

UNDERGRADUATE PROJECT REPORT

|  |  |
| --- | --- |
| **Project Title:** | **Web-based Campus Tourism Management System Design and Implementation** |
| **Surname:** | **Wei** |
| **First Name:** | **yuyang** |
| **Student Number:** | **202118010310** |
| **Supervisor Name:** | **James Blouin** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **May 6, 2025** |

# **Declaration**

**Student Conduct Regulations:**

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

<https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/>

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at

[https://www.brookes.ac.uk/students/sirt/student conduct/](https://www.brookes.ac.uk/students/sirt/student%20conduct/)

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them. Signature………Wei Yuyang……………… Date ……April 29th,2025……

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library. Signature…………Wei Yuyang…………… Date ……April 29th,2025……

# **Acknowledgment**

This paper is a summary of my knowledge accumulation and research practice during my undergraduate study, which is permeated with the teachings of teachers, the strength of classmates, the support of relatives and friends, and the selfless efforts of my family. Thanks to all those who helped.

This research was advised by Professor James Blouin from the initial conception to the final written text. In addition to undertaking heavy teaching and research tasks, Professor still systematically guided the whole process of topic selection, literature review, research design, technical implementation and paper writing. Through regular progress tracking, in-depth academic dialogue and defense simulation, the professor provided many professional and valuable opinions and work plans, specifically pointed out the modification part, and invested a lot of energy.

In addition, I would like to thank all the teachers who taught me during my study in the university. Through their careful instruction and patient reply, I have learned a lot and gained a lot. I would like to thank my classmates, friends and family who have been with me all the way through college. Their company became a strong support for me to achieve a bachelor's degree. With their support and companionship, I can find the balance between study and life. I would like to extend my sincerest respect to all those who have supported me.

# **Table of Contents**

[**Declaration** i](#_Toc196932013)

[**Acknowledgment** ii](#_Toc196932014)

[**Table of Contents** iii](#_Toc196932015)

[**Abstract** vi](#_Toc196932016)

[**Abbreviations** vii](#_Toc196932017)

[**Glossary** viii](#_Toc196932018)

[**Chapter 1 Introduction** 1](#_Toc196932019)

[**1.1** Background 1](#_Toc196932020)

[**1.2** Aim 1](#_Toc196932021)

[**1.3** Objectives 1](#_Toc196932022)

[**1.4** Project Overview 2](#_Toc196932023)

[**1.4.1** Scope 2](#_Toc196932024)

[**1.4.2** Audience 3](#_Toc196932025)

[**Chapter 2 Background Review** 4](#_Toc196932026)

[**2.1** Overview of Existing Smart Tourism Platforms 4](#_Toc196932027)

[**2.2** Competitive Analysis of Existing Tourism Management Systems 4](#_Toc196932028)

[**2.3** Research Methodology and Technical Architecture 6](#_Toc196932029)

[**Chapter 3 Methodology** 7](#_Toc196932030)

[**3.1** Approach 7](#_Toc196932031)

[**3.1.1** Development Model 7](#_Toc196932032)

[**3.1.2** Feasibility Analysis 8](#_Toc196932033)

[**3.2** Requirements Analysis 9](#_Toc196932034)

[**3.2.1** Function Analysis 10](#_Toc196932035)

[**3.2.2** Database Design 11](#_Toc196932036)

[**3.2.3** System Design 13](#_Toc196932037)

[**3.3** Technology 14](#_Toc196932038)

[**3.4** Project Version Management 14](#_Toc196932039)

[**Chapter 4 Implementation and Results** 16](#_Toc196932040)

[**4.1** Springboot Framework 16](#_Toc196932041)

[**4.2** Function Introduction 16](#_Toc196932042)

[**4.2.1** Home Page 16](#_Toc196932043)

[**4.2.2** Registration & Login 17](#_Toc196932044)

[**4.2.3** Travel Detail Page 18](#_Toc196932045)

[**4.2.4** Blog List Page 19](#_Toc196932046)

[**4.2.5** Blog Detail Page 19](#_Toc196932047)

[**4.2.6** Subscribe List Page 20](#_Toc196932048)

[**4.2.7** Message List Page 20](#_Toc196932049)

[**4.2.8** Notice List Page 21](#_Toc196932050)

[**4.2.9** Personal Center Page 21](#_Toc196932051)

[**4.2.10** User Chat Page 22](#_Toc196932052)

[**4.3** Function Test 22](#_Toc196932053)

[**4.4** Performance Test 29](#_Toc196932054)

[**4.4.1** Normal Load 32](#_Toc196932055)

[**4.4.2** Heavy Load 33](#_Toc196932056)

[**4.4.3** Scalability Testing 35](#_Toc196932057)

[**4.5** Unit Test 35](#_Toc196932058)

[**4.6** Product Deployment 38](#_Toc196932059)

[**Chapter 5 Professional Issues** 40](#_Toc196932060)

[**5.1** Project Management 40](#_Toc196932061)

[**5.1.1** Activities 40](#_Toc196932062)

[**5.1.2** Schedule 41](#_Toc196932063)

[**5.1.3** Project Data Management 41](#_Toc196932064)

[**5.1.4** Project Deliverables 42](#_Toc196932065)

[**5.2** Risk Analysis 42](#_Toc196932066)

[**5.2.1** Potential Risks 42](#_Toc196932067)

[**5.2.2** Mitigated Risks and Effective Strategies 45](#_Toc196932068)

[**5.2.3** Future Risk Considerations 46](#_Toc196932069)

[**5.3** Professional Issues 46](#_Toc196932070)

[**5.3.1** Legal Issue 46](#_Toc196932071)

[**5.3.2** Social Issue 46](#_Toc196932072)

[**5.3.3** Ethical Issue 46](#_Toc196932073)

[**5.3.4** Environmental Issue 47](#_Toc196932074)

[**Chapter 6 Conclusion** 48](#_Toc196932075)

[**Bibliography** 49](#_Toc196932076)

# **Abstract**

The rise of digital services has created new expectations for smarter travel management systems. Universities and campuses in particular need platforms that are tailored to their specific travel needs. Traditional travel planning still relies on fragmented information. This dispersed information leads to slow processes and weak real-time interactions. As these challenges become more apparent, the need for focused digital solutions is growing. Mainstream travel platforms continue to lead the market, but often fail to meet the needs of campus users. Navigation feels complicated. Information overwhelms rather than guides. Support for community tourism planning remains limited. Structured organization and meaningful social engagement are both critical to academic communities, but remain difficult to achieve on these platforms.

