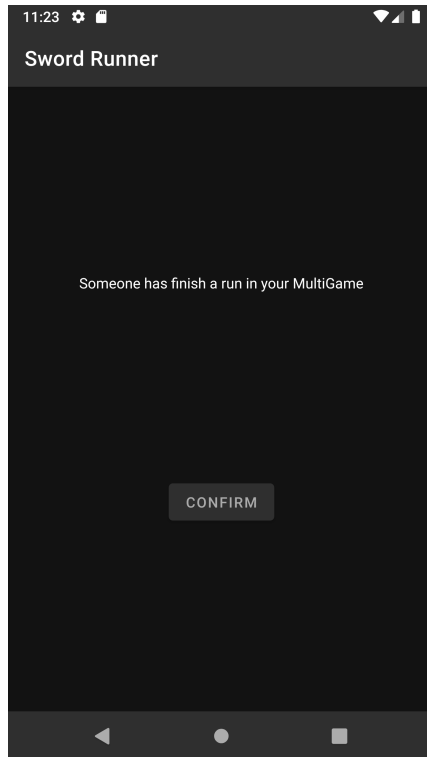
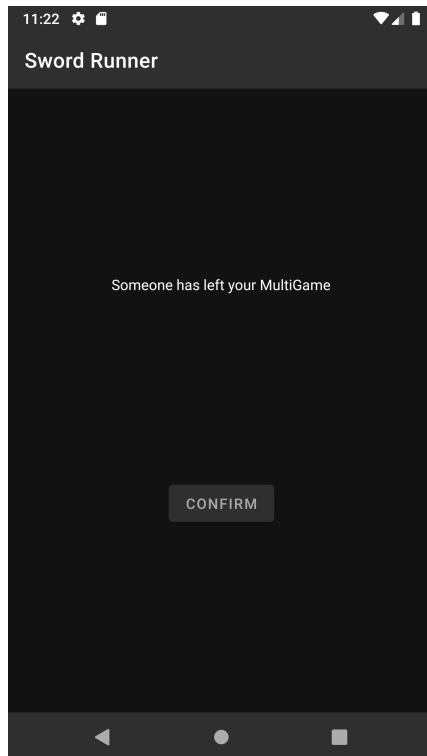Figure 3.2.2.5 Notification of participant's status (finish a run)

Figure 3.2.2.6 participant delete a game

3.2.3 Game mode revisit

1. Survival judgement
The server will calculate the game status at 24:00. If the requirement is not meet, the user will fail the game thus survival days return to 0.

2. Running status judgement
Speed larger than 25 km/h won't be judged as running. Once the speed is lower than 25 km/h, it will start recording.

3. Some other details
The running record will be count to every game initiated of that day. For example, if the user ran 3km in the morning and initiate another game in the afternoon, these 3 km will still count to this game. The user doesn't need to run more due to the time of initiation.

3.3 Friend Feature

3.3.1 Friend Function

In the top-right corner you can see a plus icon, using which you can add friends. In the main middle you can see chatting area. All friends' chatting status will be placed here.
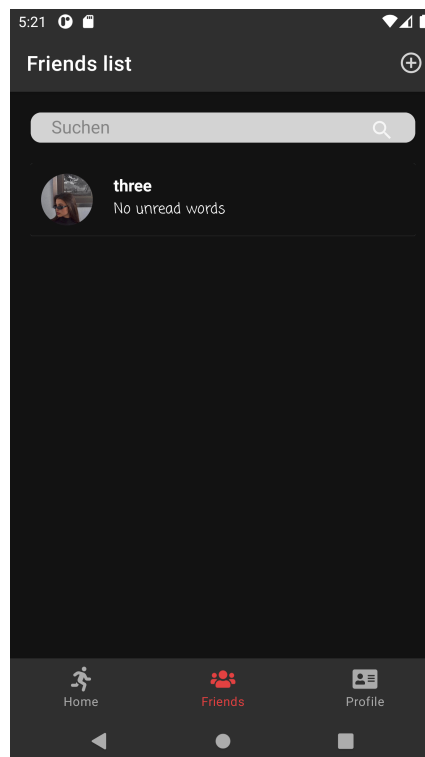
Figure 3.3.1 Friend List (only one friend now)

