



University College Dublin

Beijing-Dublin International College

System Document

Author:

Jiahe Zhang 20205722

Supervisor:

Dr. Brett Becker
Dr. Catherine Mooney

An Assignment submitted for the BDIC:

COMP3030J Software Engineering Project 2 - 2022-2023

November 24, 2023

Contents

1 Abstract	2
2 Overall	2
2.1 introduction	2
2.2 Technologies	2
3 Teamwork	3
3.1 Introduction	3
3.2 Team Structure	3
3.3 RACI Matrix: individual Responsibilities	4
4 Plan	4
4.1 Milestones	4
4.1.1 Milestone 1: Chat Module(2023.5.1~2023.5.8)	4
4.1.2 Milestone 2: Profile Module (2023.4.17~2023.5.1)	4
4.1.3 Milestone 3: Travel Project Information Module (2023.3.26~2023.4.10)	4
4.1.4 Milestone 4: Travel Project Management Module (2023.3.13~2023.3.27)	5
4.1.5 Milestone 5: Reservation Management Module (2023.4.3~2023.4.17)	5
4.1.6 Milestone 6: Portal Module (2023.3.6~2023.5.22)	5
4.2 Timelines	5
5 Requirements	5
5.1 Functional Requirement	5
5.2 Non-Functional Requirement	6
5.3 Use case Diagram	6
6 System Architecture	7
6.1 Front-end	7
6.2 Back-end	7
6.3 Database	8
7 Design & Solution	8
7.1 Functional Requirement Implementation - Customer Side	8
7.1.1 Maps and Street Views	8
7.1.2 Data Visualizations: Footprint Wall, Heat Maps, and Charts	9
7.1.3 Efficient and accurate project recommendation services based on auxiliary tools	9
7.1.4 Diversified display of program information	10
7.1.5 Browse Travel Program Details	10
7.1.6 Publish and View Customers on a Travel Program	10
7.1.7 View Basic Information about the Travel Agency(About us Page)	11
7.1.8 Modify Personal Profile	11
7.1.9 Weather and flight inquiry services	11
7.1.10 Reserve Travel Programs	12
7.2 Functional Requirement Implementation - Staff Side	12
7.2.1 View Dashboard	12
7.2.2 Reservation and Travel Programs Table Management	12
7.2.3 Edit and Send Notification to Customers	12
7.2.4 Upload New Travel Programs	13
7.2.5 Epidemic Management	14
7.3 Functional Requirement Implementation - Customer and Staff Side	14
7.3.1 Login and Register	14
7.3.2 Track Reservation Status	15
7.3.3 Modify Reservation Status	15
7.3.4 AI Customer Service	16
7.3.5 Chat Between Customer and Staff	17
7.3.6 Chinese-English switching	18
7.4 Non-Functional Requirement Implementation	18
8 Process	19
9 Conclusion	19

1 Abstract

This project involves the development of a travel agency website with a focus on providing efficient and user-friendly features for both staff and customers. The implementation of functional requirements covers various aspects such as dashboard views, reservation management, notification editing and sending, travel program uploading, epidemic management, login and registration, AI customer service, chat functionality, and language switching. Additionally, non-functional requirements such as performance, security, maintainability, and usability were prioritized. The development process followed an agile methodology with separate front and back-end approaches, emphasizing teamwork and collaboration. The project has provided valuable experiences and insights, enhancing the team's skills in software engineering and preparing them for future professional development in the field.

2 Overall

2.1 introduction

Tabibito is a unique travel website offering a wide range of domestic and international tours. With detailed itineraries, cost information, and visual aids like pictures and maps, users can easily plan their trips. The website features a smart chat function for live assistance and personalized customer service. Users can also contribute reviews and ratings to help fellow travellers. Our website is committed to providing affordable tours without compromising on safety and quality. Their goal is to make travel comfortable and memorable for users by offering a comprehensive one-stop travel service.

2.2 Technologies

To implement the website to fulfil the given requirements, the following technologies are used in our project:

Languages We have used the following languages:

- Python
- HTML
- CSS
- JavaScript

Basic Technologies

- **Vue.js:** We have incorporated Vue.js into our project to improve the appearance of our website. Vue.js is a flexible JavaScript framework that enables the use of beautiful and powerful external component libraries by breaking web pages into reusable components.
- **Flask:** Flask is a lightweight Web Server Gateway Interface (WSGI) web application framework with minimal dependencies on external libraries. Furthermore, it is based on Python. Thus, we can develop flexibly and conveniently to reduce time and human resource costs.
- **MySQL:** MySQL is a free, open-source relational database management system (RDBMS) that uses SQL to manage data inside a database. By using MySQL, we can protect sensitive data from intruders since MySQL has a solid data security layer.

External Libraries

- **Naive UI:** A fairly complete and theme-customizable Vue 3 Component Library. [2]
- **i18n:** An internationalization framework written in and for JavaScript. [9]

- **Tencent Instant Messaging:** A matured Instant Messaging Solution provided by Tencent. [11]
- **ChatGPT API:** A powerful tool that provides developers access to OpenAI's advanced Large-scale Language Model, ChatGPT. [1]
- **Google Map JavaScript API:** A JavaScript API provided by Google to integrate Google Map and Google Street View into your website. [7]
- **Echarts:** An open-source JavaScript visualization library that provides a variety of chart types and components that can be combined to create interactive data visualizations. [8]
- **FlightAware Aero API:** An API System provides access to flight information of places all over the world. [6]
- **Weather API:** An API System provides access to weather information of places all over the world. [12]

3 Teamwork

3.1 Introduction

Teamwork is a critical aspect of Tabibito's success. As a travel website that aims to provide quality, affordable tours while ensuring the safety and quality of all the tours offered on its website, Tabibito relies heavily on teamwork to achieve its mission. Our team of six, consisting of Donglin Liu, Jiahe Zhang, Wenbei Xie, Haoran Yan, Wenjie Wu, and Feihe Huang, is committed to working collaboratively to contribute to the company's success.

3.2 Team Structure

Tabibito's teams are organized in a way that allows for cross-functional collaboration and efficient project completion. Our team members are selected for specific tasks based on their skills and expertise. Donglin Liu and Jiahe Zhang are responsible for developing new tours, while Wenbei Xie and Haoran Yan handle the website's technical aspects. Wenjie Wu is in charge of marketing and promotions, and Feihe Huang manages customer service. By working together, we ensure that each aspect of Tabibito's business is handled efficiently and effectively.

Name	Role	Responsibilities	Contribution
Donglin Liu	Technique manager, Front-end developer	Participate in front-end development, provide technical support, lead mobile adaptation, and participate in document writing.	16.67%
Feihe Huang	Resource manager, Front-end developer	Participate in front-end development, coordinate internal resources and participate in document writing.	16.67%
HaoRan Yan	Maintain manager, Back-end developer	Participate in back-end development, lead project maintenance and error analysis and correction.	16.67%
Jiahe Zhang	Project manager, Front-end developer	Participate in front-end development, ensure the alignment of project progress and communication between stakeholders and developers.	16.67%
Wenjie Wu	Testing manager, Back-end developer	Participate in back-end development, lead code testing, and conduct error analysis reports.	16.67%
Wenbei Xie	Group leader, Front-end developer	Coordinate and plan project progress, supervise project development, participate in front-end development, and participate in document writing.	16.67%
Dr Catherine Mooney and Dr Brett Becker	Project sponsor	Formulate the overall vision for the project, make crucial decisions, approve the budget, ensure resource availability, and resolve conflicts.	