This project creates a modular web-based campus travel guide system. Its core features include event management, easy user registration, blog sharing, and real-time communication. The article reviews the shortcomings of existing systems, describes a user-centered platform design, and provides a new digital pathway for improved campus travel planning and community interaction.

***Keywords:*** Spring Boot Framework, MySQL, Travel Website

# **Abbreviations**

|  |  |
| --- | --- |
| Name | Description |
| JSP: | Java Server Pages |
| JPA: | Java Persistence API |
| JDBC: | Java Database Connectivity |
| DAO: | Data Access Object |
| HTML: | Hypertext Markup Language |
| CSS: | Cascading Style Sheets |
| JS: | JavaScript |
| SQL: | Structured Query Language |
| IDE: | Integrated Development Environment |
| API: | Application Programming Interface |
| UI: | User Interface |
| UX: | User Experience |
| HTTP: | Hypertext Transfer Protocol |

# **Glossary**

|  |  |
| --- | --- |
| Name | Description |
| JSP: | Java Server Pages |
| JPA: | Java Persistence API |
| JDBC: | Java Database Connectivity |
| DAO: | Data Access Object |
| HTML: | Hypertext Markup Language |
| CSS: | Cascading Style Sheets |
| JS: | JavaScript |
| SQL: | Structured Query Language |
| IDE: | Integrated Development Environment |
| API: | Application Programming Interface |
| UI: | User Interface |
| UX: | User Experience |
| HTTP: | Hypertext Transfer Protocol |

# **Introduction**

## Background

Travel websites currently play a key role in the tourism industry. Users use them to easily access travel planning tools and reliable information [1]. These platforms are especially valuable for campus-based travel, including student groups, educational field trips, and extracurricular excursions. Tourism continues to be one of the most dynamic industries in the world. Holiday seasons have seen a surge in travel, with short-haul trips accounting for a major portion of the total. The rise of internet technology, mobile adoption and social media platforms [2] has made travel websites more important. Students and faculty can use these sites to browse information about attractions and thus plan trips that meet educational and leisure goals.

For campus outings, these platforms recommend travel routes and plans. They provide options for group itineraries, budgets, and destination arrangements [3]. Paper-based experience posts and other third-party services traditional planning methods are fading away. Digital tools now offer real-time updates, user reviews, and interactive planning features. Travel websites provide students with recommended itineraries, accommodations [4] and clear safety guidelines as well as culinary experiences to advance. They also build teamwork by allowing teams to share itinerary tips [5]. These sites connect schools and travel providers in new ways.

## Aim

This project develops a web-based campus travel system to streamline campus tourism, offering secure, user-friendly trip planning, reservations, and real-time updates, with chat, blogs.

## Objectives

|  |  |
| --- | --- |
| Goal | Task |
| Literature Review and Market Analysis | Investigate existing tourism platforms to analyze their features, focusing on user interface design, booking systems, and customer interaction functionalities. |
| System Architecture and Functionality Design | Investigate existing travel management platforms to identify core functionalities.  Plan the modular structure of the travel management system, including user roles and permissions. |
| Database Design and Integration | Define relational database schemas for storing user profiles, travel packages, and booking records.  Establish secure connections between the SpringBoot backend and MySQL database. |
| Backend System Development | Develop RESTful APIs using SpringBoot to handle user authentication and booking transactions.  Integrate third-party services such as payment gateways and map navigation tools. |
| Frontend Interface Development | Design a responsive and intuitive user interface for browsing travel packages and services.  Implement features for users to view itineraries, book accommodations, and submit inquiries. |
| Testing and Quality Assurance | Conduct unit and integration testing for critical functionalities like payment processing.  Optimize system performance by addressing latency in API responses. |
| Release Plan | Prepare the project for deployment on localhost.  The release will include core functions such as travel browsing, booking services, and inquiry submission. |

## Project Overview

### Scope

This project is to design and build a Web-based travel management system for schools. The goal is to utilize modern web technologies to help users post campus tour activities, make campus tour planning easier through grouping and chatting, and also increase user participation. The platform supports two roles. Administrators can post and manage and review travel events. Users can sign up for trips, share their journeys via blogs, and chat with others in real time [6,7]. The core features of the system include smooth event registration, blog creation with comment moderation, user comments, and robust role-based access control. The system has a modular setup. As far as system functionality is concerned, the platform offers the following features：

1. Activity management function: travel activities can be created, edited and deleted. Users can post detailed information about the trip including budget, start time and address with a limit on the number of people can also edit and delete the activity.

2. User registration system: Users need to register their account by email or school registration number.

3. Blog sharing: users can post travel blogs, upload pictures, comment on posts and send comments.

4. Chat Room: Travelers can chat with the creator of the activity to get more detailed information about the task.

5. Evaluation Function: Users can evaluate the activity, this evaluation will be weighted to calculate the credibility of the activity creator, etc. so that users can use it as reference information.

The project focuses on campus communities. It brings strong social engagement tools and a modular, scalable system design. Travel planning becomes easier. Online interaction among academic travelers grows stronger.

### Audience

Web-based travel management systems serve two main groups of people[8]. Users and administrators deal with different parts of the platform.

