

Software Specification: Tripedia

Yulin Zhang

7th Group

Software Engineering

Montreal, Canada

silveralex2023820@gmail.com

Yuhang Chen

7th Group

Software Engineering

Montreal, Canada

yuhang.chen@mail.concordia.ca

Jiaxi Yang

7th Group

Software Engineering

Montreal, Canada

yjxyang2@outlook.com

Boyang Wang

7th Group

Software Engineering

Montreal, Canada

wangboyang0626@outlook.com

I. INTRODUCTION

A. Problem Statement

Travel has gradually become an indispensable part of people's lives. According to relevant research, 83% of Canadians will travel to different cities to experience different customs during their holidays. Travel planning has become a problem that troubles most Canadians. At the same time, many people don't know which places are more suitable for them. Therefore, travel recommendations and planning have become essential issues affecting people's quality of life.

B. Architecture Design

As shown in Figure1, the ICDE system architecture mainly contains two parts: client and server. Based on the limited server hardware equipment, we implement the data analysis logic as a single module in the server, which stands for ICDE 3rd Party Applications.

In the client part, we implement Tripedia's client logic based on the web. The web client service is mainly divided into two parts: the Data Capture Module and User Assistant Module. The Data Capture Module captures the user's behavior and personal information through questionnaires and behavior captures, and transmits the relevant information to the server. The User Assistant Module is responsible for displaying processed data sent by the server, such as recommendation results and data analysis results, which help users make decisions easily.

On the server side, we will develop three modules to implement the backbone of the ICDE system: Data-processing Module, Data-analyzing Module, and Data Access Service. The Data Access Service, as the median layer for database access, provides a unified database interface (deletion, modification, query, etc) for the Data-processing Module and the Data-analyzing module. The Data-processing Module is responsible for the basic logic of the server, such as registration, login, search, data preprocessing, and other interaction logic with the client. The Data-analyzing Module accesses the data in the database through the Data Access Service to perform data analysis and returns the analysis results to the Data-processing Module.

To sum up, Tripedia's architecture is a standard ICDE system. The Client part includes a Data Capture Service to collect user behavior and information and transmit relevant data to the server. The Data-processing Module in the Servers

section is responsible for implementing the interactive logic of the server; the Data Access Service provides a unified data access interface; the Data-analyzing Module serves as the core of the ICDE system to analyze user data and feedback the analysis results to the user through the server to provide User Assistance Service.

II. OVERVIEW

A. Project Goal

The goal of the project is to solve the travel planning difficulties faced by users. In response to travel planning issues, this project set out to develop search features to help users. Meanwhile, some users don't know where is more suitable for travel. This project provides clear charts through the background data analysis function to help users make better decisions.

B. Key Functionalities

Search The search feature is responsible for obtaining search results based on the user's search keywords and collected user preferences. This function includes searching for tourist cities, searching for hotels, etc. The search function can provide users with travel route planning.

Data Analysis The website provides a data analysis function if the user doesn't know where is better for vacation. Based on current user preferences and existing data in the system, the website provides data analysis charts of tourist cities suitable for users to help users make better decisions.

C. User Stories

1. The traveler opens the travel website without logging in, the website will collect his location, so that the website can show local people's favorite place to visit which provides a reference for the user.

2. The traveler registers an account and login to the Tripedia website.

3. The traveler finishes a questionnaire so that the website can recommend appropriate information for users based on the questionnaire content.

4. The traveler has no idea about the where is the best places for him. He finds the data analysis diagram panel for more information to make a decision.