3.3 RACI Matrix: individual Responsibilities

The table below shows our RACI chart [3], which outlines the roles of each team member. RACI is an acronym for Responsible, Accountable, Consulted, and Informed.

RACI Chart	Team Member						
Activity	Donglin Liu	Feihe Huang	HaoRan Yan	Jiahe Zhang	Wenjie Wu	Wenbei Xie	Han Weng(TA)
Requirements analysis	R	C	C	R	I	C	A
Develop project proposal	C	R	I	R	R	C	A
Project implementation	R	R	R	R	R	R	A
Submit change request	C	I	R	I	C	I	A
Project testing	I	R	C	C	R	R	A
Write documentation	R	R	R	R	R	R	A

4 Plan

4.1 Milestones

Our development plan consists of six milestones, each aligned with the creation of specific functional modules for our project. This section provides a comprehensive explanation of each milestone, including its characteristics and requirements for every module. Additionally, we outline the final deliverables for each module.

4.1.1 Milestone 1: Chat Module(2023.5.1~2023.5.8)

- Description: This module provides a communication platform between users and employees also between users and users, as well as communication between users and GPT-4.0 AI customer service.
- Deliverables:
 - Basic chat functionality between users and employees also between users and users.
 - Integration of GPT-4.0 AI customer service.
 - Design and implementation of chat UI.

4.1.2 Milestone 2: Profile Module (2023.4.17~2023.5.1)

- Description: This module includes personal profile management and the ability to manage your orders.
- Deliverables:
 - User registration, login, and logout functions.
 - Profile creation and editing functionality.
 - Design and implementation of profile UI.
 - Email verification code login verification.

4.1.3 Milestone 3: Travel Project Information Module (2023.3.26~2023.4.10)

- Description: This module includes travel-related functions such as browsing and searching for travel projects, viewing travel project details, and commenting on travel projects.
- Deliverables:
 - Travel project browsing and searching functions.

- Travel project detail view function, including using Google Map help customer get more detail about the destination.
- The comment function on travel projects includes the analysis of scores and then use algorithms to recommend more suitable travel items to users
- Design and implementation of travel project UI.

4.1.4 Milestone 4: Travel Project Management Module (2023.3.13~2023.3.27)

- Description: This module includes functions related to managing travel projects, such as uploading new projects, editing existing projects, and deleting projects.
- Deliverables:
 - Travel project uploading function.
 - Travel project editing and deletion functions.
 - Design and implementation of travel project management UI.

4.1.5 Milestone 5: Reservation Management Module (2023.4.3~2023.4.17)

- Description: This module includes functions related to managing reservations, such as creating, editing, and cancelling reservations.
- Deliverables:
 - Reservation creation and editing functions.
 - Reservation cancellation function.
 - Staff side management of reservation functions.
 - Design and implementation of reservation management UI.

4.1.6 Milestone 6: Portal Module (2023.3.6~2023.5.22)

- Description: This module includes functions related to employee management, such as order control, travel project control, and website data control.
- Deliverables:
 - Order control function.
 - Travel project control function.
 - Website data viewing functions.
 - Design and implementation of portal UI.

4.2 Timelines

5 Requirements

5.1 Functional Requirement

- **Customer Portal:** Allow customers to browse travel program details, publish and view comments, modify personal profiles, reserve travel program, communicate with staff and modify reservation information.
- **Staff portal:** Allows staff to manage the reservation and travel program information, upload new travel programs, send announcements to customers and answer customers' questions.
- **Innovative functions:** Include maps and street views, AI chat service, data visualizations diagrams, weather and flight inquiry service, travel video introduction and efficient program recommendation.



Figure 1: Gantt Chart

5.2 Non-Functional Requirement

- Performance: Our system must be able to handle heavy loads and provide fast response times to users. It must be optimized to ensure that it performs well under heavy load.
 - Security: Our system must be secure and protect user data. It must implement various security measures such as encryption and access controls to ensure that user data is protected.
 - Maintainability: Our system must be easy to maintain. It must be designed using modular design principles and be well-documented to make it easy for developers to understand and modify the code.
 - Usability: Our system must be easy to use. It must have a user-friendly interface and provide helpful resources to assist users in using the system.

5.3 Use case Diagram

There are two roles: customers and staff in our system. They have different permission based on their role, but they also have some common functions. The following figure 2 gives an overview of our project's use cases.

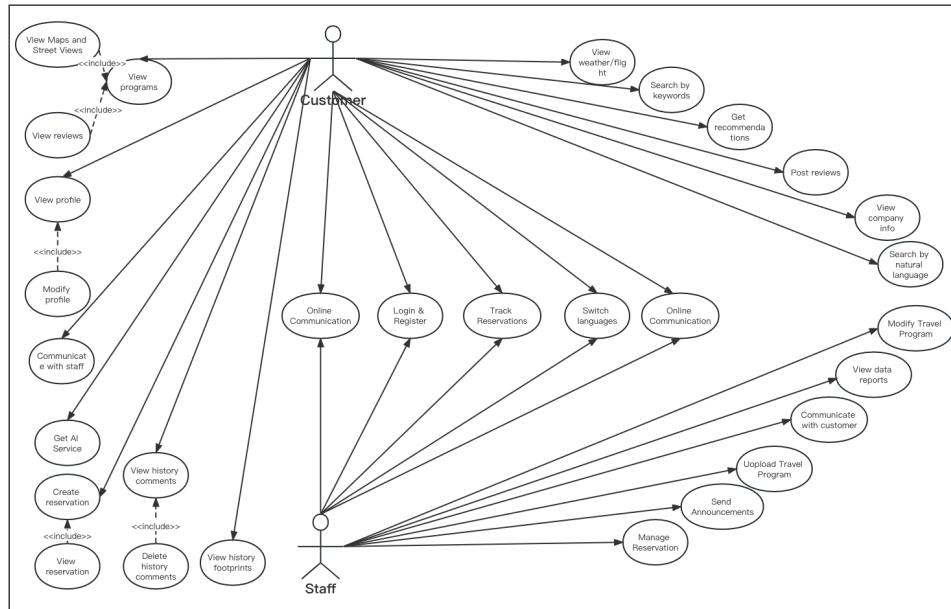


Figure 2: Use Case Diagram

6 System Architecture

6.1 Front-end

The following package diagram (Figure 3) gives an overview of the front-end system architecture:

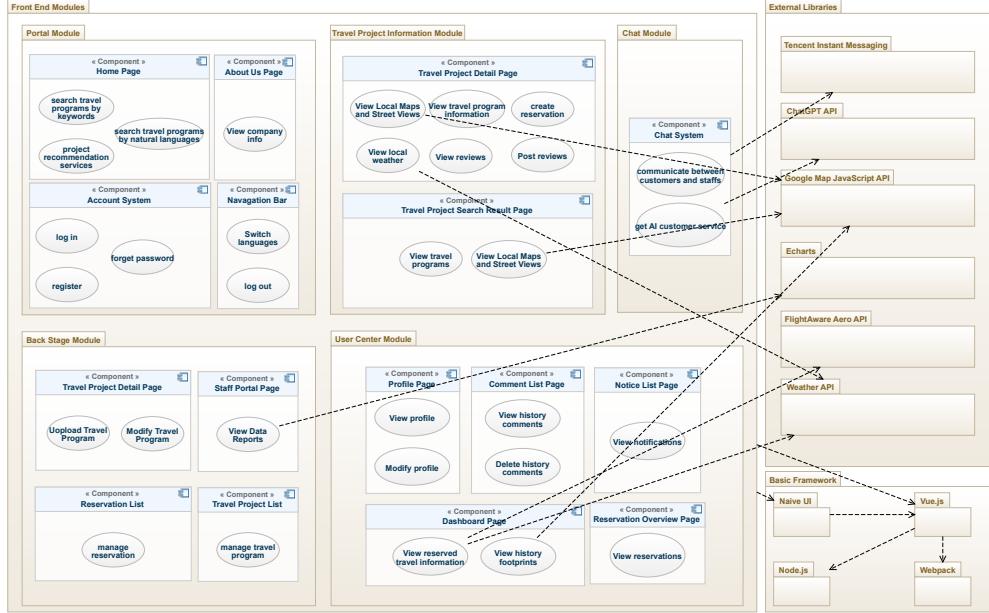


Figure 3: Front-end Package Diagram

6.2 Back-end

Figure 4 gives an overview of the back-end system architecture:

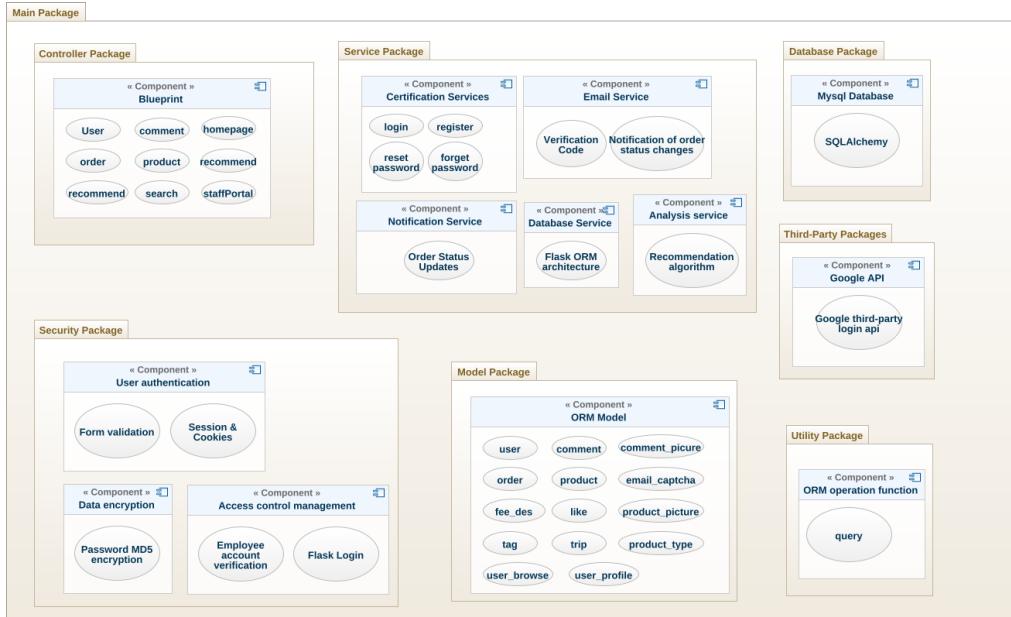


Figure 4: Back-end Package Diagram

6.3 Database

Picture 5 shows the database class diagram

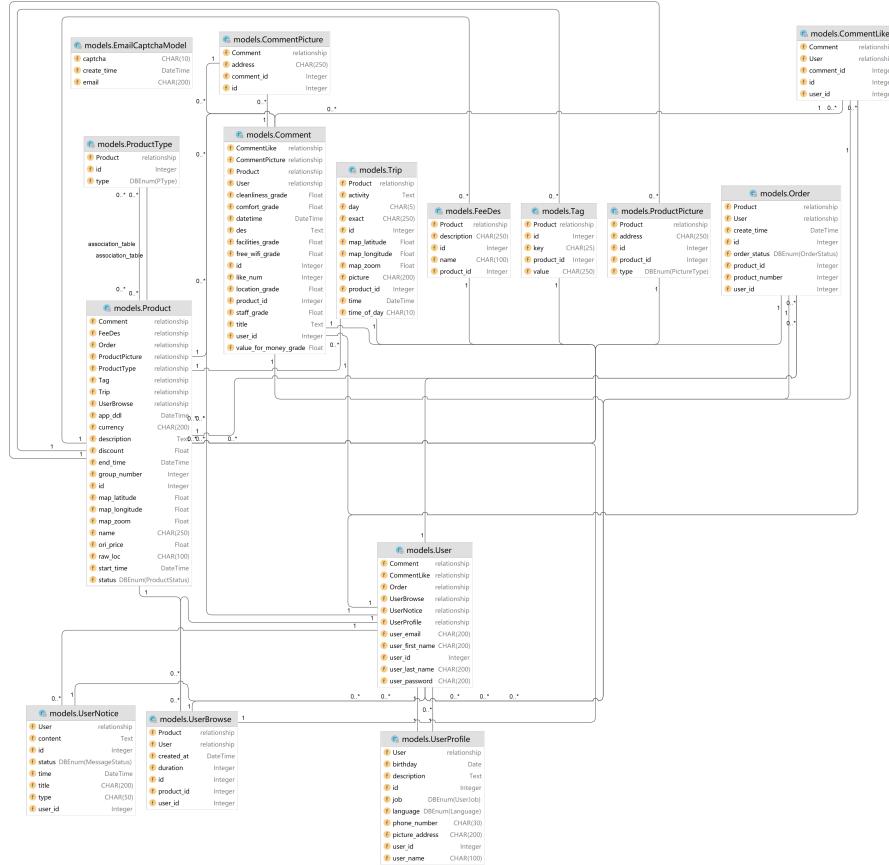


Figure 5: Database Diagram

7 Design & Solution

7.1 Functional Requirement Implementation - Customer Side

7.1.1 Maps and Street Views

User story : As a traveller, I want to see a map that can show all the sites in a travel program so that I can know more view about the program.

Design & Solution : The solution of this function is divided into the following parts.

- **Front-end Implementation:** For every travel program, a Google Map [7] will be provided to the customer to have an overview of the sites in the travel program.
- **Back-end Implementation:** When a customer opens a travel program detail page, the front end will send a request for the mark and map information to the back end. The front end can use the information to require the Google Map API to get the image of the map.