Users: Are students, faculty and staff from schools. They browse travel events, sign up for trips, join tours, and share their experiences through blogs and live chats.

Administrators are responsible for managing the entire system. They manage user roles, review activities, and can handle offending comments and ratings to keep the platform safe, review content, and handle day-to-day operations.

A clear division of roles keeps the platform organized. Each role works within a function.

# **Background Review**

## Overview of Existing Smart Tourism Platforms

The rise of technologies like the Internet of Things [9], cloud computing, artificial intelligence, and mobile networks is changing tourism. The industry is moving into a full digital transformation. Smart tourism platforms are now essential [10]. They help improve visitor experiences and make destination management more efficient. These platforms work across different devices. Tourists, travel agencies, and government staff can interact easily through PCs, mobile phones, and kiosks [11].

Such systems typically integrate multiple service modules, including real-time scenic spot information, personalized itinerary recommendations, online booking systems, navigation guidance, tourist behavior analysis, and data dashboards for management use [12]. Smart tourism platforms bring together different stakeholders into one digital ecosystem. They make daily operations smoother and faster. They help destinations manage resources better. They also lead the way in promoting sustainable tourism practices. Decision-making becomes smarter and more data-driven [13].

## Competitive Analysis of Existing Tourism Management Systems

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| System | Strengths | Limitations | Key Features | User Experience |
| System A:Ctrip  (https://www.ctrip.com/) | Extensive array of features | Intricate user interface | Vast travel information repository | High learning threshold |
| Multilingual capability | Daunting for beginners | Customer-submitted reviews and ratings | Worldwide destination reach |
| Data deluge | Numerous service suggestions | Necessitates extensive navigation | Requires significant navigation |
| System B: Kayak (https://www.cn.kayak.com/) | Elaborate design | Lack of data protection | Swift data handling | Simple interface design |
| Instant information access | Limited features | Fundamental reservation functions | Fairly easy to use |
| Cost comparison utilities | Restricted service range | Hotel-focused platform | Simple reservation procedur |
| System C: Qunar  (https://www.qunar.com/) | Optimized for mobile | Erratic software updates | Instant location-based services | Social media connectivity |
| Restricted personalization | User-generated review incorporation | User review integration | Platform offering multiple services |
| Platform offering multiple services | Fluctuating pricing options | Full-service travel booking | Easy-to-navigate mobile interface |

Table1 Analysis of Existing Tourism Management Systems

This project positions itself by comparing three major tourism management platforms. Ctrip, Kayak, and Qunar offer important lessons for system design and feature planning.

Ctrip [14] stands out for its wide range of features. It holds a large database covering destinations worldwide. It supports multiple languages and hosts many user reviews and ratings. However, the platform feels complex. New users often find the interface hard to navigate. The heavy load of information demands extra effort and slows down quick searches [15]. The user experience suffers for those looking for fast, simple answers.

Kayak [16] takes a different path. It uses a clean and simple design. Information is easy to find. Booking tools and price comparisons work smoothly. The platform feels user-friendly, especially for hotel reservations. Still, Kayak has a narrow focus [17]. Services outside hotel bookings stay limited. Data security concerns and few customization options hurt user trust slightly.

Expedia [18] offers a mobile-first, multi-service platform. Users can book flights, hotels, rental cars, and vacation packages all in one place. Real-time location features and social media tools add to the mobile experience. The mobile app feels smooth and satisfying for travelers on the move. But Qunar sometimes falls behind with slow updates and limited customization [19]. Pricing can also vary too much, weakening user confidence.

Each platform shows its own strengths and struggles. Ctrip leads in information depth. Kayak wins in quick booking. Qunar covers wide services with mobile focus. But all show pain points like complexity, narrow focus, or uneven quality.

My tourism website focuses on the needs of students. On the basis of integrating the practical functions of mainstream platforms, it solves the problems of complicated information and complex operation. The website page design takes simplicity as the core, and focuses on the development of role-exclusive functions (such as student special booking, teacher travel agency release activities) and social interaction modules (travel dynamic sharing, online chat, etc.), which are committed to creating a travel service platform with clear goals, convenient operation and full of interactive fun for students, teachers and travel agencies.

## Research Methodology and Technical Architecture

This study uses Java Spring Boot as the back-end framework [20] and Vue.js as the front-end technology to build a modern tourism management system [21]. Through microservice architecture and front-end separation design, the system aims to provide a high-performance and scalable tourism service platform. Spring Boot provides strong dependency injection [22], security features, and microservices support for the backend, while Vue.js ensures a responsive and interactive interface for the front-end. Referring to leading platforms such as Ctrip and Kayak, the system will integrate advanced features, including intelligent search, real-time booking, user rating system and personalized recommendation. The system not only meets users' diverse travel service needs, but also provides smooth and intuitive user experience. The key technical innovations include efficient data processing mechanism, flexible component architecture and comprehensive error handling strategy.

In particular, the system's real-time data processing capabilities are informed by a comprehensive review of applications [23], which emphasizes the importance of low-latency processing and robust security measures

.