5. The traveler searches for his destination and then the website recommends related information (e.t. hotel, transport

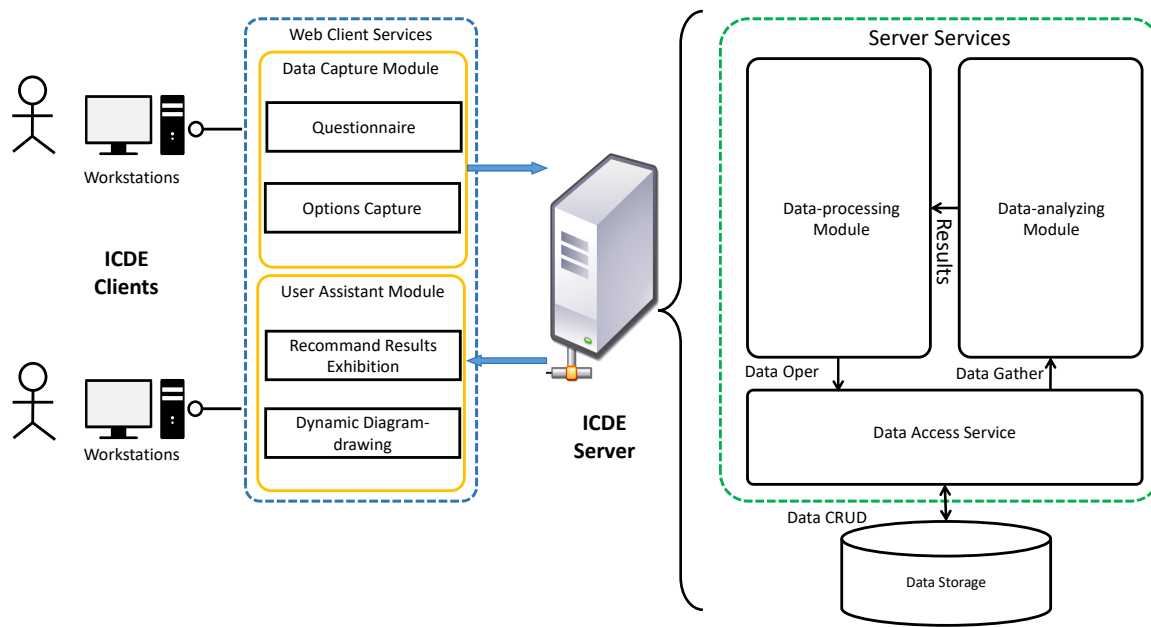


Fig. 1. The architecture graph of Tripedia.

function, Landmark Tickets, scenic spot review) so that he can make plans by reference.

6. The traveler checks suitable hotel timings so that he can book an appropriate hotel.

7. The traveler checks suitable flight or train timing so that he can book an appropriate transport function.

8. The traveler selects his favorite hotel and then clicks to jump to the booking screen so that he can make payment.

D. User Cases

1) User Cases: Open Website:

Use Case: Open Website.

Actors: Traveler, Server.

Summary Description: Allows any travelers to open Tripedia, and Tripedia shows some basic information about traveling according to the traveler's position.

Status: Medium level of details.

Pre-condition: The traveler has to have a personal computer and an available network environment.

Post-condition: Traveler has to register an account and log in to the Tripedia website to use its service.

Basic-Path:

1. The traveler enters the Tripedia website in a browser.
2. The server recommends the travel information entry based on the user's position.

3. The website displays the available travel information sent by the server.

Alternate Paths:

None.

2) User Cases: Register and Login:

Use Case: Register and Login

Actors: Traveler, Server.

Summary Description: Allows any traveler to register an account and log in to the Tripedia website.

Status: Medium level of details.

Pre-condition: The traveler has to open the Tripedia website without login.

Post-condition: Traveler has to complete the questionnaire to use Tripedia's full service.

Basic-Path:

1. The traveler clicks the "register" button on the home page.
2. The website jumps to the register page
3. The traveler fills in the register information: user name, nationality, gender, and password.
4. The traveler clicks the "submit" button.
5. The server validates the username is repetitive in the database.
6. The server saves the register information to the database.
7. The website displays the "register successfully" information and jumps to the login page.

Enter Website Activity Diagram

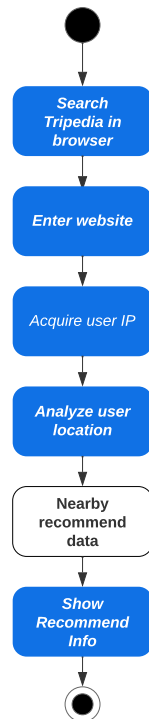


Fig. 2. The activity diagram of Open Website.

8. The traveler clicks the "login" button on the home page.
9. The traveler fills in the user information, username, and password.

10. The traveler clicks the "login" button.

11. The server validates the username with the password.

12. The website jumps to the home page.

Alternate Paths:

5a. The user name is repetitive in the system.

11a. User forgets username/password.

11b. Invalid username/password.

3) *User Cases: Questionnaire:*

Use Case: Questionnaire

Actors: Traveller, Server.

Summary Description: Allows any traveler within the login state to have opportunities to complete the questionnaire to get appropriate recommendations.

Status: Medium level of details.

