



# **Requirements Specification**

**Project Name: Char Char**

**Team Number 10**

[Binary Team/ Team 10] **Specification Document** [22.03.23]

## Document Information

**Project Name:** Char Char  
**Prepared By:** Team 10 **Preparation Date:** 22/03/2023  
**Email / Phone:** charchar20230512@gmail.com  
**Document Version No:** 1.0 **Document Version Date:** 11/05/23

## Version History

Ver. No.	Ver. Date	Revised By	Description
0.1	22/03/2023	Team 10	<p>Initial revision, main compilation and all sections not listed below</p> <p>Purpose – Jia He Li</p> <p>Section 1 – Jia He Li</p> <p>Section 2 – Meihui Li / Jia He Li</p> <p>Section 3 – Meihui Li / Yi Ting Liang</p> <p>Section 4 – Yi Ting Liang</p> <p>Section 5 – Meihui Li / Yi Ting Liang</p> <p>Section 6 – Meihui Li / Yi Ting Liang</p> <p>Section 7.1 – Yi Ting Liang</p> <p>Section 7.2 – Jia He Li / Yi Ting Liang</p> <p>Section 7.3 – Meihui Li/ Jia He Li</p> <p>Test plan – Yi Ting Liang</p>
1.0	11/05/23	Team 10	<p>General update to feedback provided by Dr. John Colquhoun and Dr. Dan Nesbitt</p> <ul style="list-style-type: none"> <li>- Version history updated/completed.</li> <li>- Purpose updated/completed.</li> <li>- Analysis updated/completed.</li> <li>- Hardware and software updated/completed.</li> <li>- Functional and non-functional requirements have been numbered and updated.</li> <li>- Modification in light of comments made or change deemed necessary added.</li> <li>- Assumptions and constraints updated</li> </ul>

[Binary Team/ Team 10]	<b>Specification Document</b>	[22.03.23]
------------------------	-------------------------------	------------

## Table of Contents

- 1 Background & Analysis 3
  - 1.1 Analysis Process 3
  - 1.2 Analysis 4
  - 1.3 Scope 4
- 2 Hardware and Software 5
  - 2.1 Software 5
  - 2.2 Hardware 5
- 3 References 6
- 4 Definition of terms 7
- 5 Solution Requirements 8
  - 5.1 Functional Requirements 8
  - 5.2 Non-functional Requirements 9
- 6 Other considerations 10
  - a. Assumptions 10
  - b. Constraints and Dependencies 10
- 7 Modification in light of comments made or change deemed necessary 11
  - 7.1 UML final version 11
  - 7.2 GUI final version 15
- 8. Initial requirements analysis of user interaction 18
  - 8.1 UML diagrams 18
  - 8.2 Initial ideas for a GUI 19
  - 8.3 User cases 25
    - 8.3.1 User case diagram 27

## Purpose (Executive summary)

The following documents will demonstrate the development processes as well as the specification for a tourist information application. It shows the how we as a group formulate the idea of building an application and how our design goals changes as time progress. The application we are asked to build is a tourist information software specifically for tourist to find near location of interest. The primary function is guiding the user towards the target through turn-by-turn instruction. We will also need to provide utilities like displaying information of tourist attractions and leaving written and star reviews. Our system is aimed to provide a much simpler user interface with more clear interaction and less cluttered layout.

---

## The Remainder of the document must contain:

### 1. Background & Analysis :

#### 1.1. Analysis Process

Our analysis process was greatly influenced by the map functionality and point-of-interest display approach of Google Maps. We were inspired by how Google Maps seamlessly integrates mapping capabilities and provides detailed information about various locations. This served as a valuable reference point for developing our own tourist information application.

The user interface of the app was inspired by Airbnb. As avid Airbnb customers, we found the frontend design to be straightforward, user-friendly, and aesthetically beautiful. Airbnb's overall colour style and

[Binary Team/ Team 10]	<b>Specification Document</b>	[22.03.23]
------------------------	-------------------------------	------------

intuitive layout provide a clear and straightforward user experience. We aimed to apply comparable design concepts to ensure that our application gives users with a visually beautiful and simple-to-use interface.

Furthermore, we drew inspiration from the way Lonely Planet showcases attractions, providing comprehensive and relevant information to travelers. By referencing Lonely Planet's presentation style and information structure, we aimed to create a robust and informative display of tourist attractions within our application.

By combining the mapping functionality and point-of-interest display principles from Google Maps, the user-friendly interface of Airbnb, and the informative approach of Lonely Planet, we seek to create a navigation application that offers a seamless and enjoyable user experience while providing accurate and valuable information to users exploring various tourist destinations.

## 1.2. Analysis

In developing our tourist information application, it is crucial to consider the broader tourism industry and the specific needs of stakeholders within this domain.

Tourists themselves are the primary stakeholders within the tourism industry. They heavily rely on information and services to plan their trips, explore attractions, and navigate unfamiliar locations. Therefore, our application is designed to provide them with a user-friendly platform for easily locating points of interest and accessing detailed information about attractions.

Additionally, local businesses and attractions can also be considered stakeholders as they stand to benefit from increased visibility and promotion through our application. By featuring their establishments within our platform, we aim to create a mutually beneficial relationship between tourists and local businesses.

To comprehensively analyze the business environment, we will examine existing systems and processes within the tourism industry. This involves reviewing other tourist information applications, travel guidebooks, online travel platforms, and related technologies. By understanding these existing systems, we can identify potential gaps, challenges, and opportunities to deliver a unique and valuable solution.

Through conducting this analysis, our objective is to develop a comprehensive understanding of the business environment, stakeholders, and existing systems within the tourism industry. Armed with this knowledge, we can proceed to develop a robust and user-centric tourist information application that effectively addresses the needs and preferences of all relevant stakeholders involved.

## 1.3. Scope

Document here what your team regards as "in scope", but also what is regarded as "outside of scope" – in other words, what functions are you explicitly excluding and including?

- A web based android application which primarily focus on walking guidance of a tourist.
- Calculate the best route (in terms of travel time) to the destination.
- Various ways for the user to search his/her place of interests, including search bar, auto search with 1 mile.
- Display a list of attractions based on different filters, and allow users sort the list base on his preference.
- Build a data base for registered user, as well as the function of signing in a new user.

- Build a data base for registered attractions.
  - Allow the user to leave feedback for both the attractions and our software.
  - Intuitive interaction with maps, including pinch to zoom, press.
- Out of Scope
- The guidance is for walking only, there is not need for considering driving or other forms of public transportation.
  - After the user leave a review for the attractions, there is not need for the attraction to reply to the comment.
  - Designed for android device only, no need to adapt it for web browser or iOS device.
  - No need to consider the options for users to upload photos for a specific attraction.
  - No need to consider the options for users to purchase a ticket of an attractions.

## 2. Hardware and software platforms to be used for developing and running your solution

**solution -** This project is expected to a web application. This application will be developed on student and departmental computers and is expected to be connected to a central MySQL database hosted by the department.