# **Methodology**

## Approach

### Development Model

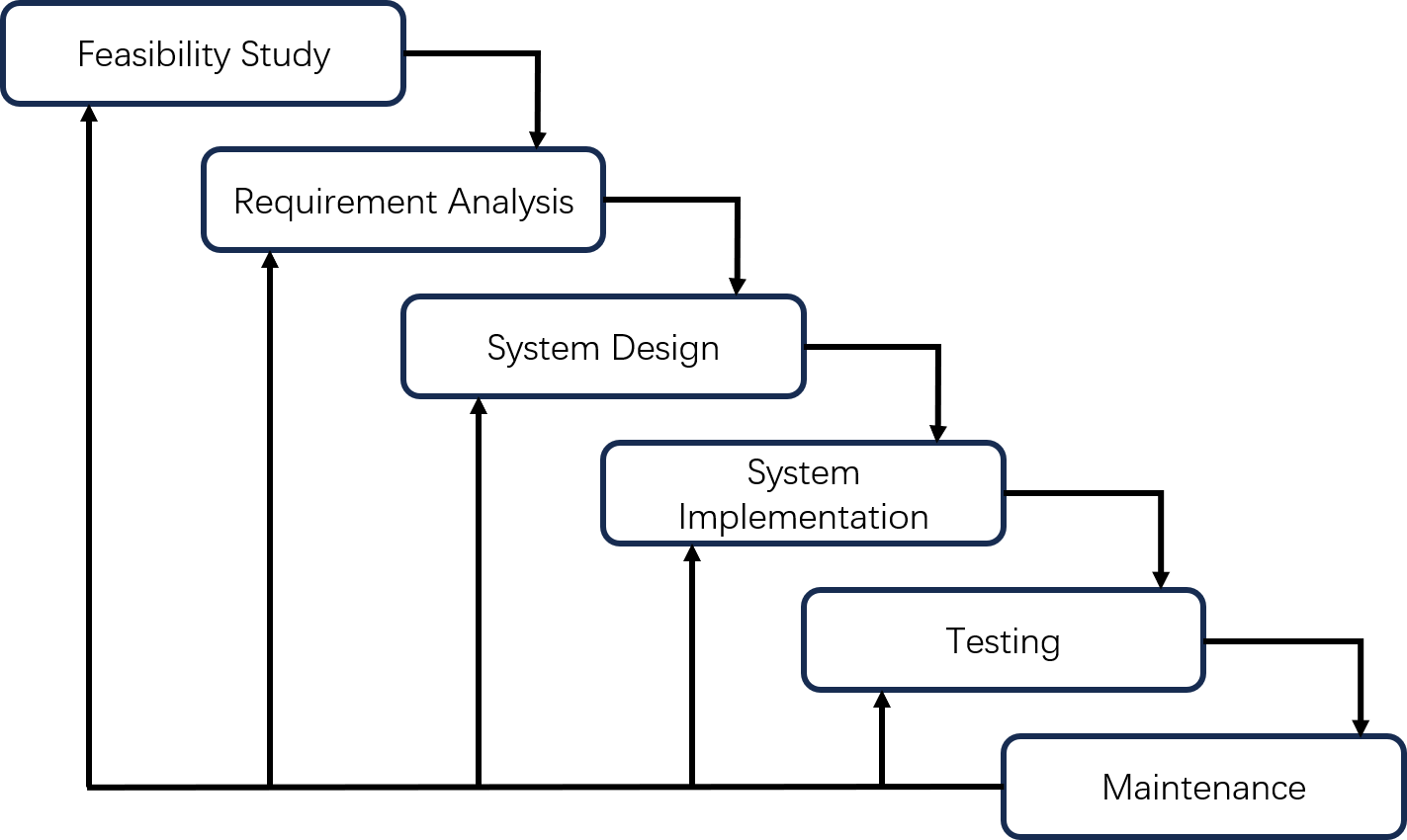
The project employs the Waterfall Model, a linear and structured software development methodology, to ensure systematic progression and rigorous documentation.

Figure 1: Waterfall Model

This model aligns with the project’s well-defined requirement. The development process starts with feasibility checks. Technical and academic viability are confirmed early. Requirement gathering follows through stakeholder interviews and use-case diagrams. The Software Requirements Specification (SRS) formalizes all needs. System design defines the full architecture. Spring Boot runs backend services. Vue.js handles frontend interactivity. MySQL manages database tasks. UML and ER diagrams support the design work. Implementation turns plans into working code. RESTful APIs and Vue components are built in a modular way. Testing checks every function through integration trials and user flow simulations. Maintenance focuses on post-launch improvements. Database query speeds and system optimizations stay a top priority. The Waterfall Model keeps each phase clear and organized. Every phase creates real deliverables like SRS files and test reports. These documents make traceability easier and lower project risks. Strong upfront planning and phase-end reviews keep everything aligned with academic goals. The model fits well for a project that demands solid documentation and high reliability in managing tourism systems.

### Feasibility Analysis

System design adopts a clean and modern style, friendly to student users. The color palette uses soft blues and neutral grays to create a calm and trustworthy feel. Action elements like sign-up buttons and chat prompts are highlighted with accents of teal or green. Visual focus stays balanced without overwhelming users. Card-based modular layouts clearly separate sections for travel activities, blogs, and chat rooms. Navigation remains easy even for first-time visitors.

Functions are built for both simplicity and engagement. Travel modules allow users to browse, filter, and sign up for activities with minimal effort. Blog sections support rich-text posts and images. Comment areas encourage community discussions under moderation. Real-time chatting works through WebSocket connections. User A sends a message through the chat box. The backend server pushes the message instantly to User B’s client. The chat window updates without needing a page reload. Fast, low-latency communication strengthens social ties during planning and sharing. The platform is built to be stable, user-friendly, and ready to make campus travel planning more connected and engaging.

Technical feasibility: The tech stack fully meets the needs of the campus travel guide system. Spring Boot delivers a stable and modular backend. RESTful APIs are built with strong error handling and security. Vue.js supports a fast, dynamic front-end. Layouts respond smoothly across desktops, tablets, and mobile devices. MySQL manages relational data with high efficiency. Transaction control and indexing ensure performance stays strong during high traffic events like mass sign-ups and comment submissions. Standard academic-grade servers support expected usage. Load tests with 1,000 concurrent users validate the system’s scalability and responsiveness.

Legal feasibility: The project follows all rules for data protection, intellectual property, and digital services. GDPR and local privacy laws are covered with encrypted user login, role-based access control, and clear consent steps for data collection. Legal experts reviewed the setup. User-generated content like blog comments and activity reviews is handled under strict ethical guidelines. Secure API design and strict database permissions lower the risk of unauthorized access.