Pre-condition: Traveler must be in login state.

Post-condition: Traveler has to use the search function to get more information.

Basic-Path:

1. The website pops up a dialog to ask the user to fill out a questionnaire.

2. The traveler clicks "yes", and the website jumps to the questionnaire page.

3. The traveler can complete questions about preference.

Regist/Login Activity Diagram

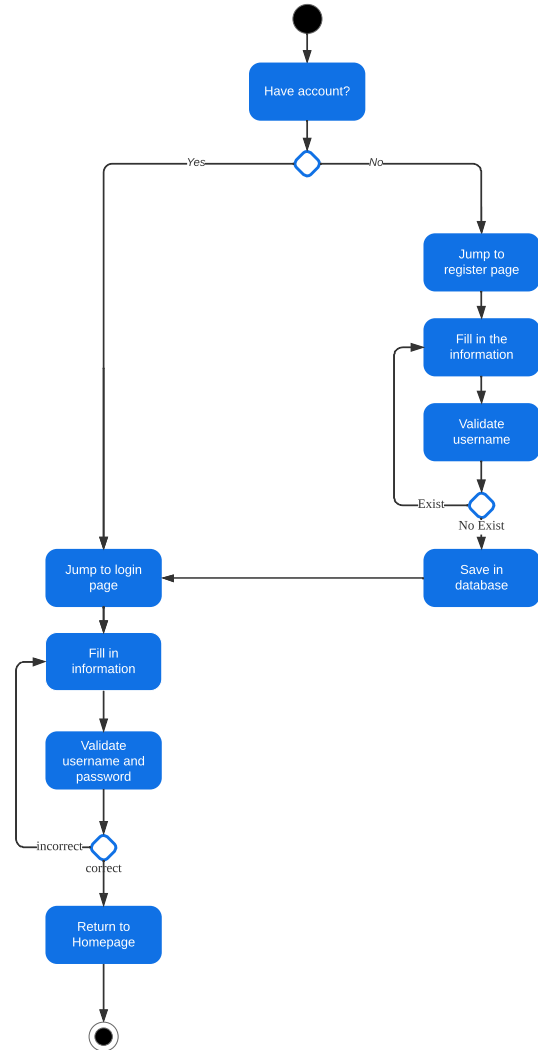


Fig. 3. The activity diagram of Login.

4. The traveler clicks the "submit" button.

5. The server stores the preference data in the Database.

Alternate Paths:

2a. The user abandons the chance to complete the questionnaire.

4) *User Cases: Data Analysis Diagram:*

Use Case: Data Analysis Diagram

Actors: Traveller, Server, Data Analyzing System.

Summary Description: Allows any traveler within the login state to check the data analysis diagram.

Status: Medium level of details.

Pre-condition: Travelers must be in login state and the website is on the home page.

Post-condition: Travelers can search the recommended cities in the search area for more information.

Basic-Path:

1. The traveler clicks the "Analysis Panel" button.

Questionnaire Activity Diagram

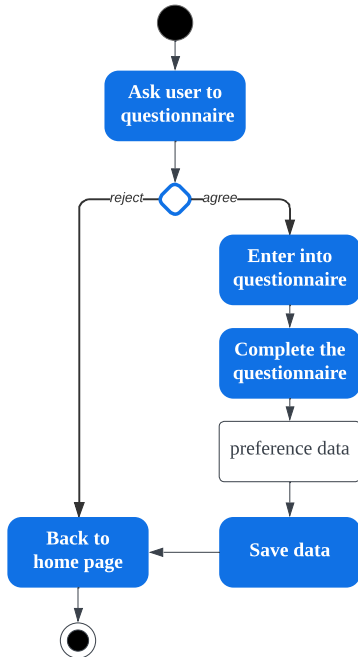


Fig. 4. The activity diagram of Questionnaire User Case.