Screen Shot : Please see figure 6 below:

itinerary

- 1 Central Park
Day 1, Morning Sun, 24 Apr 2022 03:20:00 GMT
[See details & photo](#)
- 2 Statue of Liberty National Monument
Day 1, Afternoon Sun, 24 Apr 2022 12:20:00 GMT
[See details & photo](#)
- 3 Empire State Building
Day 2, Morning Mon, 25 Apr 2022 08:20:00 GMT
[See details & photo](#)
- 4 Brooklyn Bridge
Day 2, Afternoon Mon, 25 Apr 2022 16:20:00 GMT
[See details & photo](#)
- 5 Times Square
Day 2, Night Mon, 25 Apr 2022 22:20:00 GMT
[See details & photo](#)

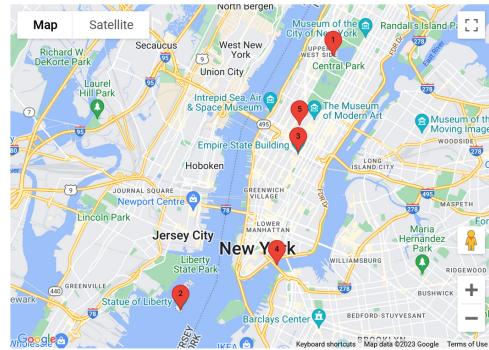


Figure 6: Edit and Send Notification to Customers

7.1.2 Data Visualizations: Footprint Wall, Heat Maps, and Charts

- User story:** As a customer, I want to be able to view my travel records and information about tourist attractions in a visually appealing and user-friendly manner so that I can relive my beautiful travel memories better.
- Design & Solution:** Our project integrates e-charts' data visualization functions to optimize the customer experience. We have added a footprint wall that uses e-charts to visually display the user's travel records. For tourist attractions, we provide heat maps based on the frequency of tourist visits, derived from our website's order information. These heat maps are also generated using e-charts, which is an open-source JavaScript visualization library that provides a variety of flexible chart types and powerful rendering capabilities [8]. This data is retrieved by the back end and passed to the front end through APIs.

The heat map displays the frequency of tourist visits to various attractions, derived from our website's order information as below.

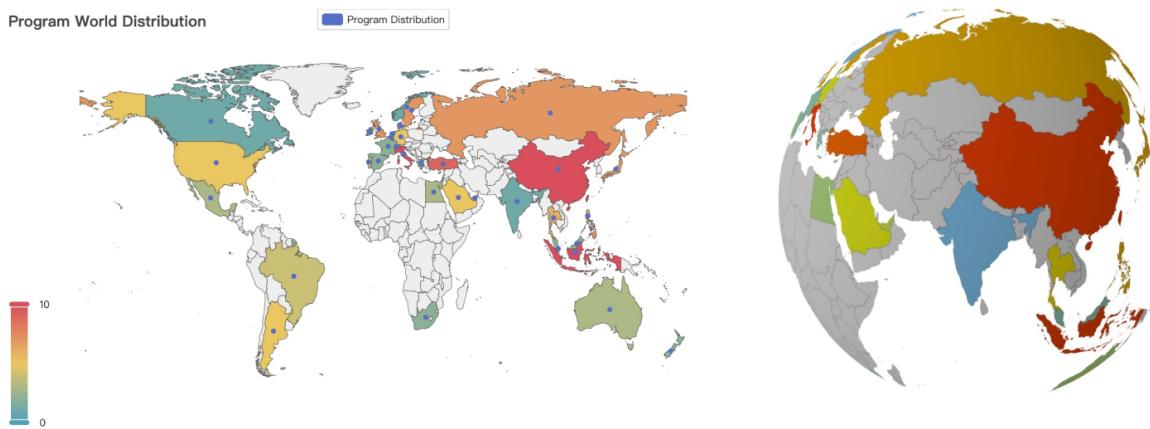


Figure 7: Heat map of tourist attractions generated using e-charts

7.1.3 Efficient and accurate project recommendation services based on auxiliary tools

- User story:** As a user, I want a travel recommendation service that considers my interests and preferences, providing me with relevant suggestions so that I can enhance my travel itinerary planning.
- Design & Solution:**
 - Scope:
The service will recommend travel products based on user preferences such as travel destination, and tags, as well as product attributes such as location and amenities.

- Technical Implementation:

Collaborative Filtering Algorithm: The system will use both user-based and content-based collaborative filtering algorithms to recommend travel products to users. The user-based algorithm will compare the preferences of the user with those of other similar users to generate recommendations. The content-based algorithm will compare the attributes of the products with those of the products the user has already shown interest in to generate recommendations. [10] Natural Language Processing: The system will use natural language processing tools to analyze user reviews and product descriptions to better understand user preferences and product attributes. [4]

```
1 def find_similar_products(browses, products):
2     similar_products = set()
3     for browse in browses:
4         for product in products:
5             # with same type or similar location
6             if set_similar(set(product.types), set(browse.product.types)) or \
7                 product.raw_loc == browse.product.raw_loc or \
8                 get_similarity(product.name, browse.product.name) >= 0.8:
9                 similar_products.add(product)
10            # with same tags
11            for tag in product.tags:
12                for browse_tag in browse.product.tags:
13                    if tag.key == browse_tag.key and \
14                        get_similarity(tag.value, browse_tag.value) >= 0.8:
15                        similar_products.add(product)
16                break
17    return similar_products
```

7.1.4 Diversified display of program information

- **User story:** As a travel customer, I want to view detailed information about a travel product in different ways, so that I can make informed decisions even though I cannot understand the description words.
- **Design & Solution:**
Except clear description in words, media information such as attractive photos or videos are included in the program-detail page. The page layout is easy to navigate and find so that users can quickly find the information they are looking for.

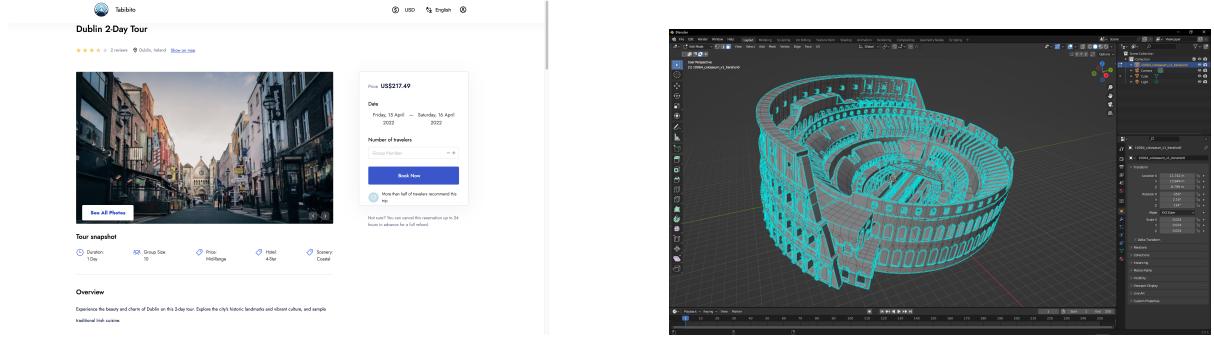
7.1.5 Browse Travel Program Details

- **User story:** As a customer, I would like to learn detailed travel destination information like the descriptions, price, pictures, travel Itinerary and date on a webpage so that I can make better decisions on these details.
- **Design & Solution:** We create a webpage for each travel destination that displays detailed information, such as descriptions, prices, pictures, travel itineraries, and dates to enable customers to learn detailed travel destination information. This enables customers to make decisions when choosing travel destinations and gives them a better understanding of the travel before booking. **We also include 3D model demonstrations made with Blender on each travel destination webpage, allowing customers to get a more intuitive understanding of the destination's appearance.**

Screen Shot : Figure 8 below shows a travel program details page:

7.1.6 Publish and View Customers on a Travel Program

- **User story:** As a customer, I would like to publish some comments on a travel program so that I can share my feelings on a trip.