### - Software platforms

Frontend:

HTML, CSS,

TypeScript: version 4.5.5

Vue.js version 3.2.13

vue-router version 4.0.3

Moment version 2.29.4

Lib-flexible :0.3.2

element-plus version 2.2.31

Backend:

Jdk version 1.8

Mysql version 8.0

Mybatis version 2.2.0

Springboot version 2.7.10

Redis version 7.0.11

Test:

Postman

Jdk version 1.8

Jmeter

### - Hardware platforms

Frontend:

1.Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz

RAM 8.00 GB (7.88 GB Available)

2.CPU:Apple M1

memory: 8GB

macOS 13.3

## Backend:

1.CPU AMD Ryzen 9 6900HX with Radeon Graphics 3.30 GHz  
RAM 16.0 GB (15.3 GB Available)

2.Device name DESKTOP-KMES7MR  
Processor Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz  
Installed RAM 8.00 GB (7.88 GB usable)

System type 64-bit operating system, x64-based processor

3.MacBook Pro  
CPU: Apple M1 Pro  
memory: 16GB  
macOS 13.2.1(22D68)

## Testing:

1. Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz RAM 16.0 GB (15.8 GB Available)

2.Chip:Apple M1  
Memory:8G  
macOS:12.4

**3. References –****-Frontend**

- <https://colorhunt.co/>(2023.03.10)
- Vue.js Course for Beginners: [https://youtu.be/FXploQ\\_rT\\_c](https://youtu.be/FXploQ_rT_c) (2023.03.12)
- Ui - Top 10 Travel Apps UI/UX Design Case Studies:  
<https://medium.muz.li/top-10-travel-app-ui-design-case-study-efe41dcf7194> (2023.03.13)  
<https://www.youtube.com/watch?v=uqpM7WVTKl4&t=465s>(2023.04.01)  
<https://www.youtube.com/watch?v=qieXC2pfeN0>(2023.04.03)  
<https://chu1204505056.gitee.io/#/en-US>(2023.04.15)  
[https://youtu.be/l\\_xLMnNeLDY](https://youtu.be/l_xLMnNeLDY)(2023.04.15)  
<https://cn.vuejs.org/guide/introduction.html>(2023.04.16)

**-Backend**

## Learn to Build a Spring Boot Web App:

- <https://youtu.be/xaR7Nk0E1Aw> (2023.03.22)
- <https://github.com/in28minutes/SpringBootWebApplicationStepByStep> (2023.03.12)
- <https://youtu.be/9SGDpanrc8U>(2023.03.15)
- <https://youtu.be/9SGDpanrc8U> (2023.04.01)
- <https://spring.io/projects/spring-boot>(2023.04.02)
- <https://spring.io/guides/gs/spring-boot/>(2023.04.15)
- [https://aiven.io/redis?utm\\_source=bing&utm\\_medium=cpc&utm\\_campaign=redis\\_uki\\_en\\_exact&utm\\_content=redis\\_database\\_rsa&creative=&keyword=redis%20database&matchtype=e&network=o&device=c&msclkid=d086d3bcb1931daa4faeb7e2921b1586](https://aiven.io/redis?utm_source=bing&utm_medium=cpc&utm_campaign=redis_uki_en_exact&utm_content=redis_database_rsa&creative=&keyword=redis%20database&matchtype=e&network=o&device=c&msclkid=d086d3bcb1931daa4faeb7e2921b1586)(2023.04.16)

[Binary Team/ Team 10]	<b>Specification Document</b>	[22.03.23]
------------------------	-------------------------------	------------

### -Google Maps Platform:

<https://developers.google.com/maps/get-started#:~:text=Go%20to%20the%20Google%20Maps%20PIatform%20%3E%20Credentials%20page.&text=On%20the%20Credentials%20page%2C%20click,Clik%20Close.> (2023.03.12)

### - Testing

<https://spring.io/guides/gs/testing-web/> (2023.04.15)

<https://www.baeldung.com/spring-mvc-test-exceptions> (2023.04.17)

<https://www.youtube.com/watch?v=pNiRNRgi5Ws> (2023.04.21)

### - SRS

<https://jelvix.com/blog/functional-vs-nonfunctional-requirements> (2023.05.01)

<https://www.mpug.com/back-to-basics-assumptions-vs-constraints-vs-dependencies/> (2023.05.08)

<https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>  
2023.05.01

## 4. Definition of terms –

- CSS – stands for cascading style sheets and describes how html is displayed.
- IntelliJ – an integrated development environment (IDE) aimed at Java software development - <https://www.jetbrains.com/idea/>
- Database – A structured collection of data organised in a way to support accessing this data.
- HTML – HyperText Markup Language. The main mark-up language for displaying web pages and other information that can be displayed in a web browser
- Element-plus:Element Plus is a popular UI framework for Vue.js that provides a set of ready-to-use components and tools for building responsive and interactive web applications.
- Functional Correctness – Refers to the input-output behaviour of the program e.g. for each input the program produces the correct output.
- Lib-flexible-Lib-flexible is a JavaScript library for mobile development, mainly used to solve mobile adaptation problems.
- Graphical User Interface (GUI) – A user interface that allows users to interact with the system using images rather than text commands.
- Jmeter-JMeter is an open source stress testing tool for testing web applications or other types of applications, including databases, web services, FTP services, etc.
- MyBatis:MyBatis is an open-source persistence framework for Java that provides a SQL mapping framework to map Java objects to SQL statements, and provides a framework for executing custom SQL queries and stored procedures.
- MySQL – Open-source relational database management system that runs as a server providing access to a number of databases.
- Moment.js -Moment.js is a JavaScript library that provides a simple way to parse, validate, manipulate, and format dates and times in JavaScript
- Redis-Redis is an open-source, in-memory data structure store that can be used as a database, cache, and message broker.

- Reliability – Ability of the system to perform and maintain its functions in routine circumstances as well as hostile or unexpected circumstances.
- Spring Boot – an open sourced, Java based application framework - <https://spring.io/> - Usability – The ease of use of the software application.
- TypeScript-TypeScript is a programming language developed by Microsoft, which is a superset of JavaScript. It adds features such as static typing, classes, interfaces, and other advanced language constructs to JavaScript, making it more scalable, maintainable, and robust for building large-scale applications.
- Vue.js-Vue.js is a popular open-source JavaScript framework used for building user interfaces and single-page applications. It allows developers to create reusable UI components and supports reactive data binding, making it easy to update and manage changes in the UI based on changes in the underlying data.
- Vue Router -Vue Router is a routing library for Vue.js, which provides a way to navigate between different views or pages in a Vue application.

## 5. Solution requirements

### 5.1. Functional Requirements

Requirement	Priority (H, M, L)
1.The user can return to the main screen by clicking on the home button in the top right corner	H
2.The system allows the user to filter the attractions by category in order to find the one they want to visit	H
3.System captures and displays the user's current location	H
4.System presents a list of tourist attractions within 1 mile / 2 miles / 5 miles (default sort: by distance)	H
5.System can use GPS and calculate journey speed (walk)	H
6.System allows users to rate and comment on tourist attractions.	H
7.The system allows user to select a tourist attraction and displays the information of the tourist attraction (picture/place name/phone number/email/rating/opening time/ticket price)	H
8.The system allows the user to log out of the application/profile	M
9.The system allows users to log out of their account	M
10.The system allows users to edit their password	M
11.The system allows the user to create an account	M
12.The system allows user to login to app/profile	M

13.The system allows users to view their own comments	L
14.The system allows the user to edit their username	L

## 5.2. Non-Functional Requirements

Requirement	Priority (H, M, L)
1. Performance requirements: The user expects the navigation application to be fast and responsive, and needs to ensure that the application response time and page load time should be within 2-3 seconds.	H
2. User-friendliness: users expect navigation applications to provide easy-to-use functionality and that users should be able to use the application without any guides or help.	H
3. The system can be used with the most commonly used browsers (Chrome, IE, Firefox, Safari)	H
4. Security and reliability: The user's location tracking is privacy sensitive and in order to create a more trustworthy product for the user the system uses the following strategies	M
5. Encryption - for all privacy-sensitive data that must be stored on the server.	M
6. Giving the end user control over private data (at least one option to delete the private repository).	M
7. Reliability requirements: Users expect navigation applications to be stable and reliable, not prone to crashes and failures. The system should be able to run continuously for up to 24 hours.	M
8. Easy to maintain requirements: The system's code should have clear comments, naming conventions and code structure, and the system needs to be clearly documented so that maintenance staff can quickly understand and use the system.	M
9. Ensure that the software system adheres to legal and compliance rules.	M
10. Scalability: The system should be able to service 1000 users and upgrade the server as more users are added.	L

[Binary Team/ Team 10]	<b>Specification Document</b>	[22.03.23]
------------------------	-------------------------------	------------

## 6. Other considerations

### a. Assumptions

1. It is assumed that the product can be used by users before or during their travels, thanks to its adjustable positioning mechanism.
2. Users need map and navigation functions to better navigate and locate themselves during their travels. The app needs to integrate mapping services.
3. Users need to communicate with other travelers. They require a platform to share experiences, comments, and suggestions with others.
4. Users will be using the English language
5. Users require personalized travel recommendations. It is assumed that users need to obtain travel advice and recommendations based on their personal criteria such as interests, budget, and time.
6. While the application is focussed on Newcastle, it should be easy to adapt for another city or location.
7. The app should be optimised for mobile phones.
8. The app will show at least one picture of each attraction.