3.3.2 Chat Function

Tap the text blank you can chat with your friend. The message is synchronized and the message will be saved in server. Moreover, the chatting message will trigger a notification of receiver side's phone.
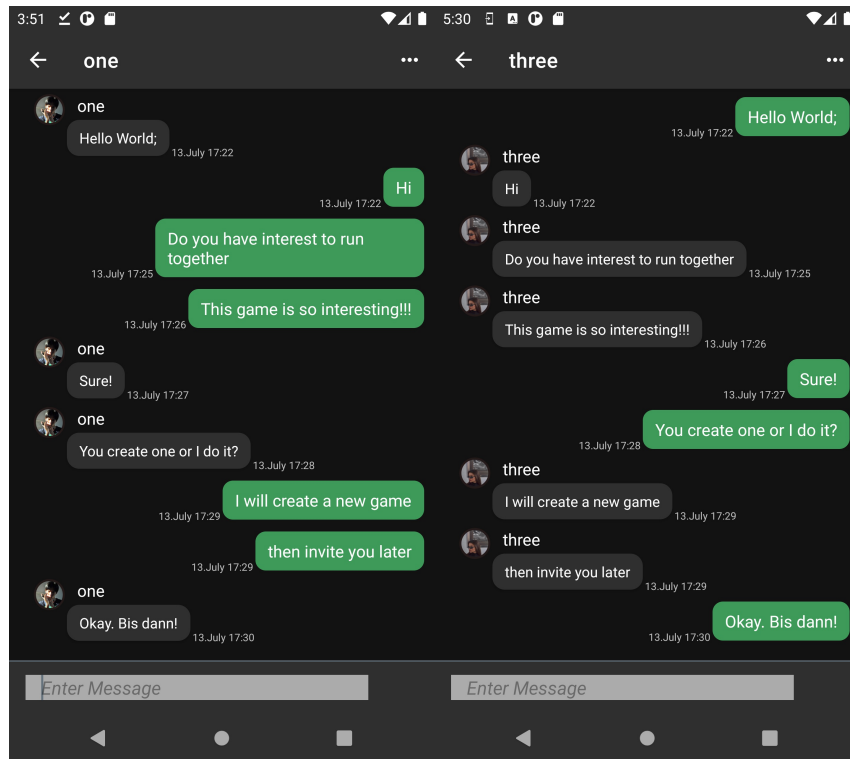Tap ellipsis in the top-right corner to check his/her personal details.

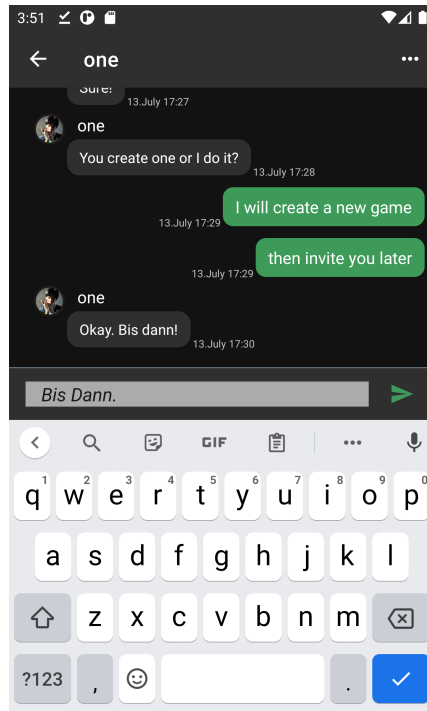Figure 3.3.2.1 Chatting Function (both sides compared)
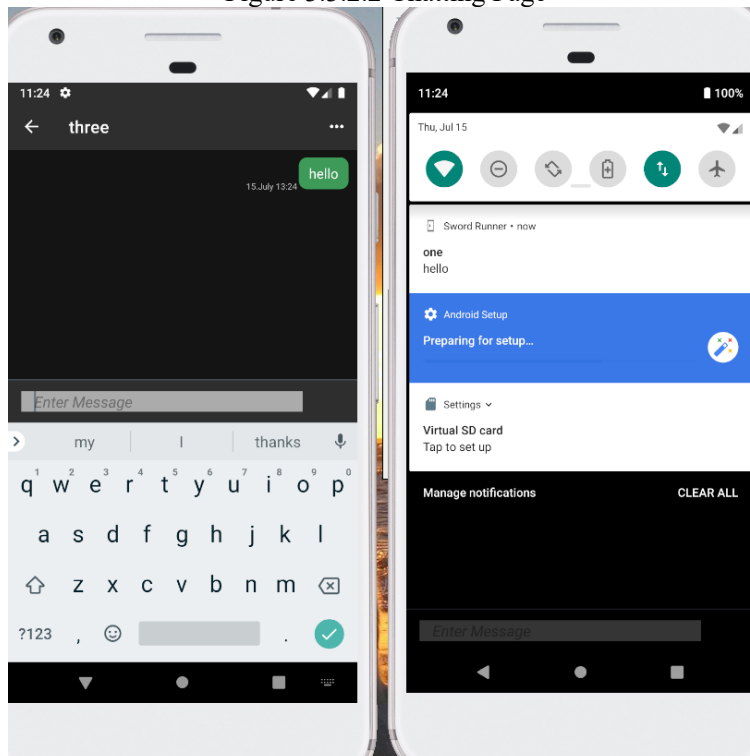
Figure 3.3.2.2 Chatting Page