(a) Travel Program Details Page

(b) Blender Model

Figure 8: Two Images Side by Side

- **Design & Solution:** We create a webpage for each travel program that allows customers to add comments by filling out a form that includes a rating, title, and description.

Customers can scroll through the comments section on the travel program webpage. We use the Vue-vite framework to ensure fast rendering and responsiveness.

7.1.7 View Basic Information about the Travel Agency(About us Page)

- **User story:** As a customer, I want to learn about the background and team information of the website so that I can assess its reliability and credibility.

- **Design & Solution:**

Service Introduction:

Provide a brief overview of the services offered by the company. Highlight the key features of the services and how they benefit the users: Company Introduction, Team Introduction, and User Feedback.

Page Design: Use a clean and modern design that is visually appealing. Choose a colour scheme that aligns with the company's branding. Use easy-to-read fonts and clear headings. Ensure the page is mobile-friendly and responsive.

7.1.8 Modify Personal Profile

- **User story:** As a customer, I want to modify my personal profile information, so that my account information is accurate and up-to-date.

- **Design & Solution:**

Display the user's current personal profile information, such as username, profile picture, etc. for the user to review. Provide a form and an "Edit" button, a "Save" button, and a "Cancel" button on the user's personal profile page to allow the user to modify their personal information.

7.1.9 Weather and flight inquiry services

- **User Story:** As a customer, I want to get information on the flight and weather of the place I want to go so that I can have better preparation.

- **Design & Solution:** To use the real data of the current flight, we use an external library named **Amadeus travel API [6]**. It can return the price of a flight on particular routes.

To use the real data of the current weather of one location, we also use an external library named **Free Weather API [12]**. It can return to the current weather conditions of particular places.

7.1.10 Reserve Travel Programs

- **User story:** As a customer, I want to make a reservation on the travel program so that I can tell the merchant that a booked the travel program.
- **Design & Solution:** After the customer presses the book button on any program detail page, a requirement will be sent to the back end to be stored in MySQL.
- **Sequence Diagram:** The sequence diagram of this use case is shown below (Figure 9):

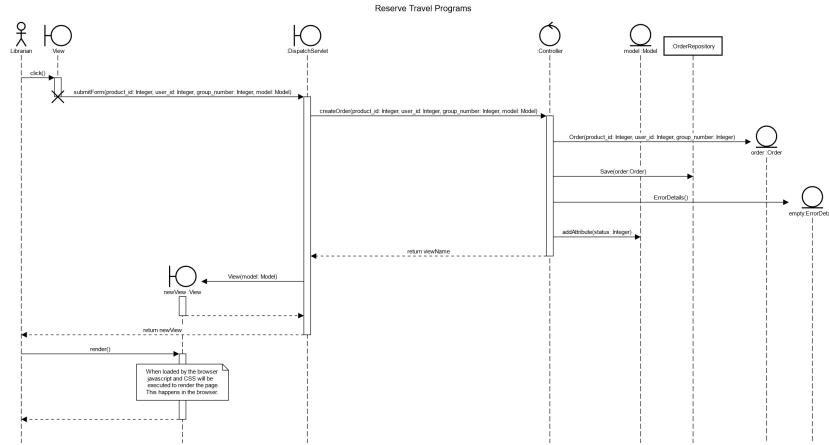


Figure 9: Sequence Diagram for "Reserve Travel Programs"

7.2 Functional Requirement Implementation - Staff Side

7.2.1 View Dashboard

- **User story:** As a staff, I would like to view the website's revenue situation and latest booking orders in a more intuitive way so that I can better understand the website's performance and manage operations.
- **Design & Solution:** Our project provides a View Dashboard page for employees. On this page, employees can view data visualization charts based on simple data displays to see the website's revenue situation in a more intuitive way. Employees can also see some of the latest booking orders on this page. This page allows employees to further go to the reservation list page, blacklist page, and other pages for targeted management operations.

7.2.2 Reservation and Travel Programs Table Management

- **User story:** As a staff, I want to be able to view, modify, and delete information related to orders and travel projects in an efficient and user-friendly manner so that I can improve my work efficiency.
- **Design & Solution:** Our project employs the MVC architecture to handle reservations and travel programs in the employee portal. The front end, based on the vue-vite framework, provides a user-friendly interface for employees to view, modify, and delete information related to orders and travel projects. The front end interacts with the back end through RESTful APIs to pass data and requests. When modifying a reservation, employees can update the order status in real-time based on factors such as payment and travel progress. This design allows for efficient and user-friendly management of reservations and travel programs by staff.
- **Sequence Diagram:** The sequence diagram illustrates the process of an employee removing a reservation using the MVC architecture. See figure.

7.2.3 Edit and Send Notification to Customers

User story : As a staff, I want to edit and send a notification to customers whenever I need so that I can save time on communication with customers.

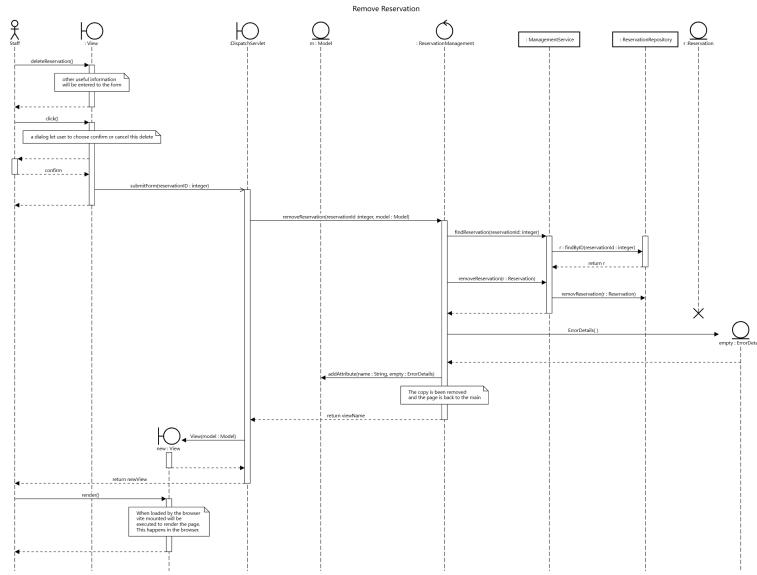


Figure 10: Sequence diagram for removing a reservation

Design & Solution : The solution of this function is divided into the following parts.