### b. Constraints and Dependencies

1. Map and navigation API: The app will need to use the Google Maps API to implement navigation functionality. As we are using a third-party API service, it is important to be aware of its limitations, such as request rate limits and request count limits, to ensure that we do not exceed them when using the service.
2. Electronic map data: The app will integrate map coordinates, street view, route planning, and other data provided by third parties (such as Google Maps).
3. Database: We will use MySQL to store attraction information, such as attraction names, addresses, descriptions, pictures, etc.
4. Data source: We will find reliable data sources and attraction information through official attraction websites, social media sites, or on-site surveys.
5. User data and privacy: We will ensure that the app complies with relevant privacy laws and policies, such as GDPR and CCPA, regarding how to handle user data and privacy issues.
6. Target users: master's students of Newcastle University.
7. Interface design: The interface design will be intuitive, aesthetically pleasing, and consistent with the brand image. Colours, fonts, icons, buttons, and other elements should be unified and visually appealing.
8. The user must give permission to obtain the current location
9. The current location must be obtained before the search nearby attraction function can be used
10. Internet availability
11. availability of power
12. No new accounts can be created using existing account information
13. the user's login email and password must be valid and must consist of 6 to 16 digits or lower case letters
14. The project is to be free of charge for the duration of the development process.
15. The project will be delivered on 12 May 2023.
16. The project will be completed with all functional and non-functional requirements within that time frame.

## 7. Modification in light of comments made or change deemed necessary

### 7.1 UML final version: updated by 9<sup>th</sup> May 2023

After careful consideration and evaluation of the UML design for our product, we have made specific updates. These modifications primarily involve the feedback functionality and the addition of a "like" feature to the product review. The following changes have been implemented:

- **Feedback Functionality:**

Replaced the original feedback design with a direct provision to provide a contact email.

Updated the UML diagram to reflect this change.

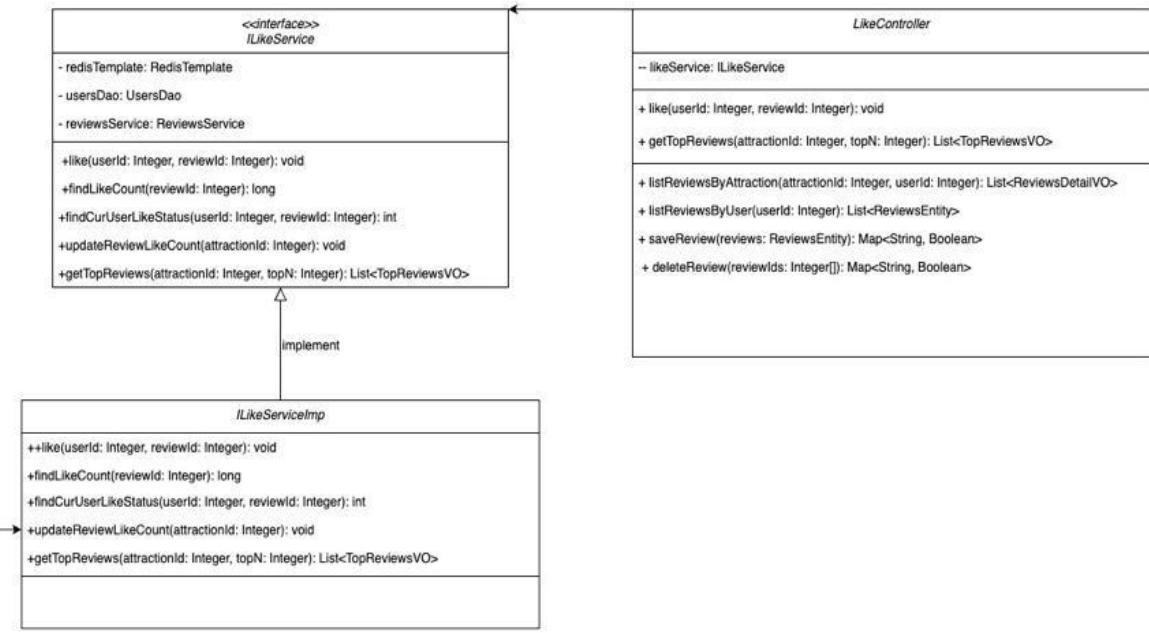
Adjusted the system components and interactions to accommodate the new email contact feature.

- **Product Review "Like" Feature:**

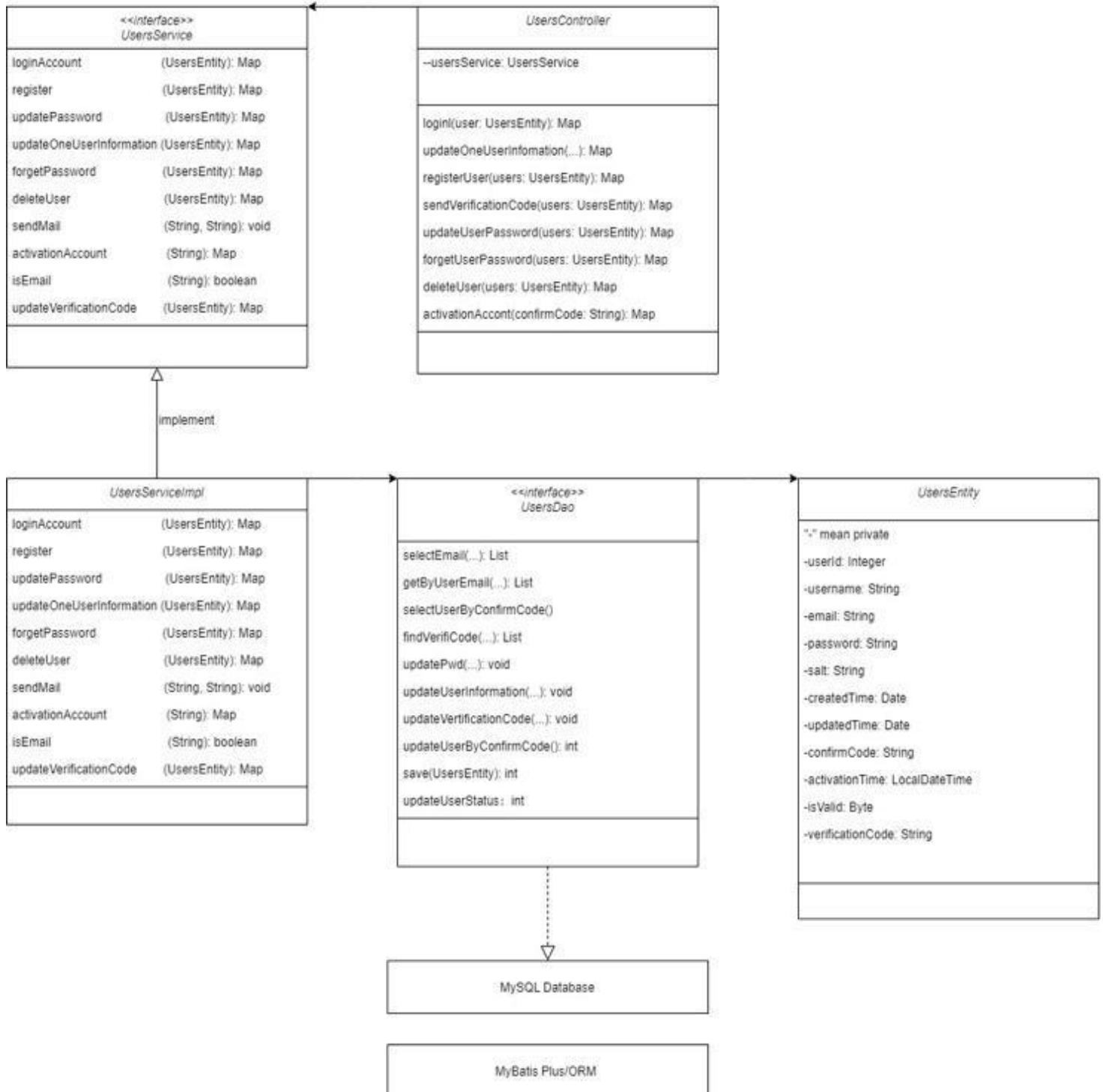
Updated the user interface design wireframes and prototypes to showcase the inclusion of a "like" button in the product review section.