Diagram Activity Diagram

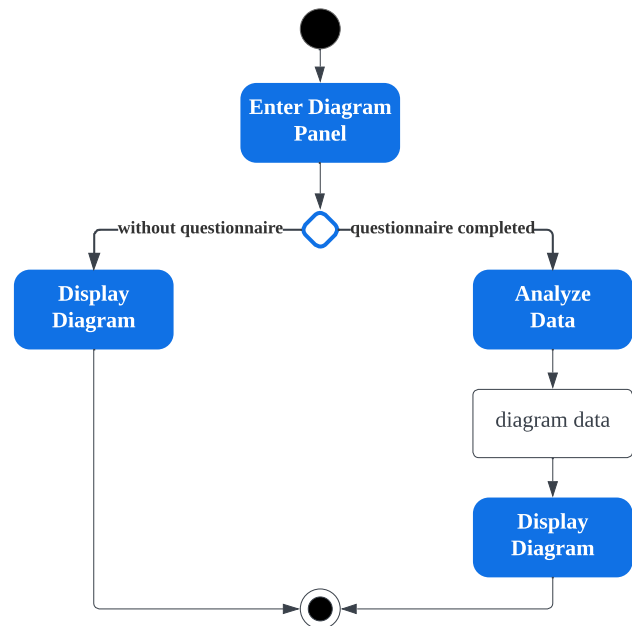


Fig. 5. The activity diagram of Data Analysis Diagram User Case.

2. The website jumps to the Data Analysis Page.
3. The data analyzing system analyzes the user's preference data.
4. The server processes the analyzed data.
5. The website displays the diagrams of recommended cities, such as the visitors' number, the visitors' nationalities proportion, etc.

Alternate Paths: None.

5) *User Cases: Recommend Related Information:*

Use Case: Recommendation

Actors: Traveler, Server, Data Analyzing System.

Summary Description: Allows any traveler to search for his destination, then the website recommends related information.

Status: Medium level of details.

Pre-condition: The traveler has to open the Tripedia website.

Post-condition: The website displays the relevant recommendation information.

Basic-Path:

1. The user enters the website.
2. The user enters the traveling destination in the input box.
3. The user clicks the 'search' button.
4. The data analyzing system analyzes the preference data of the user.
4. The server processed the analyzed data.
5. The website displays the relevant recommendation information in the form of a list.

Alternate Paths: None

Recommendation Activity Diagram

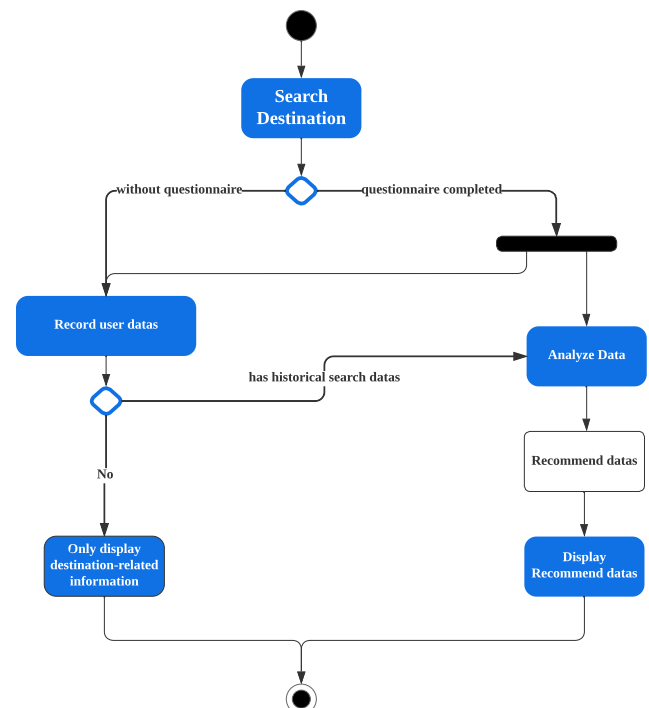


Fig. 6. The activity diagram of Recommend Related Information User Case.

6) *User Cases: Search Hotels: H*

Use Case: Search Hotels

Actors: Traveler, Server.

Summary Description: Allows any traveler to search appropriate hotels.

Status: Medium level of details.

Pre-condition: Travelers must be in login state and the website is on the home page.

Post-condition: The website displays the appropriate hotels list.

Basic-Path:

1. The user enters the website.
2. The user enters the destination in the input box.
3. The user chooses the check-in date and check-out date from the date form.

4. The user clicks the 'search' button.
5. The server collects the target hotels' data.
5. The website displays suitable hotels.