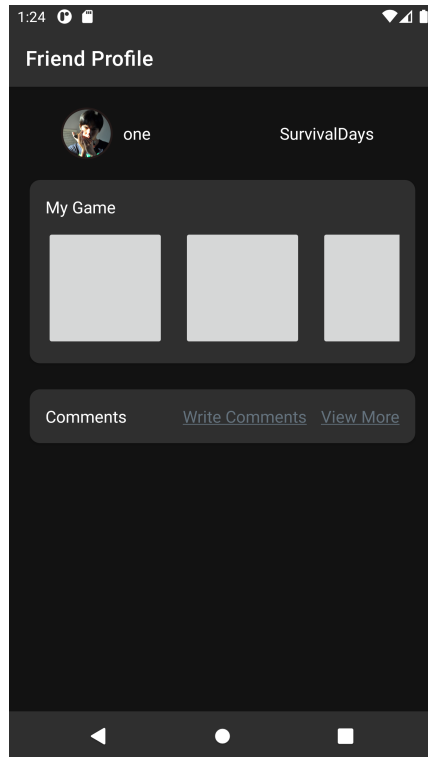Figure 3.3.2.3 Chatting Notification

Figure 3.3.2.4 Check friend's profile by tapping the ellipsis

3.3.3 Adding Function

Search a friend by his/her name. The request will be shown in the friend's side. You can chat with your friend if your request is accepted.
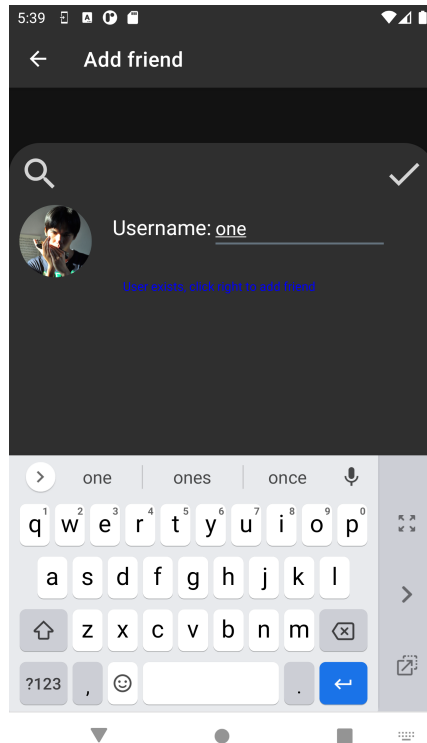
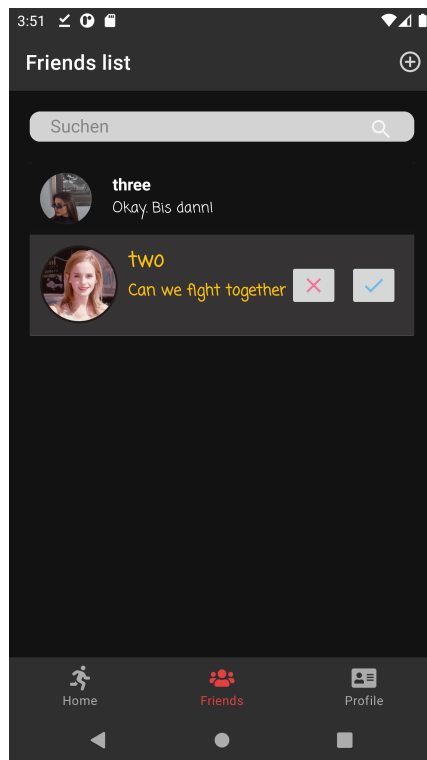Figure 3.3.3.1 Friend Adding Page