## Requirements Analysis

A full requirements analysis shaped the campus travel guide website. The core function of this tourism website is determined by comparing the major head tourism platforms, the suggestions of middle school students in Tieba and the comments of travel agencies around the school. Common problems in existing travel websites became clear[24]. Navigation felt complicated. Services lacked focus. User engagement stayed low. Specific improvements were mapped out to guide the system design. User-centered ideas led the process. The goal was to close the gap between old planning methods and modern digital needs. A use case diagram was created based on the findings.

The functions of the administrator mainly include: login registration, activity type, activity, activity reservation, activity travel evaluation, blog, blog comment, notification information management.

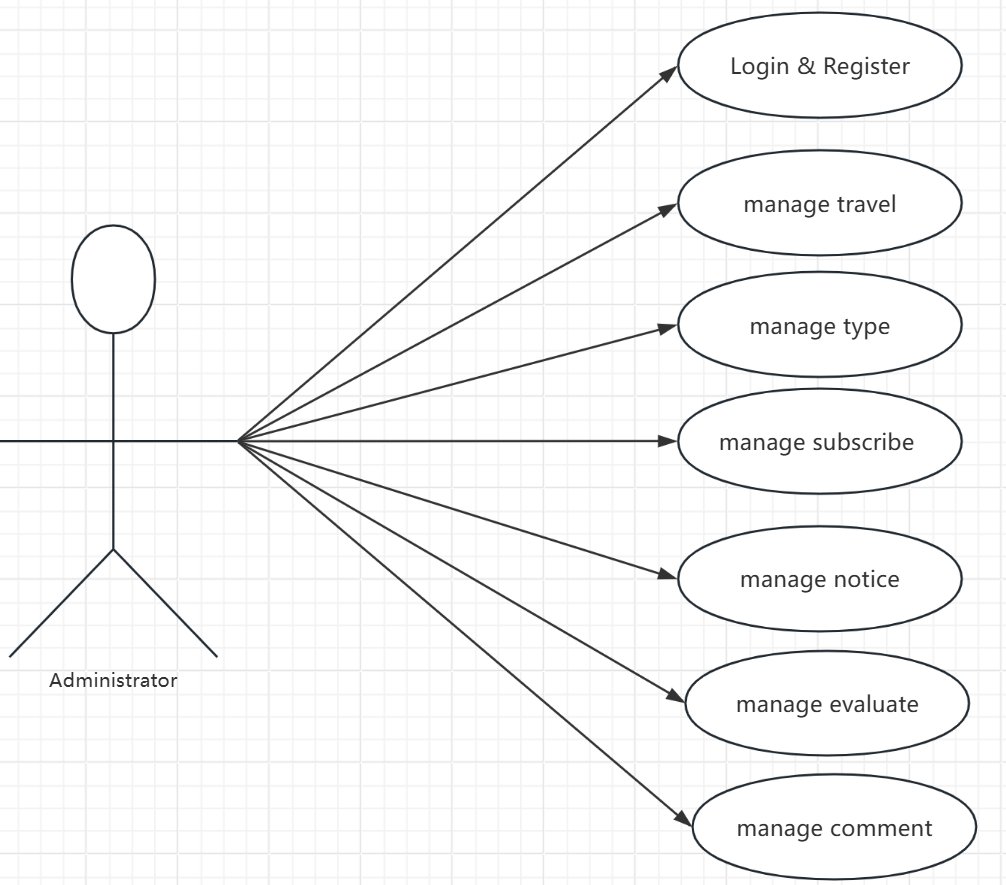


Figure 2: Administrator Case

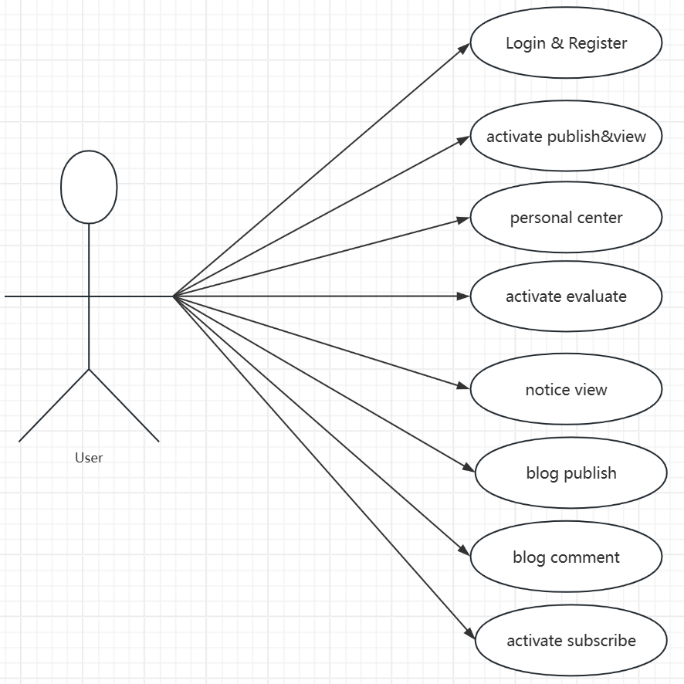
The main functions of users include: login and registration, Posting and browsing registration activities, evaluating activities, Posting and commenting on blogs, and viewing notice and announcement information.