- **Notification Structure:** We have given a basic design for the notification. A notification consists of a title, type, status and content.
 - **Front-end Implementation:** When a staff submit the modification of a specific travel program, a page is provided for the staff to edit the notification for the user. After pressing the submit button to modify the travel program, a notification will be sent to the customers who reserved this travel program.
 - **Back-end Implementation:** When a staff submit the modification of a specific travel program, the former types will be required by API. After receiving the requirement, the controller will return all the types reserved in the database.

Screen Shot : Please see figure 11 below:

1. Basic Information 2. Location 3. Route 4. Price 5. Notification 6. Submit

Title

Tags

Time Change X

New Tag

Add

Content

Content

Figure 11: Edit and Send Notification to Customers

7.2.4 Upload New Travel Programs

- **User story:** As a travel agency staff, I want to upload travel destination information like the name, pictures, location, and price on the website for customers to book so that I can save time on the filling application form.

- **Design & Solution:** We create a form for staff to input the necessary information, such as the name, pictures, location, and price of the travel destination to implement the function for travel agency staff to upload travel program information. The form will be accessible via a staff portal, which will require staff authentication to access. Once the staff member fills out the form, they will submit it, and the information will be stored in a MySQL database.

7.2.5 Epidemic Management

- **User story:** As a travel agency staff, I want to manage the reservation statutes and announce them to the customer so that I can deal with the pandemic and protect the customer's safety.
- **Design & Solution:** We create a reservation management system using Flask and MySQL. This system will allow staff to update reservation statuses for each customer, including confirming, rescheduling, or cancelling reservations. The staff will track the epidemic information and manage the reservation information. To keep customers informed of reservation updates information, we also implement an automated messaging system. The customer will receive announcements once the staff changes the reservation status.

7.3 Functional Requirement Implementation - Customer and Staff Side

7.3.1 Login and Register

- **User story:** As a customer, I want to access personalized information and services so that I can protect my information with my account.

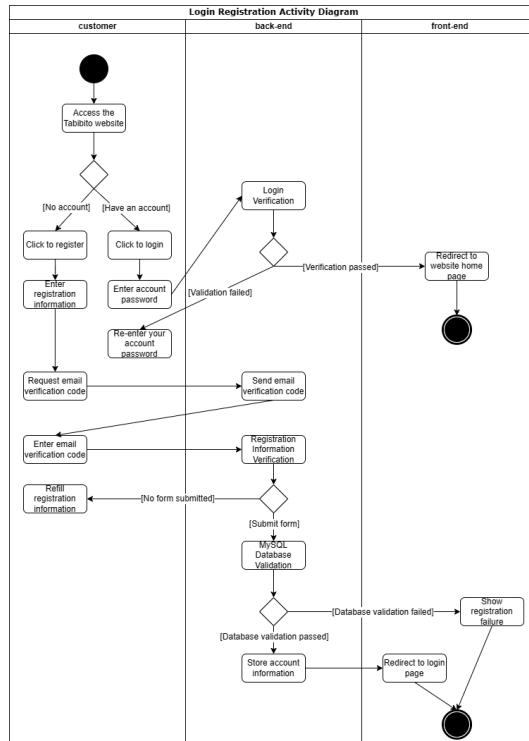


Figure 12: Activity diagram

- **Design & Solution:** On the Tabibito website login page, users are presented with two options: 'Sign in with Google' and 'Sign up for a new account for free'.

Upon clicking on the Sign-up for free account option, the user lands on a page where they have to put in their name and other necessary information as well as create an account password, and by clicking on the registration form's email verification code get button, an email is sent to users containing a unique validation passcode. In order for them to confirm their email and successfully register for an account, they will have to insert this code into a specific field, but upon choosing the Sign in with Google option users are taken

to a dedicated sign-in page where they can fill in their login details. For the "Register a new account for free" option, the system sends a unique verification code to validate the registered user's mail ID in order to complete account creation, so the ultimate step towards finishing up with registering is entering this confirmation code which was sent by us onto our pre-registration email. Opting for Sign in with Google will lead users directly to a page prompting them for their email address and password needed for signing in to our services, and upon successful authentication by Google, users will be directed to the travel agency's website and logged into their accounts automatically. Creating a user profile for each case ensures that personal data is kept safe within our database while enabling users to retrieve tailored info and utilize our website's service.

- **code:** Here is part of the code for using Google third-party login with oauth2 authentication.

```

1 def authorize():
2     flow = google_auth_oauthlib.flow.Flow.from_client_secrets_file(
3         CLIENT_SECRETS_FILE, scopes=SCOPES)
4     flow.redirect_uri = flask.url_for('oauth2callback', _external=True)
5     authorization_url, state = flow.authorization_url(
6         access_type='offline', include_granted_scopes='true')
7     flask.session['state'] = state
8     return flask.redirect(authorization_url)

```

- **Screen Shot :** Please see the figure below:

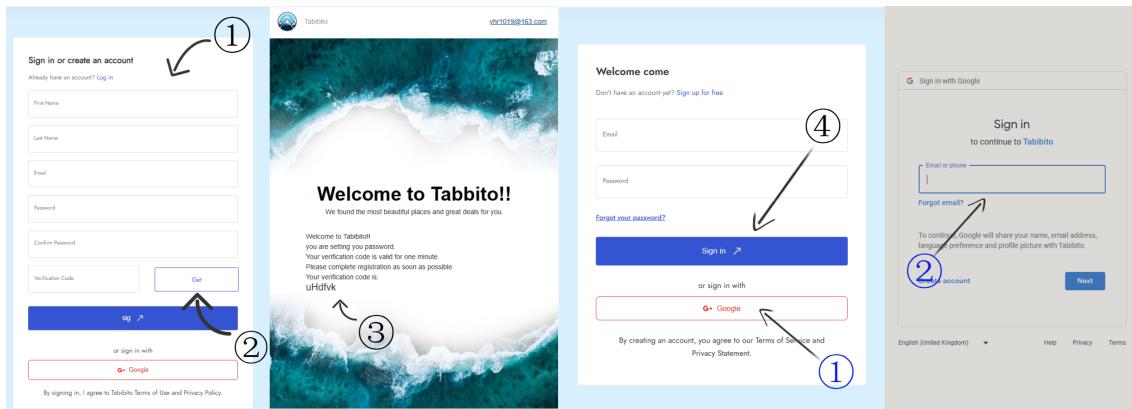


Figure 13: Login and registration page

7.3.2 Track Reservation Status

- **User story:** As a customer or staff member, I want to be able to track the status of my reservation (or all reservations, for employees), so that I can be aware of any changes or updates.
- **Design & Solution:** When logging into the Tabbito website with computer browsers or mobile browsers, customers will be able to track and find out information about any current bookings they may have. Employees are allowed to see the status of all reservations. On the personal information page, customers can keep track of reservation statuses. The status of any reservation can be checked by both the reservation maker and employees.
- **Screen Shot :** Please see the figure below:

7.3.3 Modify Reservation Status

- **User story:** As a customer, I want to modify reservation statuses including cancelling reservations and having better control over my travel plans so that I can manage the reservation efficiently.