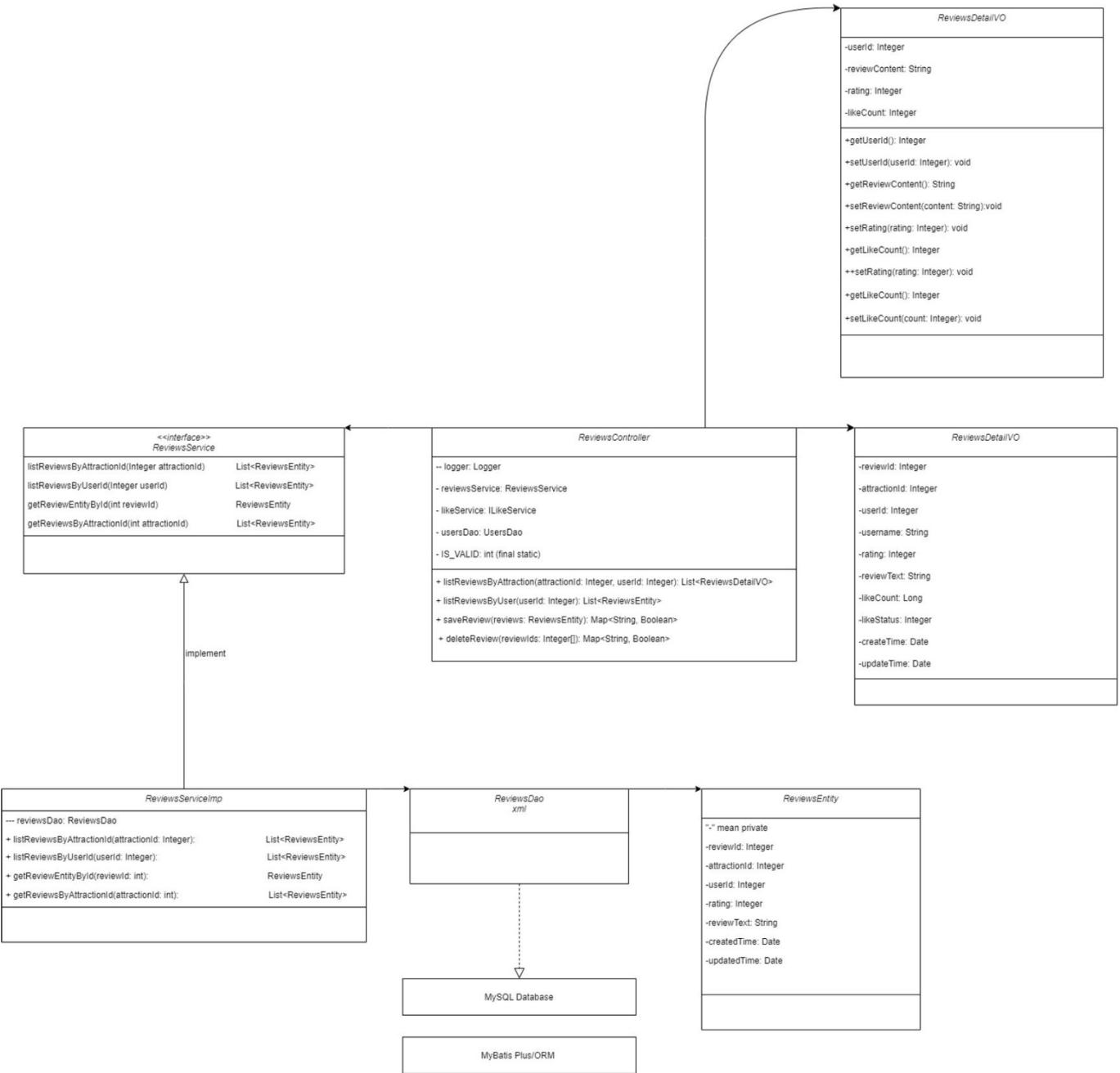
These modifications were deemed necessary to improve the overall user experience and align our product with the evolving requirements. By replacing the feedback mechanism with a direct provision to provide a contact email, we enable users to establish a more direct and personalized means of communication. Additionally, the introduction of the "like" feature enhances user engagement by allowing them to express their appreciation for valuable product reviews. These updates ensure that our UML design accurately represents the functionality and features of our product.