Figure 3: User Case

### Function Analysis

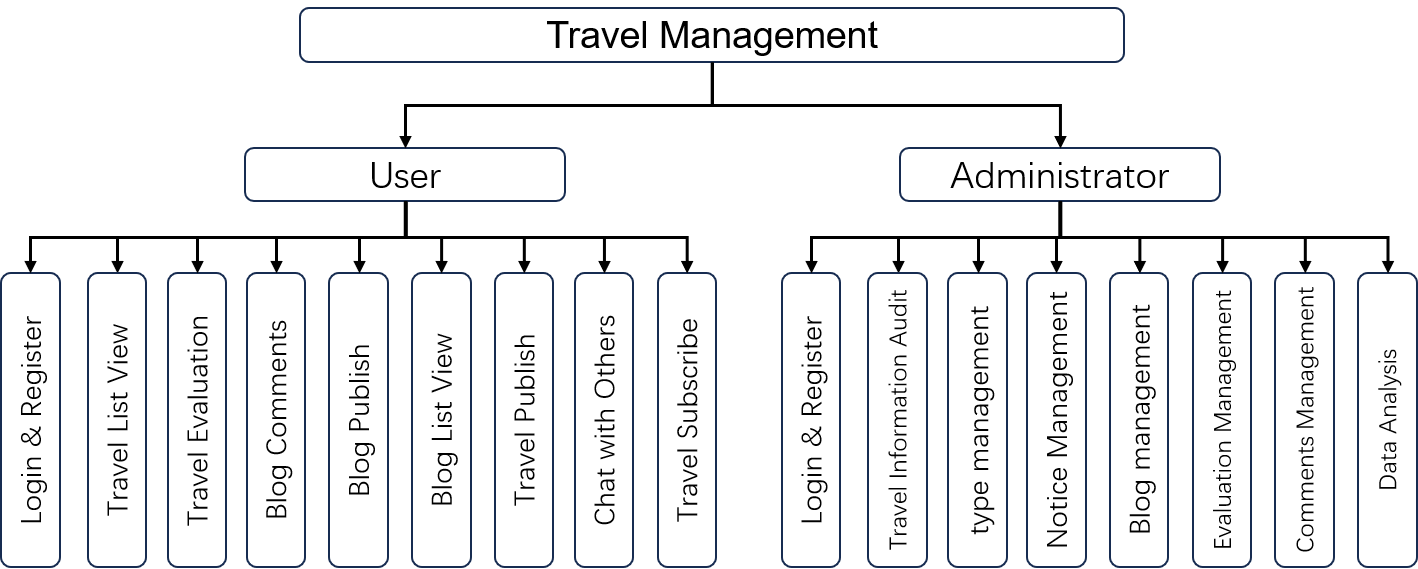
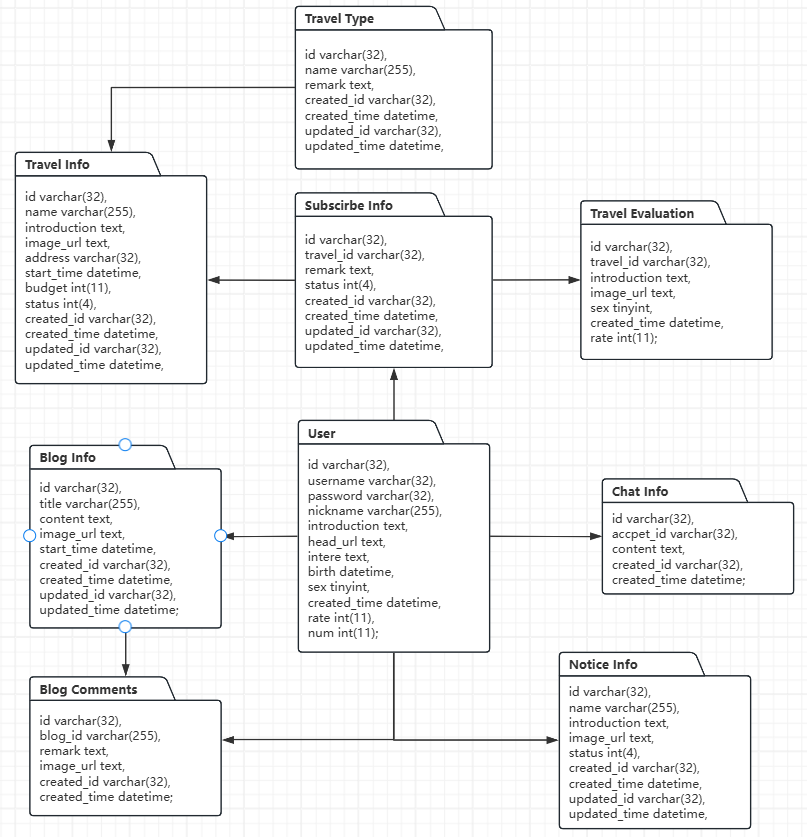
This website is a powerful travel management website. The various functions it performs provide users with a very convenient one-stop service. The specific functions are shown in Figure 4.

Figure 4: Function Analysis of Website

Each function supports a smoother and more connected user experience. The real-time chat module stands out as a key part of the platform. The integrated chat system allows users to exchange information instantly. Trip details, travel experiences, and group coordination can all happen without delays. WebSocket technology supports the chat feature. Messages sent by one user appear immediately for others, without any need to refresh the page. Fast communication strengthens community ties and makes group planning easier. The platform builds a more dynamic, interactive travel community through this seamless real-time connection.

### Database Design

Figure 5: Database

A database is absolutely essential for this project. The database stores information about user and project characteristics. This includes a history of user behavior, such as browsing, purchases, and bookmarks, as well as information about product descriptions, labels, and categories. The database can also be used to quickly retrieve and obtain user and item characteristics for similarity calculations. This database has 9 tables: users, travel, type, evaluations, comments, blog, chat and notice, subscribe mainly stores users' personal information and registration information. It also stores basic information such as tour name, budget, description and pictures.