Recent Bookings						View All
#	Item	Total	Paid	Status	Created At	
#1	New York City, New York	\$400	\$0.2	Completed	2023-04-04 19:52:50	
#2	New York City, New York	\$600	\$0.2	Cancelled	2023-01-26 19:53:17	
#3	Beijing, China	\$960	\$0.8	Completed	2023-01-16 19:54:40	
#4	Tokyo, Japan	\$5625	\$0.75	Cancelled	2023-04-16 16:42:36	
#5	New York City, New York	\$1200	\$0.2	Completed	2023-04-16 16:57:03	
#6	Xi'an, Shaanxi Province, China	\$2700	\$0.75	Cancelled	2023-04-16 16:57:14	

Figure 14: Staff-side tracking order status page

- **Design & Solution:** A complete list of orders can be found on the staff management's order page in tabular form. The system provides buttons that permit the completion of orders along with status options. Logging into their account as an employee directs them to the staff management orders page, whereby means of this platform they can get a full overview of all existing orders as well as their statuses. The worker is able to modify or cancel an order by selecting the corresponding button. Only the staff members have authorization for accessing and managing such listed items.
- **Screen Shot :** Please see the figure below:

The screenshot shows the 'Reservation List' section of the Tabibito staff interface. At the top, there are buttons for 'Back To Dashboard' and 'View all Programs'. Below this is a table with columns: Tourism Title, Start Time, End Time, Reservation Time, Price, Discount, Booking Holder, Status, Success, Cancel, Delete, and Edit. The table lists six tourism programs with their respective details and status buttons. Navigation arrows at the bottom right indicate the list spans multiple pages.

Tourism Title	Start Time	End Time	Reservation Time	Price	Discount	Booking Holder	Status	Success	Cancel	Delete	Edit
Weekend Getaway in New York City	2022-07-26	2022-07-28	2023-04-04	400	0.2	Wenjie Wu	Completed	<button>Complete</button>	<button>Cancel</button>	<button>Delete</button>	<button>Edit</button>
Weekend Getaway in New York City	2022-07-26	2022-07-28	2023-01-26	600	0.2	Wenjie Wu	Cancelled	<button>Complete</button>	<button>Cancel</button>	<button>Delete</button>	<button>Edit</button>
Discovering Beijing's Hidden Gems	2022-07-31	2022-08-02	2023-01-16	960	0.8	Wenjie Wu	Completed	<button>Complete</button>	<button>Cancel</button>	<button>Delete</button>	<button>Edit</button>
Explore Japan	2023-05-21	2023-05-28	2023-04-16	5625	0.75	Wenjie Wu	Cancelled	<button>Complete</button>	<button>Cancel</button>	<button>Delete</button>	<button>Edit</button>
Big Apple	2022-07-26	2022-07-28	2023-04-16	1200	0.2	Wenjie Wu	Completed	<button>Complete</button>	<button>Cancel</button>	<button>Delete</button>	<button>Edit</button>
Discover the Ancient Wonders of Xi'an: 10-Day Adventure	2022-07-18	2022-07-26	2023-04-16	2700	0.75	Wenjie Wu	Cancelled	<button>Complete</button>	<button>Cancel</button>	<button>Delete</button>	<button>Edit</button>

Figure 15: Staff-side order status change page

7.3.4 AI Customer Service

- **User story:** As a customer or staff, I want to quickly find information and get my questions answered without having to search through multiple pages or menus so that I can quickly get the information about the website I need.
- **Design & Solution:** Our project integrates OpenAI's GPT-4.0 [1] chat service to help users search for information and improve their user experience. GPT-4.0 is a natural language processing (NLP) tool that can understand and generate human-like responses to questions and queries [5]. By integrating GPT-4.0 into our website, we can offer customers a chatbot that can quickly and accurately answer their questions and provide them with relevant information.

To implement this feature, we will create a chatbot interface on the website that will allow customers to ask questions and receive responses from the GPT-4.0 chat service. When a customer asks a question,

the chatbot will send the query to GPT-4.0, which will process the question and generate a response. The response will then be sent back to the customer via the chatbot interface.

To ensure that the chatbot is user-friendly and easy to use, we will design it to have a simple and intuitive interface. The chatbot interface will be prominently displayed on the website and accessible from all pages. Customers will be able to type in their questions using natural language, and the chatbot will respond with clear and concise answers.

In addition to answering customer questions, the GPT-4.0 chat service can also be used to provide personalized recommendations to customers based on their preferences and browsing history. This feature will help customers find the information they need quickly and easily, without having to search through multiple pages or menus.

To ensure that the chatbot is reliable and accurate, we will test it extensively before launching it on the website. We will also monitor customer feedback and make adjustments as necessary to improve the chatbot's performance.

By integrating OpenAI's GPT-4.0 chat service into our website, we can provide customers with an efficient and convenient way to find the information they need and improve their overall user experience.

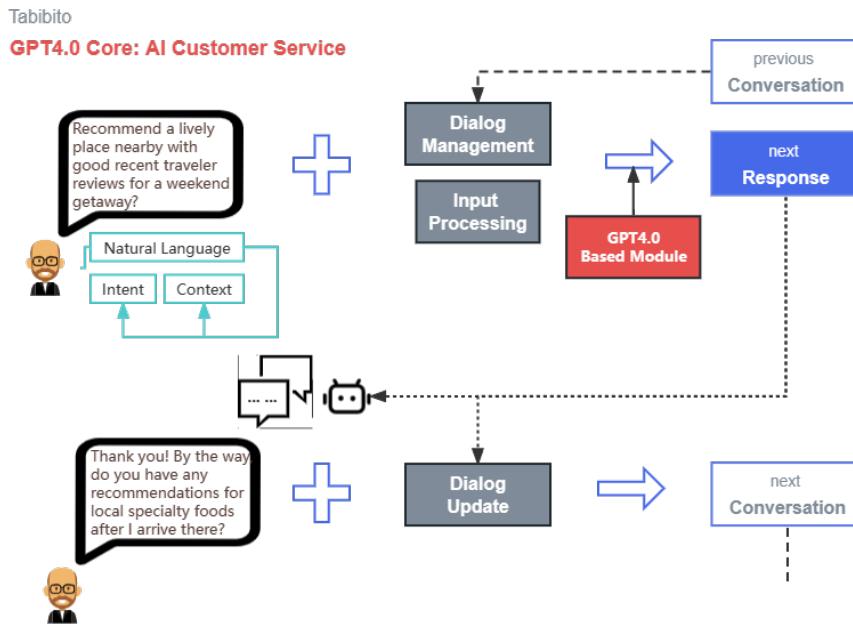


Figure 16: Pipeline of Our GPT-4 Customer Service

7.3.5 Chat Between Customer and Staff

- **User story:** As a customer or staff, I would like to have an immediate chat online with the staff so that I can learn more details about the travel information.
- **Design & Solution:** Customers can use the online chat system to contact staff for travel information. To implement the immediate chat feature, we integrate **Tencent Cloud's instant messaging service (IMS)** [11]. Once the customer has logged in, they can access the chat interface at any time, allowing them to send text and images to the staffer. Once the user has entered a message and clicked send, the message will be sent instantly to the staff via IMS and the staff will immediately receive the message and reply to the customer's question. The chat function provides a more personalized and interactive experience for customers, as they can get immediate responses to their queries and concerns. It also enables staff members to manage customer inquiries more efficiently and provide better customer support.
- **Swimlane diagram:** Figure 17 shows the swimlane diagram of the chat function