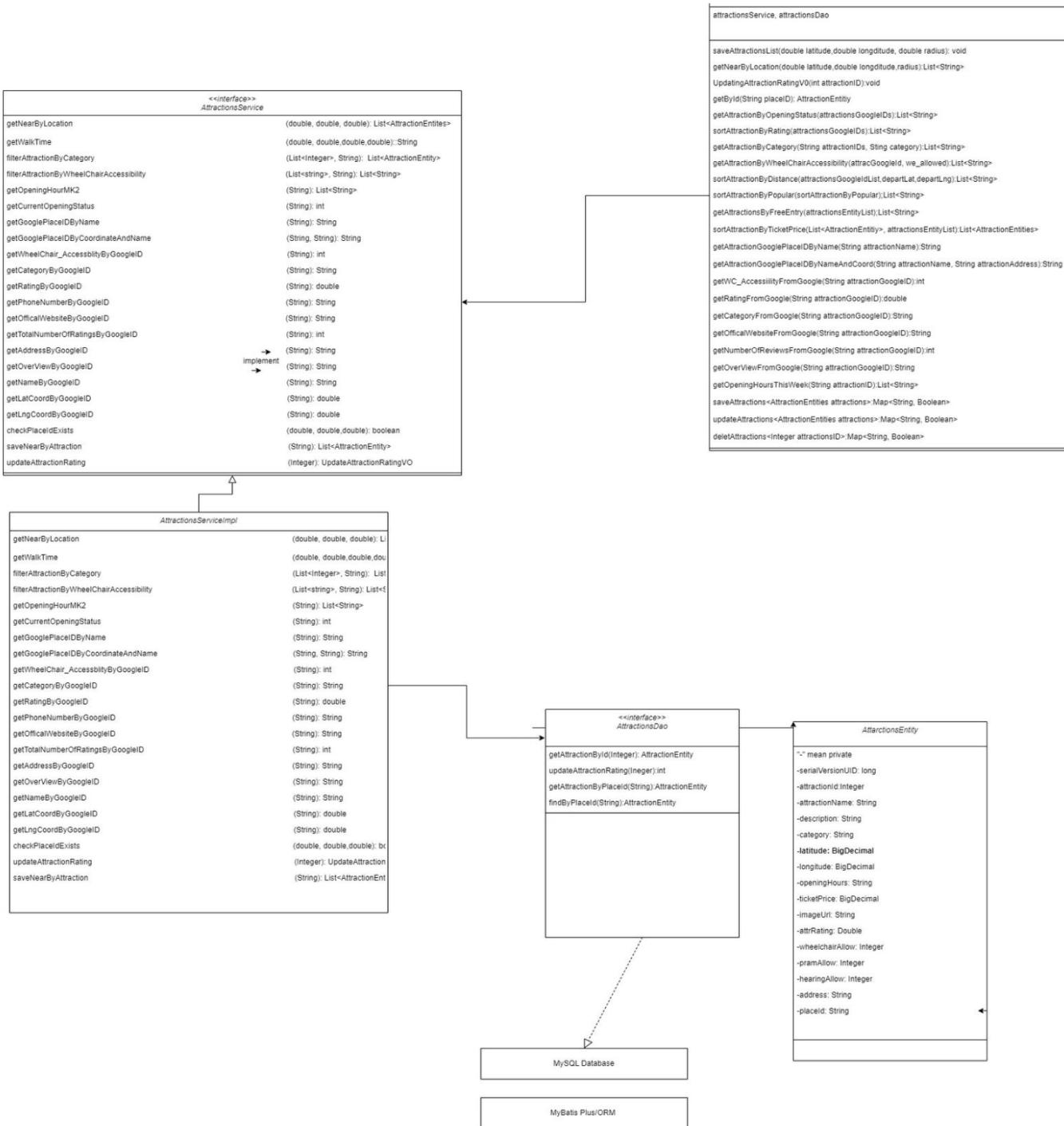
#### a. Like controller



**b. User controller**



**c. Review controller**

**d. Attraction controller**

**7.2 GUI final version:**

Following a thorough review and analysis of the specification interface design for our project, we have incorporated several updates. These modifications are aimed at enhancing the user experience, streamlining functionality, and ensuring alignment with the project requirements. The following changes have been made:

**User Interface Enhancements:**

The overall layout and visual elements of the interface have been refined to enhance aesthetics and improve usability.

Complex navigation structures have been simplified, resulting in a more user-friendly and less burdensome experience.

Consistent design elements and standardized UI components have been incorporated across different screens to promote coherence and familiarity.

The screen responsiveness and scalability have been optimized to ensure a seamless experience across various devices and screen sizes.

**Workflow Refinements:**

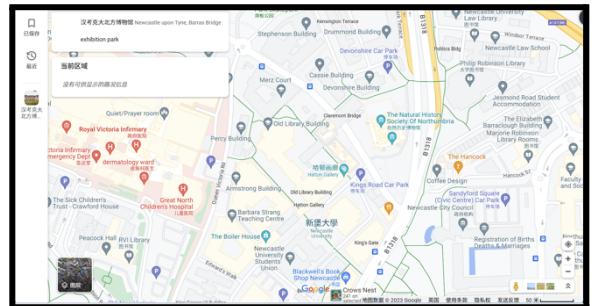
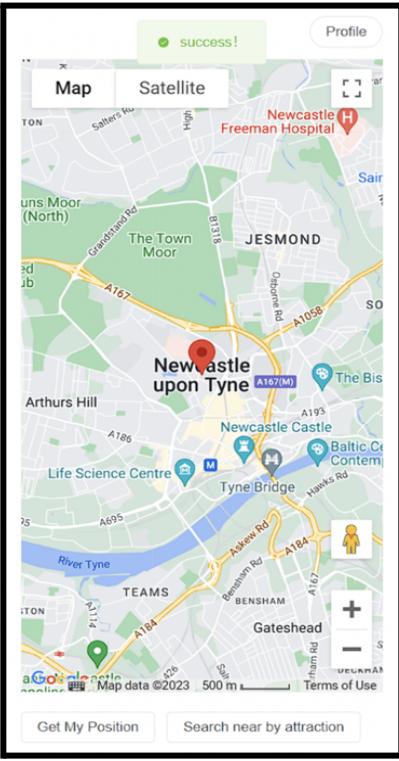
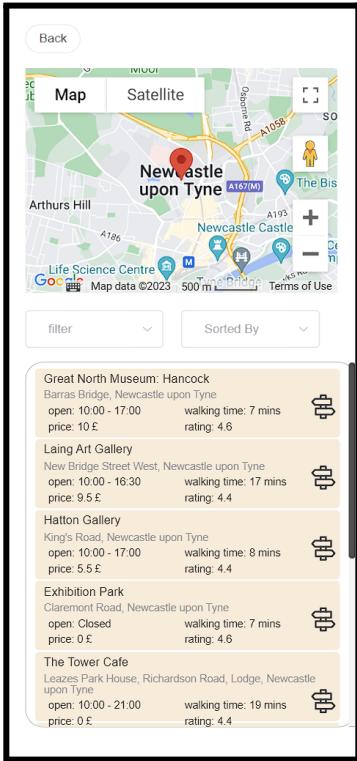
The user flow has been streamlined by removing unnecessary steps and minimizing user input requirements where feasible.

Instructions, labels, and tooltips have been improved to provide clearer guidance and facilitate user navigation through the interface.

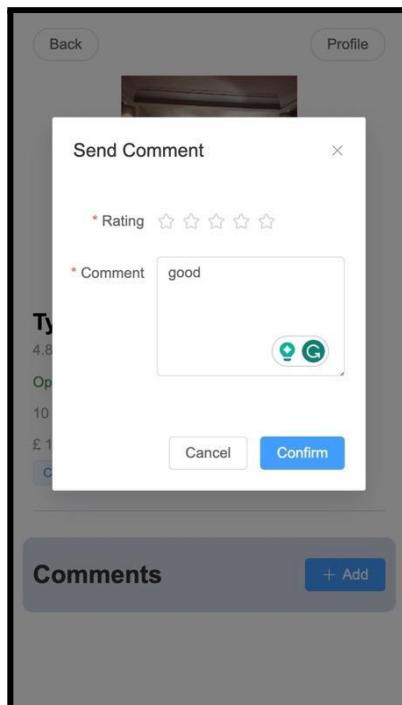
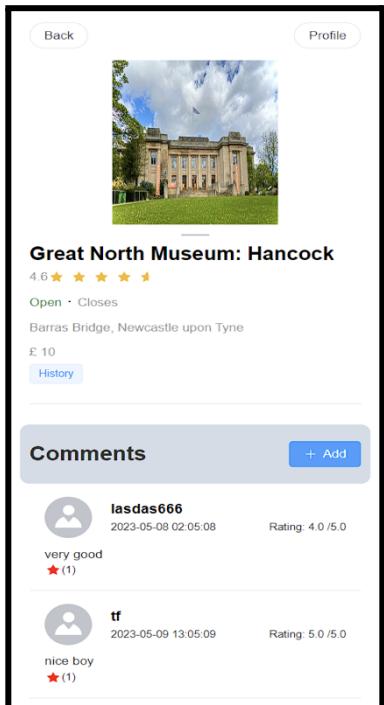
Intuitive and interactive elements have been integrated to increase user engagement and facilitate task completion.