Alternate Paths: None

Searching Hotel Activity Diagram

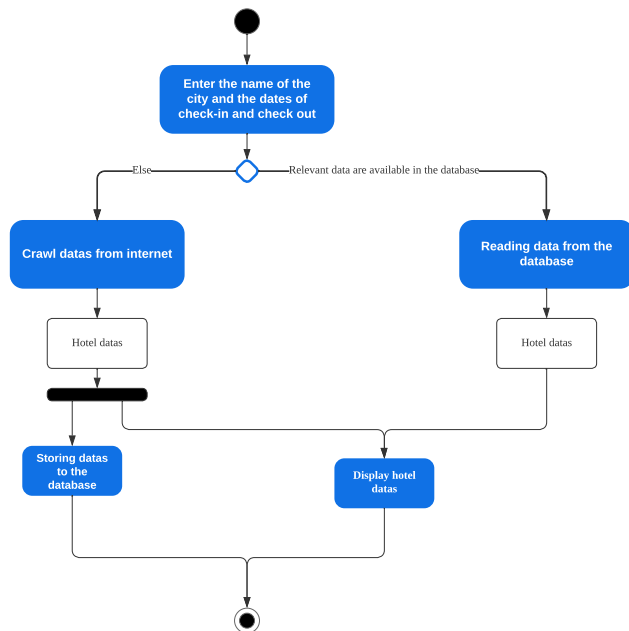


Fig. 7. The activity diagram of Search Hotels User Case.

7) *User Cases: Search Transportation Tickets:*

Use Case: Search Transportation Tickets

Actors: Traveler, Server.

Summary Description: Allows any traveler to search for the right mode of transportation

Status: Medium level of details.

Pre-condition: The traveler has to open the Tripedia website.

Post-condition: The user can choose his favorite transportation function.

Basic-Path:

1. The user enters the website.
2. The user enters the destination in the input box.
3. The user chooses departure time and return time from the date-form.
4. The user clicks the 'search' button.
5. The server collects the transportation data. (include flights or trains)

6. The website displays suitable tickets.

Alternate Paths: None

Search Transportation Tickets Activity Diagram

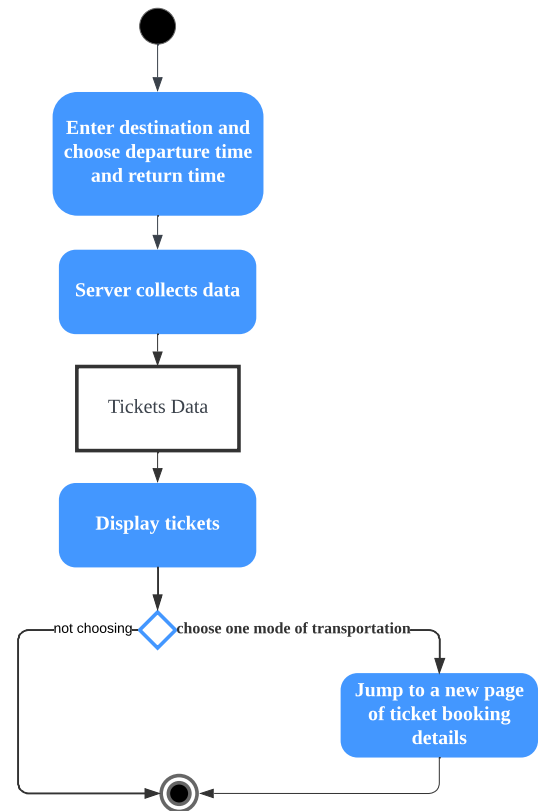


Fig. 8. The activity diagram of Search Tickets User Case.

8) *User Cases: Check Hotels:*

Use Case: Check Hotels

Actors: Traveller, Server.

Summary Description: Allows any traveler to check the details of the hotel.

Status: Medium level of details.

Pre-condition: The traveler has to open the Tripedia website.

Post-condition: The user clicks one of the hotels, and the website jumps to a new page of hotel booking details.

Basic-Path:

1. The user enters the website.
2. The user enters the destination in the input box.

3. The user chooses the check-in date and check-out date from the date form.
 4. The user clicks the 'search' button.
 5. The server collects the hotel's data
 6. The website displays suitable hotels.
- Alternate Paths:** None