These tables form the infrastructure for the database of a travel management system, and each table serves a specific purpose. Linked through foreign keys, they establish a robust relational data model that efficiently stores and queries travel information. The design supports tracking user interactions (e.g., itinerary bookings, blog interactions) . Below are the specifics of each table and their relationships:

1. User: This table stores user profiles, including user ID (id), username (username), password (password), nickname (nickname), profile image (head\_url), interests (intere), and registration time (created\_time). The user ID is the primary key, referenced across tables like Subscribe Info and Blog Comments to associate actions with user identities.
2. Travel Info: This table manages travel itinerary details, with fields such as travel ID (id), name (name), destination address (address), budget (budget), and creator ID (created\_id). The travel ID serves as the primary key, linking to tables like Subscribe Info (for subscriptions) and Travel Evaluation (for reviews).
3. Travel Type: Categorizes travel themes (e.g., adventure, cultural), with type ID (id) as the primary key. It connects to Travel Info via travel\_id to classify itineraries, enabling filtered searches (e.g., "find all beach vacations").
4. Subscribe Info: Tracks user subscriptions to specific itineraries. The subscription ID (id) is the primary key, while travel\_id (linked to Travel Info) and created\_id (linked to User) ensure data integrity. Cascading updates maintain consistency when travel details change.
5. Travel Evaluation: Stores user reviews and ratings for trips. The evaluation ID (id) is the primary key, with travel\_id referencing Travel Info to associate feedback with specific itineraries. Fields like rate (score) and image\_url (review photos) enhance recommendation algorithms.
6. Blog Info: Manages travel blogs posted by users. The blog ID (id) is the primary key, and created\_id (linked to User) identifies the author. Relationships with Blog Comments allow threaded discussions, supporting community engagement.
7. Blog Comments: Records comments on blogs, with blog\_id (linked to Blog Info) and created\_id (linked to User) ensuring traceability. The comment ID (id) enables moderation tools, while image\_url allows media attachments in discussions.
8. Chat Info: Facilitates user-to-user communication. The chat ID (id) is the primary key, with acccpet\_id (receiver) and created\_id (sender) both referencing User, enabling real-time messaging features.
9. Notice Info: Manages system announcements (e.g., policy updates). The notice ID (id) is the primary key, and created\_id (linked to User) tracks administrative actions, ensuring audit compliance.

### System Design

The design of system is adapted in a Vue frontend, Spring Boot backend, and MySQL5.7 database architecture to create a structured and maintainable system. Vue serves as the View layer, rendering dynamic user interfaces and handling client-side interactions, while Spring Boot acts as the Controller, managing HTTP requests, processing business logic, and interacting with the Model layer. MySQL5.7, as the Model, ensures secure and efficient data storage and retrieval. This separation of concerns facilitates parallel development, enhances scalability, and simplifies maintenance by clearly defining the responsibilities of each component.

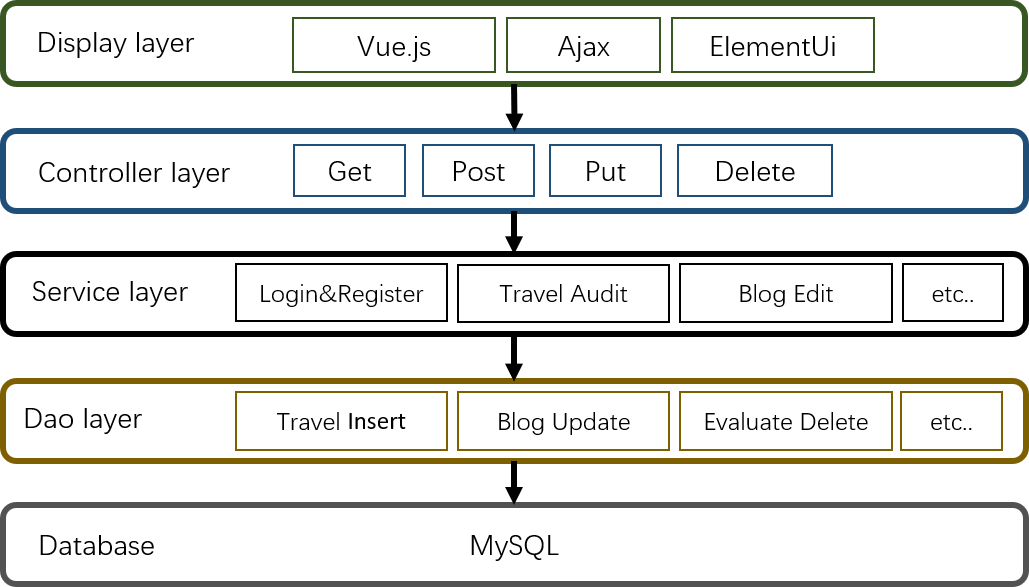


Figure 6: System Structure

In this architecture, Vue components focus on presenting data and capturing user inputs, leveraging its reactivity system for seamless UI updates. Spring Boot controllers handle incoming requests, invoking services to process business logic and access the MySQL5.7 database through repositories. The database layer handles relational data with full ACID compliance. Complex queries run smoothly and reliably. The system design keeps front-end and back-end separate. Each side can grow and change without breaking the other. Code becomes easier to reuse. Resources get managed more efficiently. The result is a strong, flexible system ready to handle new demands.

## Technology

The technology used in this project are as follows:

|  |  |
| --- | --- |
| Name | Description |
| a) Hardware: | ROG Strix SCAR |
| b) Software: | IntelliJ IDEA2020.1.3, Navicat 16.3.9, Vs Code 1.87.2. |
| c) Operating system: | Windows 11. |
| d) Visualization technology: | Matplotlib, ECharts . |
| e) Background frame: | Torch 1.13.0,SpringBoot 2.6.3, Vue 2.0. |
| f) User Interface: | JavaScript, HTML, CSS. |
| g) Database: | MySQL5.7. |
| h) Deployment | FastAPI server deployment with Uvicorn for backend services; Local hosting via localhost for development;In the future, we will deploy our project on cloud servers and use nginx as the code for the front-end servers. |

Table 2: Technology List

## Project Version Management

The tourism management system uses Gitee for version control. Each finalized version is uploaded to the Gitee repository for backup and tracking.

- URL:

Git handles all source code management during the project. It records every change, manages different branches, and makes merging updates simple. These features keep the development process organized and reliable.