**- Sign up and Log in**Three screenshots of the mobile application's sign-up and log-in screens. The first is a 'Sign up' screen with fields for username, email, password, and password confirmation. The second is a 'Log in' screen with fields for email and password, and links for forgot password and sign up. The third is a 'Forgot Password' screen with fields for email and verification code, and buttons for send new password and forgot password.

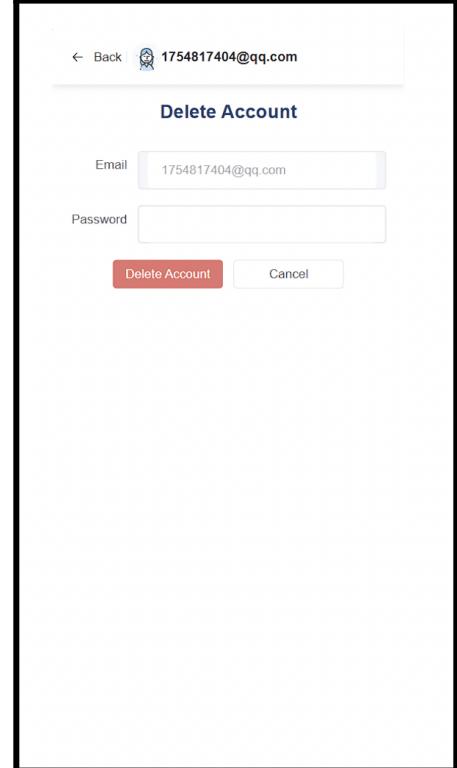
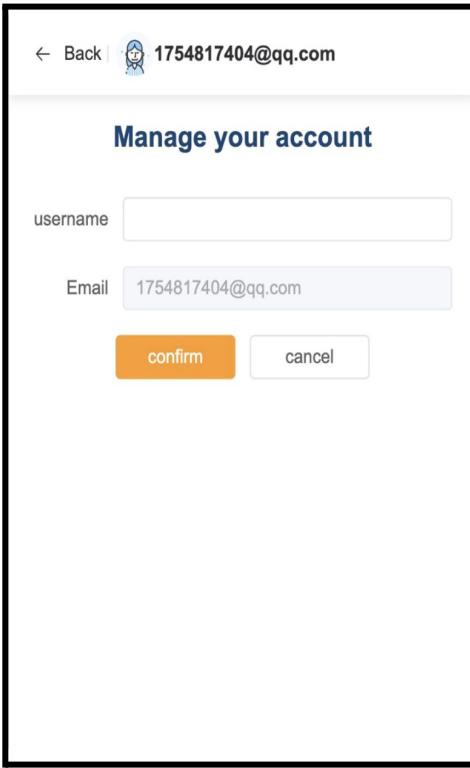
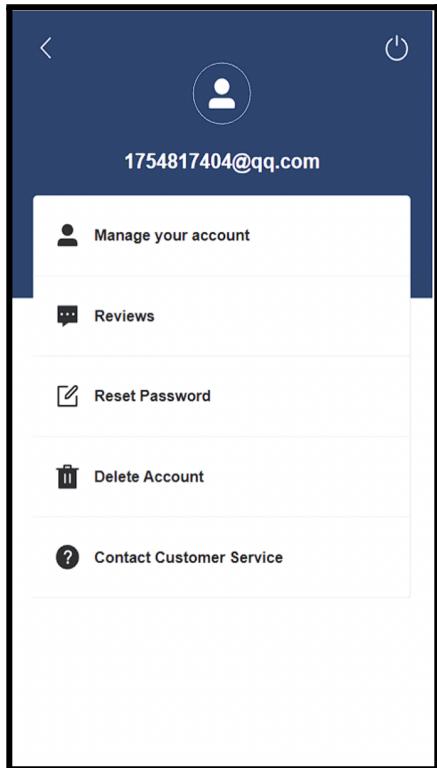
### - Map & Navigation



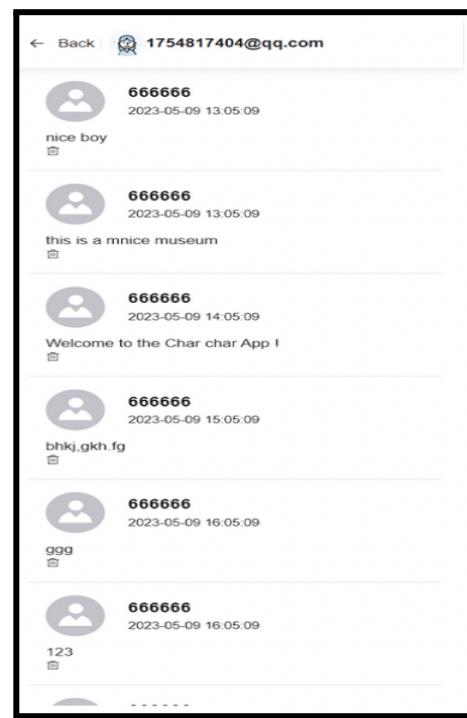
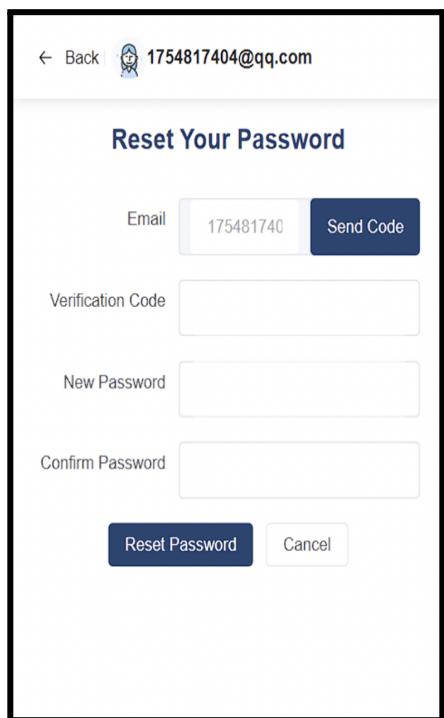
### - Attraction information and comment



- User profile / Manage account / Delete account. :

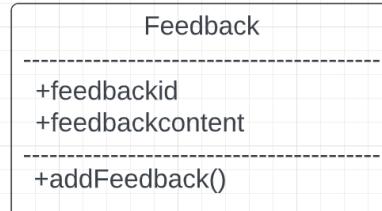
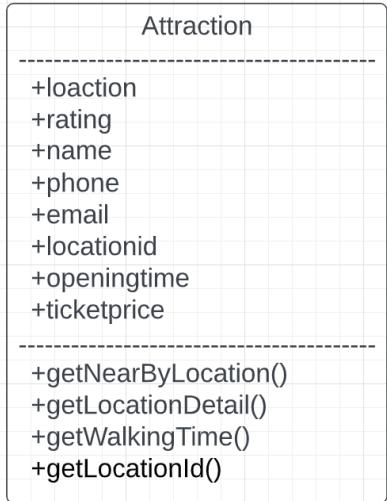
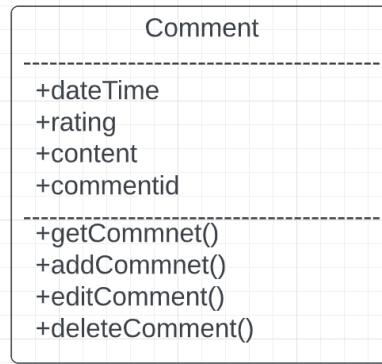
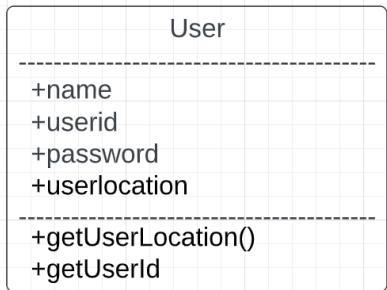


- Reset password / reviews / Contact Customer Service



**8. Initial requirements analysis of user interaction**

## 8.1 UML diagrams:



The UML diagram illustrating the basic functionality of our product utilizes the 'use dependency' relationship exclusively. As such, arrow notation has been omitted from the diagram.