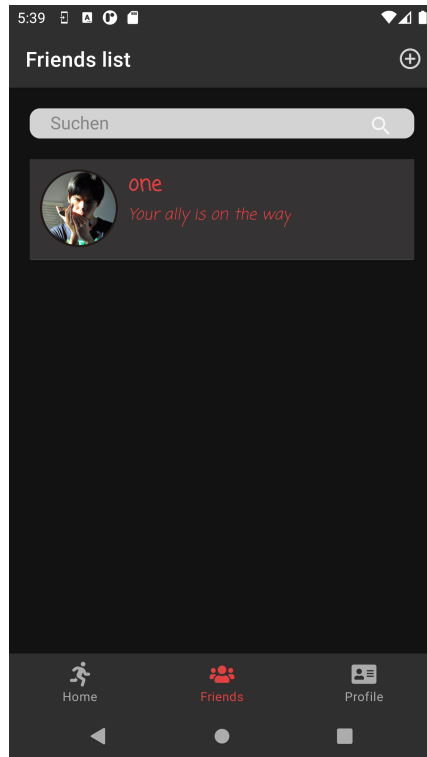Figure 3.3.3.2 Friend's side to decide accept or not

Figure 3.3.3.3 User's side for waiting for accept
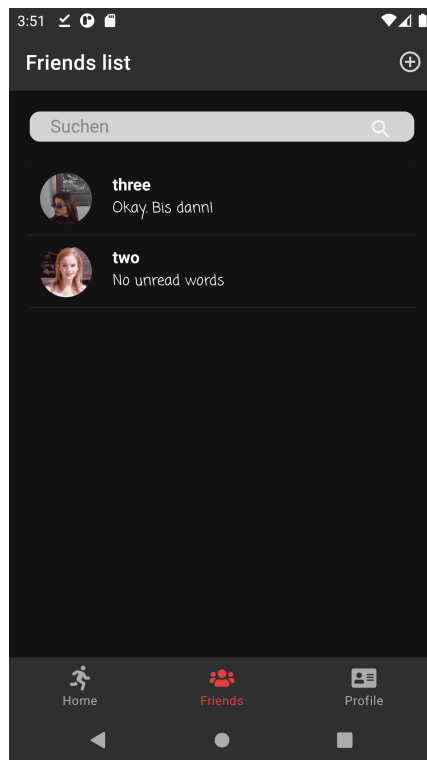

Figure 3.3.3.4 Friend Added Successfully

3.3.4 Delete Function

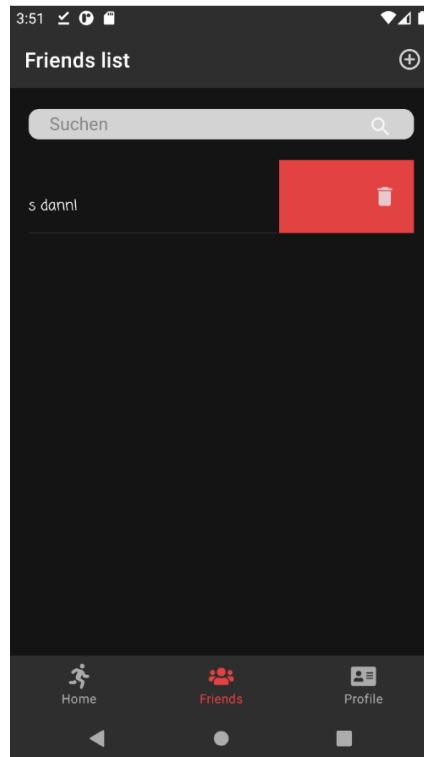Swipe right the chatting blank to delete the friend you want to delete.

Figure 3.3.4.1 Friend delete Function

3.4 Profile Feature

3.4.1 Profile main page

In this page you can check your personal details: your avatar, your username, comment and game list. You can change your personal details in the right-up corner of the page.
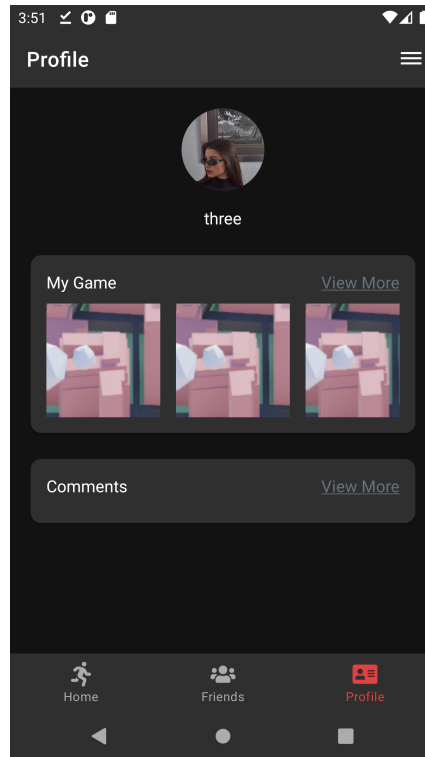
Figure 3.4.1 Profile Page