Check Hotel Activity Diagram

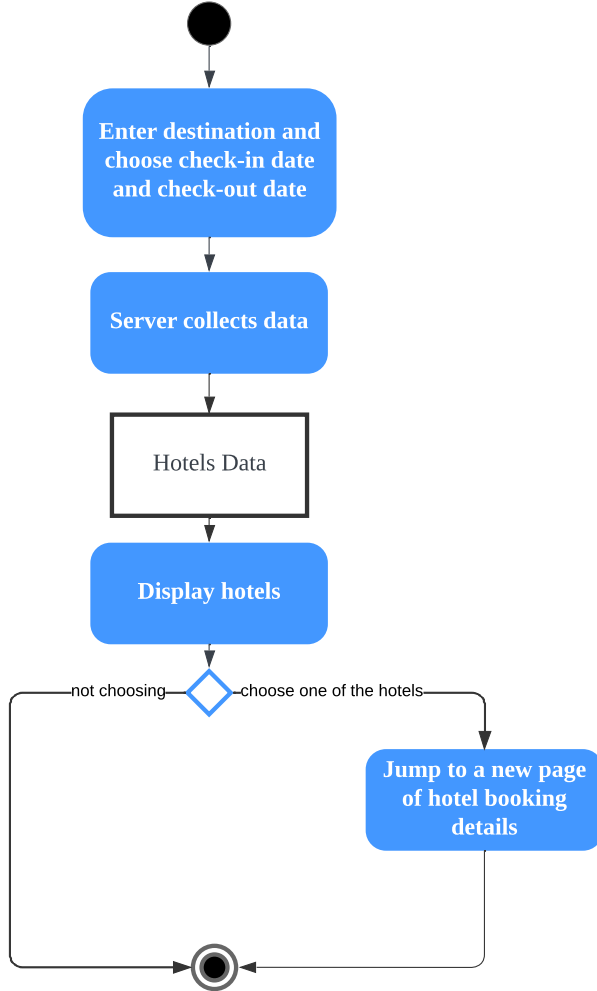


Fig. 9. The activity diagram of Search Hotels User Case.

E. User Case Diagram

III. FUNCTIONAL REQUIREMENTS

A. The Requirements Ordering

We should consider the importance of each requirement and confirm the ordering of requirements. We divide the requirements into two types: **mandatory** and **optional**. A mandatory requirement is essential in the system, so these requirements are the first targets to implement for us. As for optional requirements, these requirements are designed to

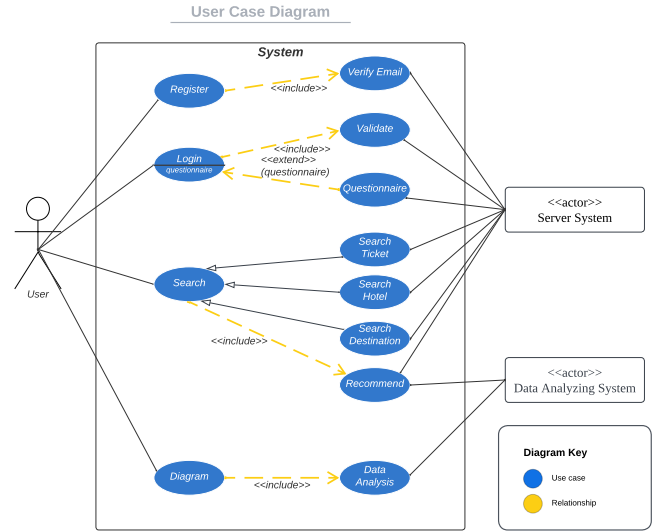


Fig. 10. The user case diagram of Tripedia.

improve the usability and consummate the functions in the system. We would like to complete these requirements in the incremental stage.

B. Login (mandatory)

As the user case: Register and Login shows, the requirement of Login is an essential functional feature. If users have their accounts, they can log in to the website and get all services. This requirement can be divided into three sub-requirements: Enter Username and Password, Validation, and Edit Personal Information.

1) *Enter Username and password(mandatory)*: The users can enter their username and password in the input box. The user can click on the Show Password button in the password box. When the user clicks on this button, the password will be displayed in plaintext in the password box so that the user can check if the password is correct before logging in.

2) *Validation(mandatory)* : When the user clicks on the login button if the user enters an incorrect password or username, the message "Username or password error" will be displayed. If the user enters the correct username and password, the system will automatically jump to the home page, and the user's login status will be displayed on the home page.

C. Register(mandatory)

Users can create a new account by entering a username, name, password, email, and other relevant personal information. This requirement can be divided into two sub-requirements: Enter Personal Information, and Verify Email Address.

1) *Enter Personal Information(mandatory)*: When the user clicks on the register button on the home page, the page will display several input fields to let the user enter his personal

information. After finishing the input, the user can click the “Register” button to create a new account.