## 8.2 Initial ideas for a GUI.

### 1. Interface A: Login interface:

- Account: user can type their account
- Password: user can type their password
- Log in bottom: after the user enters the account number and password and clicks the "Log in" button, the system will check whether the user is a registered member.
- Forget my password: When user forget the password, they can press this button. The system will automatically jump to the "Change Password" interface.
- Sign in: for people who want to be our users. If they click on it, the system will automatically jump to the "create account" interface.

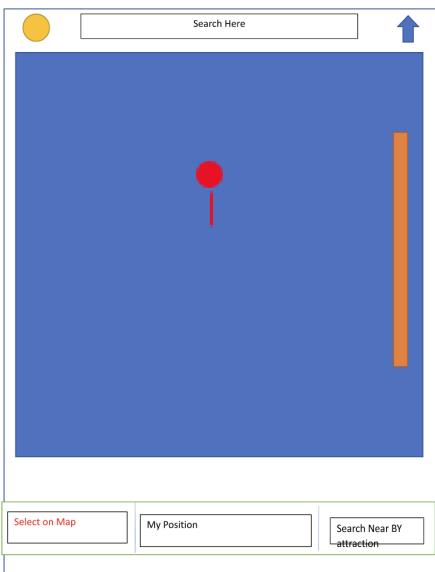


(Note: the current background image is not final and may be changed depending on the actual situation.)

### 2. Interface B: Main Interface

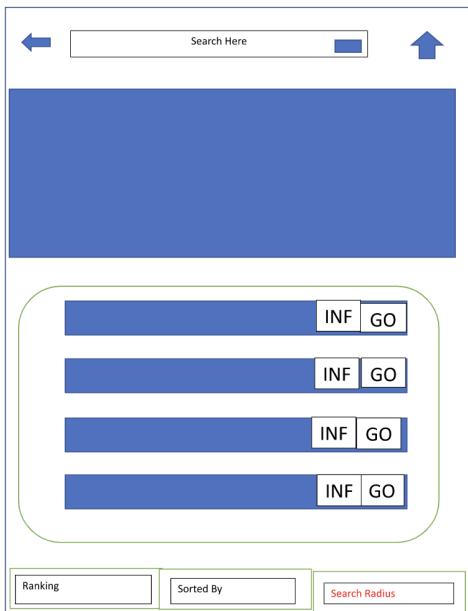
When the user enters the main interface, the system automatically navigates to the user's current location.

- Yellow bottom: jump to the "user account" interface.
- Search Here: users can search for the tourist attractions they want to visit through this search bar.
- Blue arrow: back to Main interface.
- Blue central area: map
- Red pin: The red pin is fixed in the middle of the interface, and the user can slide the map area to match the red pin to the desired attraction. Then press "Select on Map" and the system will start navigating.
- My position bottom: map jump to the user's current location.
- Search nearby attraction bottom: Attractions list pops up. (See interface C)



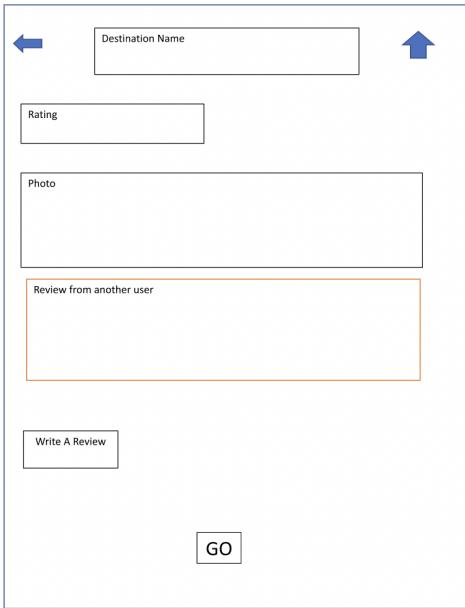
3. Interface C: information list display

- 1-mile attractions sorted by rating from highest to lowest
- INF: Basic information of attractions.
- GO: When user clicks on it, jump to attraction details interface. (see interface D)
- Sorted By: Users can change the sorting criteria. e.g., distance.



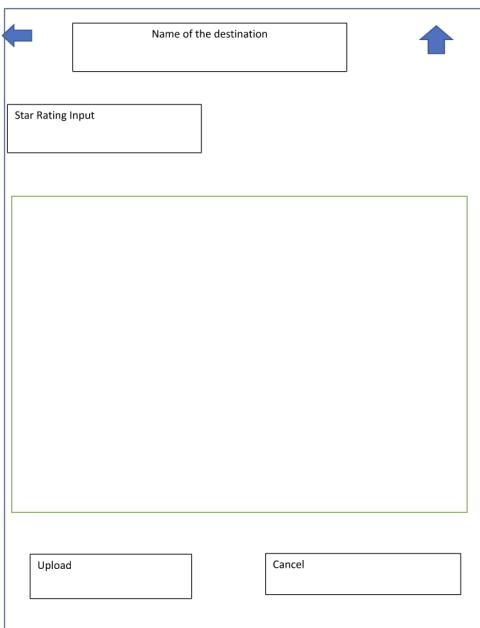
## 4. Interface D: attraction details interface

- Blue arrow: back to Main interface.
- Review from another user: The top three highly rated reviews are shown here.  
If user want to see more review, they can click on it.  
Then the interface jumps to “users reviews” interface. (See interface F)
- Write a review: If user wants to write a review, they can click it. Then the interface jumps to “Review Interface”. (See interface F)
- GO: start navigating



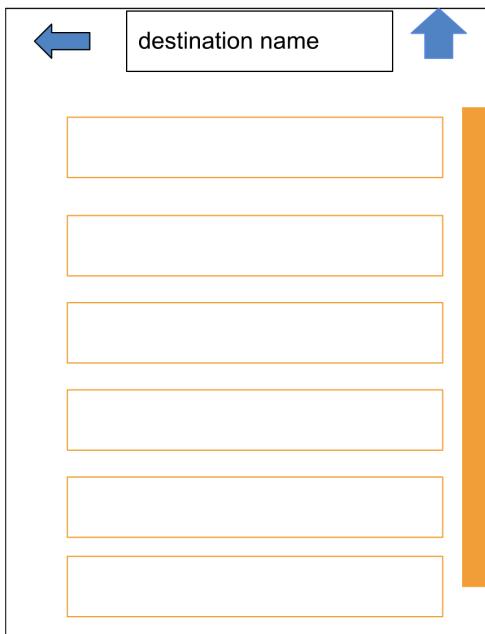
## 5. Interface E: Review interface

- Green framed area: Users can write their comments.