3.4.2 Game List page

In this page, you can check your game list. In the Figure 3.4.2, the up one which with one man means single-player mode game. The downer one with more people means multiple-player mode.
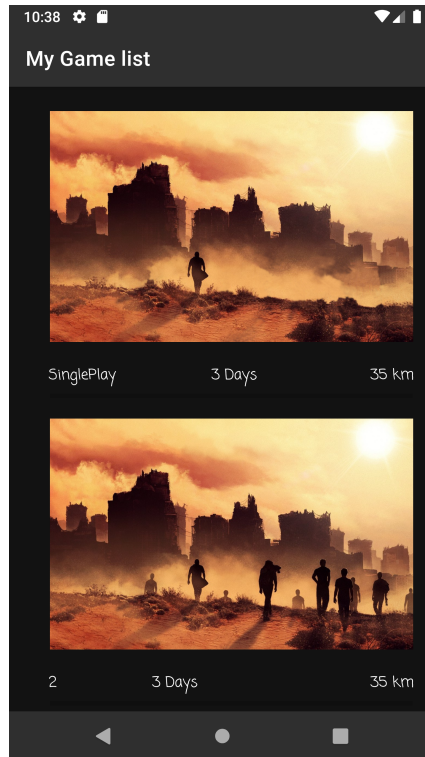
Figure 3.4.2 Game List Page (upper is single-player mode and lower is multiple-player mode)

3.4.3 Comment System

We designed a comment system in order to let player know each other well when playing multiple-player mode, which is similar to rate & comment system of ebay to some extent. You can see your friend's average stars and appraise condition. We believe in this way players can learn more about each other before cooperating in a multiple-player game thus improve their user experience.
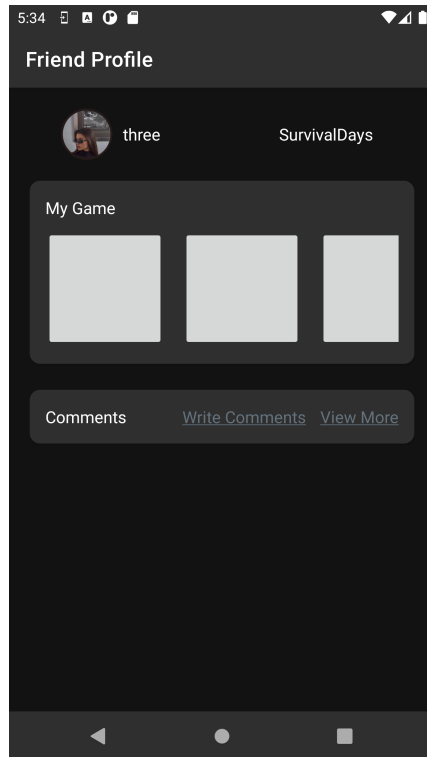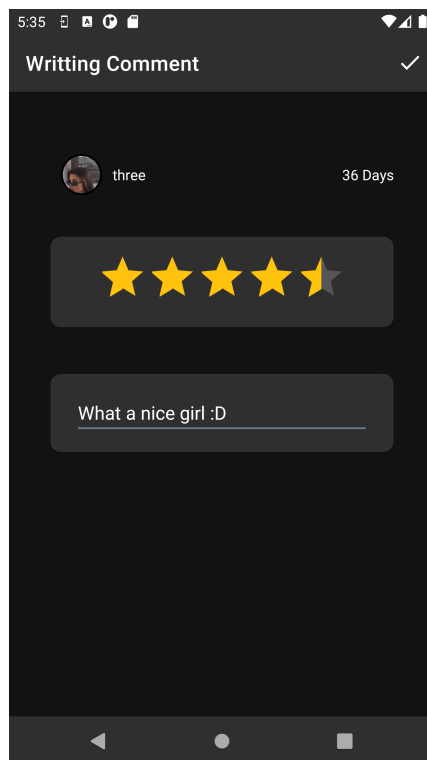
Figure 3.4.3.1 Friend's profile Page