2) *Verify Email Address(mandatory)* : After the user inputs their email address, they can click the “Verify the email” button, and then the system will send an email to the address. The user should open the email and click the link in the email to finish the registration. If the user doesn’t receive the email, they can click the “don’t receive the email” button and the system will send the verification email again.

D. *Edit Personal Information(optional)*

1) *Change Password(optional)* : When the user wants to change their password, they can enter their account page, and click “Change the password” to edit it. The user needs to type the old password first to ensure safety. If the user inputs the wrong password, a system will display the message “ wrong password. You can’t change the password.” If the password is right, the user can type the new password and they should input it for the second time to confirm the new password. After confirming the new password, the user clicks on the “Save” button to save the change.

2) *Update Email Address(optional)* : When the user wants to change their email address, they can enter their account page, and click “Change the email address” to edit it. The user can input the new email address and click the “Verify the email” button, and then the system will send an email to the address. The user should open the email and click the link in the email to finish the update. If the user doesn’t receive the email, they can click the “don’t receive the email” button, and the system will send another email.

3) *Edit Other Information(optional)* : When the user wants to change other personal information, they can enter their account page to edit it. After inputting the new information, click the “Save” button to save the edit.

E. *Search(mandatory)*

Users can enter keywords, locations, or other criteria to search for travel destinations. A list of related travel destinations is displayed and the user can click to view details.

1) *Search Method(mandatory)* : Users should be able to search for the information by using the search bar. When users are typing, the system will list the related entries under the search bar to help users to search. Moreover, Users should be able to use filters and restrictions to narrow down search results. The system should be able to save the user’s search history so that the user can quickly find the required information the next time.

2) *Search Result Display(mandatory)* : Search results should be clear and sorted in order of relevance so that users can quickly find the information they need. When there are too many search results, pagination should be supported. Users can enter the details page by displaying the results, which introduces specific information about the scenic spot, and the user can enter the official page of the tourist destination through the link displayed on the website to learn more.

F. *Data Analysis Diagram(mandatory)*

As the User Cases: Data Analysis Diagram shows, users can check the data analysis diagrams calculated by the ICDE system. Meanwhile, the precondition of the Data Analysis Diagram is the Questionnaire, so the data analysis diagram requirement combines the processing of the diagram and questionnaire. Furthermore, the Data Analysis Diagram requirements can be divided into three sub-requirements: Questionnaire, Data Analysis, and Diagram Display.

1) *Questionnaire(mandatory)* : When the user first logs in to the website, the website pops up a diagram asking the user to complete a questionnaire to get an accurate recommendation. If the user chooses to complete the questionnaire, the system will jump to the questionnaire page. The questionnaire page contains several one-choice questions, and at the end of the page, there is a submit button and a quit button. If the user clicks the “quit” button, the website jumps to the home page. After completing the questionnaire, the website will send the questionnaire information to the data-analyzing module. At the same time, the server program(data-processing module in architecture diagram) save the questionnaire information to the database.

2) *Data Analysis(mandatory)* : The sub-requirement of Data Analysis is not visible to users, but it is an essential requirement in an ICDE system. While the information on the questionnaire has been sent to the data-analyzing module, the data-analyzing module will calculate the recommendation weights of cities based on the user’s information. We will design a KNN algorithm as the core algorithm for data analysis.

3) *Diagram Display(mandatory)* : When the user clicks the “Analysis Panel” button on the home page, the website jumps to the diagram display page. The website receives the information processed by the data-analyzing module and shows the results by several diagram types. The diagrams are initially designed as follows:

- * The most appropriate travel destination: pie chart.
- * The most popular travel destination: bar chart.

IV. NONFUNCTIONAL REQUIREMENTS

A. *Usability*

The website’s interface should be easy to use, clear, and appropriate for all age groups of users. The user interface should be simple and intuitive. Elements such as color, fonts, and layout should all contribute to the user experience. When the user enters incorrect information, there will be a clear error message to guide the user on how to correct the error.

B. *Security*

To ensure the security of user passwords, the system performs a hash processing to encrypt the passwords instead of storing them in the database in plaintext.

C. *Compatibility*

1) *Browser compatibility* : The website is compatible with mainstream browsers such as Chrome, Firefox, Safari, etc.

2) *Operate System compatibility* : Users can access the server page through computer operating systems such as Linux, Windows, macOS, and mobile device operating systems such as iOS and Android.