## 6. Interface F: User reviews interface

- Orange framed area: reviews from other users
- Sort by date (New -> Old)

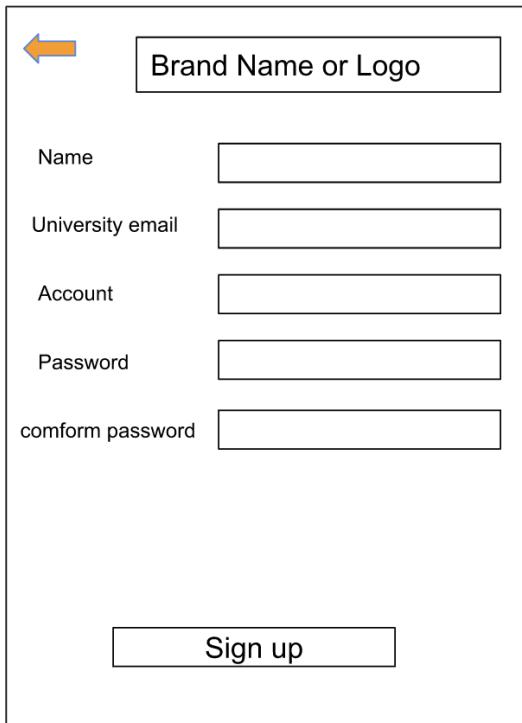


## 7. Interface G: Change password interface.



## 8. Interface H: Sign in interface

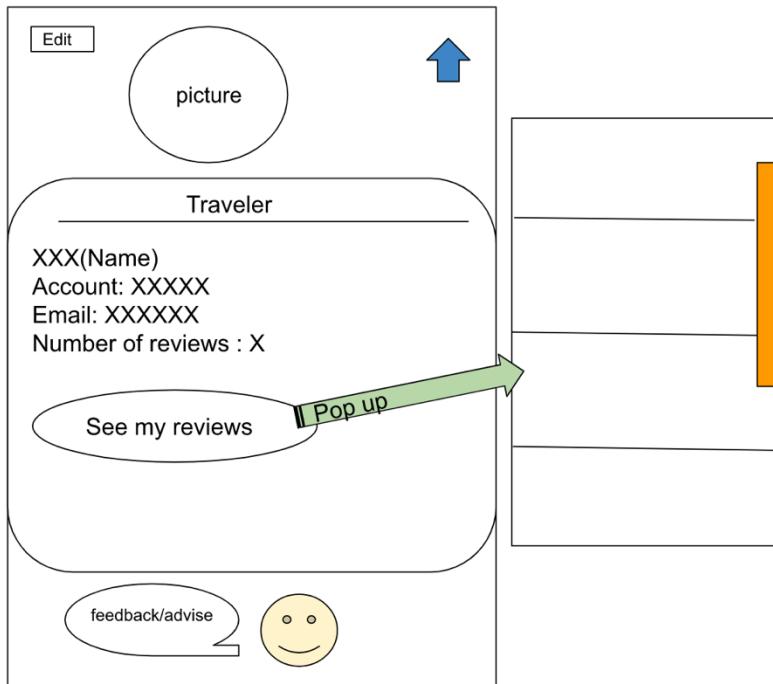
- Orange arrow: back to "Login interface" (Interface A)



A wireframe diagram of a sign-up interface. At the top is a header box containing a blue left-pointing arrow icon and the text "Brand Name or Logo". Below this are five input fields: "Name", "University email", "Account", "Password", and "comform password", each with a corresponding empty rectangular box. At the bottom is a "Sign up" button.

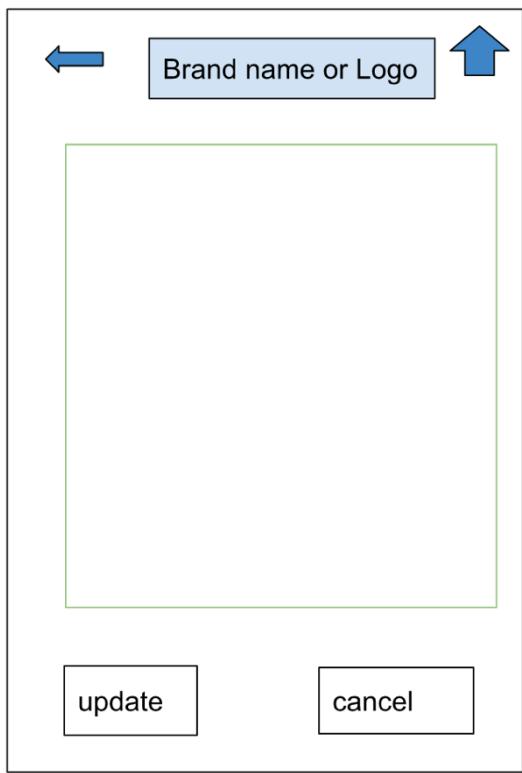
## 9. Interface I: User account interface

- See my review: pop up review list
- Feedback: jump to "Feedback interface" (interface J)



A wireframe diagram of a user account interface. On the left, a large rounded rectangle contains a "picture" (represented by a circle), an "Edit" button, and a "Traveler" section with the text "XXX(Name)", "Account: XXXXX", "Email: XXXXXX", and "Number of reviews : X". Below this is an oval button labeled "See my reviews". A green arrow labeled "Pop up" points from this button to a small rectangular window titled "Pop up" which displays a list of reviews. On the right side of the main interface, there is a vertical orange bar.

## 10. Interface J: Feedback interface



### 8.3 Use cases – Describe how the user accesses functionality.

In general, the operation scenarios can be roughly divided into 2 categories, prior to visiting the attraction and after the visit.

#### Scenario A

Before the visit, a user can use the app to accomplish the following settings related actions.

1. Change the password and other account settings.
2. Leave feedback to our developer team.

Scenario	Operation Process
A-1	<ul style="list-style-type: none"> <li>■ (Interface A, Setting Button)</li> <li>■ (Submenu, Change Password)</li> </ul>
A-2	<ul style="list-style-type: none"> <li>■ (Interface A, Setting Button)</li> <li>■ (Submenu, Feedback)</li> </ul>

#### Scenario B

Before the visit, a user might acquire the information of desired attraction via the following action sets.

1. The user knows nothing about the surrounding tourist attraction, decided to use the auto search feature to gain some information about the nearby attractions.
2. The user doesn't have clear liking on which specific attractions, so she or he decided to use satellite map to visually search appealing features and attractions.
3. The user just wants to know his or her current location to get a bearing.

Scenario	Operation Process
B-1	<ul style="list-style-type: none"> <li>■ (Interface A, My Position Button)</li> <li>■ (Interface A, Search Nearby Attraction Button) → (Interface B)</li> </ul>
B-2	<ul style="list-style-type: none"> <li>■ (Interface A, Pick on the Map button)</li> </ul>
B-3	<ul style="list-style-type: none"> <li>■ (Interface A, My Position Button)</li> </ul>

### Scenario C

Before the visit, the user has made some search (Scenario B) input but wish to refine the search results.

1. The user wishes to refine the listing by the category.
2. The user wishes to refine the listing by ranking the ratings.
3. The user wishes to re-input the search wording and then start a new search.
4. The user wishes to expand or shrink the search radius.

Scenario	Operating Process
C-1	■ (Interface B, Sort Button)
C-2	■ (Interface B, Ranking Button)
C-3	<ul style="list-style-type: none"> <li>■ (Interface B, Search Bar)</li> <li>■ OR</li> <li>■ (Interface B, Back Button)</li> <li>→(Interface A)</li> <li>■ (Interface A, Search Bar)</li> </ul>

### Scenario D

Before the visit, the user input the search word and acquired a list of search results, he or she wishes to examine the attraction in detail.

1. The user wishes to access the detail information on Interface D.
2. Users click into the attraction details to leave comments

Scenario	Operating PRocess
D-1	■ (Interface B, INFO Button)
D-2	■ (Interface B, INFO Button)→(Interface D) add a comment

**Scenario E**

Before the visit, the user has input some search work and acquired a search results, he or she wishes to start the navigation.

1. After gain the listing of desired attractions, he decided to pick on and start going there.
2. After gain the detail information of a specific attractions, he decided to go to the place immediately.

Scenario	Operating Process
E-1	■ (Interface B, GO Button)
E-2	■ (Interface B, INFO Button) →(Interface C) ■ (Interface C, GO Button) →(Interface D)

**8.3.1 Use case diagram:**