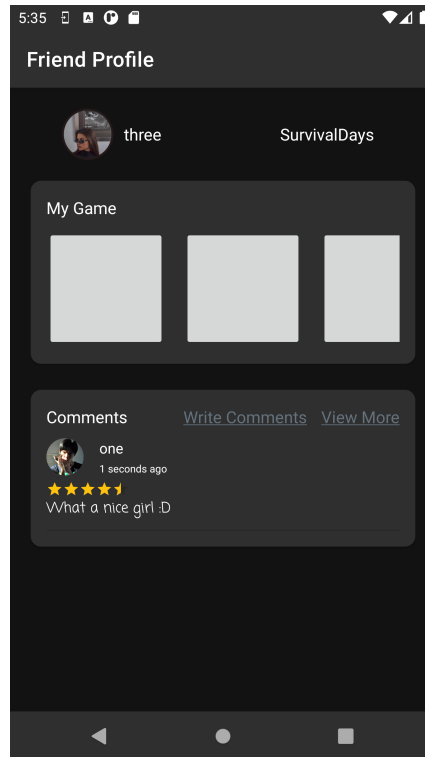Figure 3.4.3.2 Give stars and comment

Figure 3.4.3.3 Stars and comment will be shown in the page

3.5 UI Design

We are committed to making the interface simple, easy to understand and user-friendly.
1. Color matching: We make our software more attractive through the five elements of the smooth design system: Light, Depth, Material, Motion, and Scale.
2. Icons: We reduce redundant decorations, use symbolic icons, reduce buttons and options, so that users can more intuitively understand the required information, reduce cognitive barriers and learning costs
3. Operation logic: clear logic, no bugs. We make effort to let it easy for users to use.
4. Multi-interaction: Make it easier for users to use our software through simple and attractive interactive mode, thereby reducing time cost.
5. Design specification: Although we are a distributed development, we have a very unified UI design. And we chose the color scheme of doomsday survival related to the theme of the game to make the game look more natural and interesting.

4. Bullet point feature list

- Secure Login
  - Register
  - Login
- Profile Function

- Show user's details: avatar, username,
- Change name, avatar, e-mail
- Check game list, comments
- Delete account
- Game mode
  - Single-player mode
    - Finish game by running
    - Route will be shown on the map
    - Earn coins by running
    - Using coins to skip
  - Multiple-player mode
    - Invite friends to join a multiple-player game
    - The multiple-player game will be deleted if one participant quit
- Friend Function
  - Add/delete friend
  - Chat in real time
  - Check friend's profile
- Comment System
  - Give stars to friend
  - Comment to friend
- UI design
  - User friendly color matching
  - Easy to understand icons
  - Easy user logic

5. Conclusion and outlook

5.1 Conclusion

Inspired by serious lock-down situation and people are in lack of exercise, we designed a game with a doomsday survival theme story to motivate people come out and run. In this project, we six members as a team designed and realized an android based game which use osmdroid, nodejs, mongodb. It is a front-end separation project. We finished all mandatory features like secure login, profile system, friend system, single play and multi-play system. Furthermore, we designed a comment system to let users appraise each other. We also conceived and implemented a coin system which can let users skip if they have difficulty in finishing task of a specific day.

5.2 Outlook

In general, the game meets our requirement. But there're still many places can be improved.

Firstly, we should improve playability of the game, so more interactions like CG, story introducing anime, instruction tutorial of the game should be added in future efforts. These elements can make the game look more fantastic thus the players are more would like to play.

Secondly, we should design some functions to keep players. For example, a shop system and a collection system can be introduced in future efforts. Also, more game modes can be implemented. Ideas for further more competitive game modes: A Last-Man-Standing mode were the players compete on the same settings which player keeps his village alive the longest. In addition, the same mode just with teams that compete with one another. Also, a sabotage mode where its possible to buy items to make it harder for the other team or players. The possibility to create running tracks and compete for best times, is one more possible extension of the game. Moreover, it would be interesting to include a setting option to increase the daily running distance over time.

Thirdly, if there are too much users play at the same time, the server we use may shut down. We concern to use better use if user more than the load of our server. Some programming strategies can be used to avoid high concurrency.

To finish, I want to thank all the members of IPTK lecturers to give us a chance to do this project. All of us learnt a lot in this project. Besides the technology we used in the project, we also learnt how to cooperate as a team and experienced scrum software engineering.