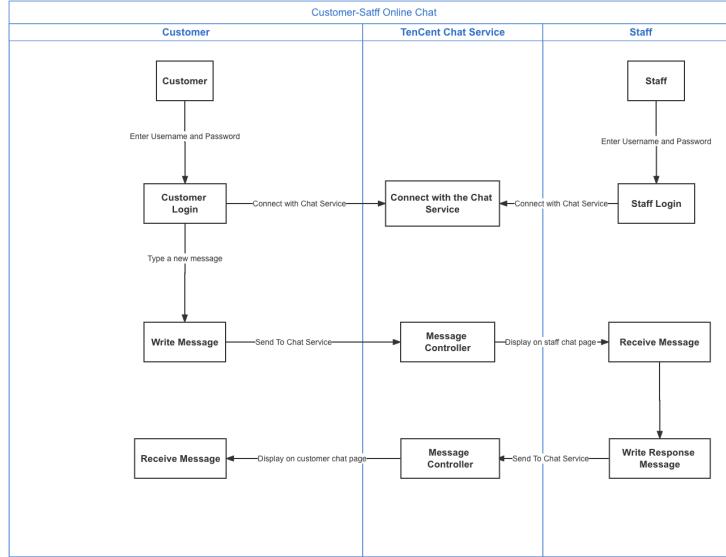


Figure 17: Chat Swimlane diagram

7.3.6 Chinese-English switching

- **User story:** As a customer or staff, I want to view the website in the language that I am familiar with so that I can easier to use the website.
- **Design & Solution:** Our project use **vue-i18n** [9], the abbreviation of vue-internationalization, which is an external library of Vue.js. This library makes our website adaptable to different languages without preparing different HTML templates. Customers and staff can just change the language of the website by clicking the language button on the navigation bar.

To use this library, there are two processes to do. First, create two **.json** files to store the corpus, each one for a specific language. For example, the following code shows the structure of the **.json** file about the navigation bar.

```

1 navi: {
2     lang: "English",
3     USD: "USD",
4     CNY: "CNY",
5     EUR: "EUR",
6     profile: "Profile",
7 },

```

Then, replace the word in the HTML template with interpolation to use corpus. For example, the code following realizes the part of the language switch in the navigation bar.

```

1 <div class="actionButton languageButton" click="changeLang()">
2     <div class="actionIcon"></div>
3     <div class="actionText">{{ $t('navi.lang') }}</div>
4 </div>

```

7.4 Non-Functional Requirement Implementation

- **Performance:** To ensure that our system performs well under heavy load, we have implemented various optimization techniques such as Vue lazy loading. We also realize the lazy loading of the translation corpus of our website.

- Security: Security is a top priority for our project. We have implemented various security measures such as password encryption, and authority management to ensure that our system and data are secure.
- Maintainability: Our system is designed to be easy to maintain. We have implemented modular design principles and have documented our code extensively to make it easy for developers to understand and modify the code. The component-based design also improves code utilization.
- Usability: We have conducted extensive user testing to ensure that our system is easy to use. We have also implemented a user-friendly interface and have provided user documentation to assist users in using our system.

8 Process

- Our development process uses an agile development methodology along with a separate front and back end approach to development. The popular Vue framework and the fast Vite packaging tool were used for the front end and the Python-based Flask framework was used for the back end to deliver the service. Although an agile development approach was used, requirements analysis, design, development, testing and deployment were carried out at various stages. We have broadly divided the entire development process into five phases: requirements specification, design and implementation, validation, evolution and deployment.
- For task planning and management, we used Microsoft's Worktile to plan and organise our work. To facilitate collaborative work between the front and back ends, we also used the APIfox tool to manage our interfaces.
- We use Git as a code version control system to facilitate us tracking changes to our source code and collaboration on code changes. We also use PyCharm and WebStorm, both integrated development tools with built-in code review tools, and use the GitLab platform to host our code for development team collaboration and code management.
- After each iteration, we conduct regular Retrospectives meetings to review the work and results within the past iteration. During this process, we discuss what has worked successfully and what areas need to be improved and optimised for better performance and efficiency in future work.

9 Conclusion

Throughout this project, our team of six individuals collaborated to successfully develop a travel agency website. This endeavor has brought us valuable experiences and insights, with a strong emphasis on teamwork. We have come to realize that effective teamwork is not just a concept, but a mindset and attitude crucial for achieving success.

Meanwhile, we applied the theories and fundamental principles of Software Engineering in a group project setting. We learned to design and create solutions based on problem specifications, utilizing appropriate development methodologies. These allowed us to contextualize our work within the profession of Software Engineering, effectively communicating our outputs through reports, presentations, and demonstrations to both technical and non-technical audiences.

Moreover, we enhanced our abilities in team communication, collaboration, problem-solving, creativity, work ethic, interpersonal skills, and time management. These soft skills are essential for thriving in the field of software engineering.

Furthermore, we engaged in self-directed professional development through research, both individually and as a team. We continuously sought opportunities to learn and grow, adapting to the rapidly evolving technologies and industry demands.

In summary, this project has enabled us to achieve the intended learning outcomes of the course design, and we have gained a wealth of knowledge. We deeply understand the importance of teamwork and have acquired core concepts and skills in software engineering through practical experiences. We believe that these experiences will have a positive impact on our future professional development, shaping us into exceptional software engineers.

References

- [1] Gpt-4 is openai's most advanced system, producing safer and more useful responses. <https://openai.com/product/gpt-4>, 2023.
- [2] 07akioni. Naive ui. <https://www.naiveui.com/en-US/os-theme>.
- [3] Forbes Advisor. Raci chart: Definitions, uses and examples for project managers, 2022.
- [4] Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with python. Online, 2009.
- [5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [6] flightaware.com. Aeroapi - flightaware. <https://flightaware.com/commercial/aeroapi/>.
- [7] Google. Map javascript api. <https://developers.google.com/maps/documentation/javascript?hl=en>, 2023.
- [8] Justineo. Vue echarts, 2021.
- [9] kazupon. vue-i18n. <https://github.com/kazupon/vue-i18n>.
- [10] Elaine Rich. User modeling via stereotypes. *Cognitive Science*, 3(4):329–354, 1979.
- [11] TenCentCloud. Tuikit. <https://cloud.tencent.com/document/product/269/68433>.
- [12] weatherapi.com. Free weather api. <https://www.weatherapi.com>.