Git works in a distributed way. Every team member keeps a full copy of the project’s history. This setup lowers the risk of data loss and makes collaboration easier.

Git is trusted across open-source communities and industry projects. It fits into many development workflows without trouble. Its powerful branching system lets developers create new features, fix bugs, and experiment safely. Code stays clean, traceable, and easy to maintain throughout the entire development cycle.

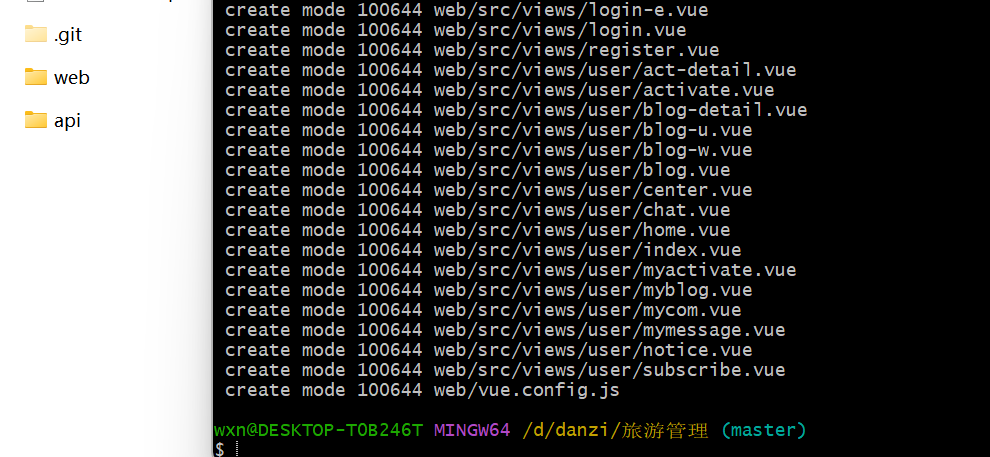


Figure 7. Git Cmd

# **Implementation and Results**

## Springboot Framework

The Spring Framework, an open-source Java platform, provides an IoC container for dependency-injected object management and AOP for declarative transaction handling, enhancing maintainability and modularity. It supports diverse persistence tools, web frameworks, and features like robust transaction management and data access integration, while Spring MVC improves layer separation over legacy systems.

## Function Introduction

This project has been carefully designed to provide good interaction for users in order to have a better experience. When the user is not logged in, there is a navigation bar at the top of any page containing a search box and buttons for home, login, and registration. After the user logs in, the home page, travel list and user avatar are displayed in the top navigation bar. The project also designed a home page, travel details page, registration page, blog page, blog comments, and a conversation page. Finally, the basic login and registration page.

### Home Page

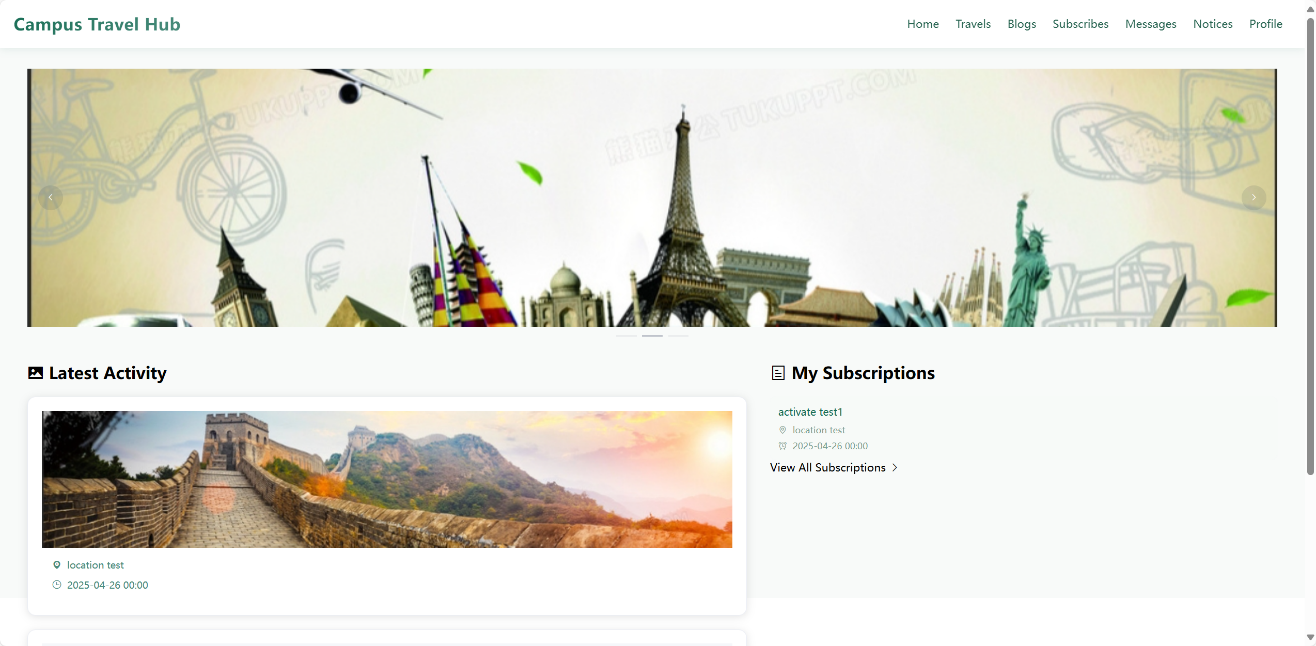
Users can easily find the page they want to go to on the homepage and can easily return to the homepage from any page. The products on the homepage are easy to see and can be browsed at will. There is a search box at the top of the homepage and on the left side of the homepage. Detailed information can be found in Figure 8.

Figure 8: Home Page

### Registration & Login

Figure 9: Register Page

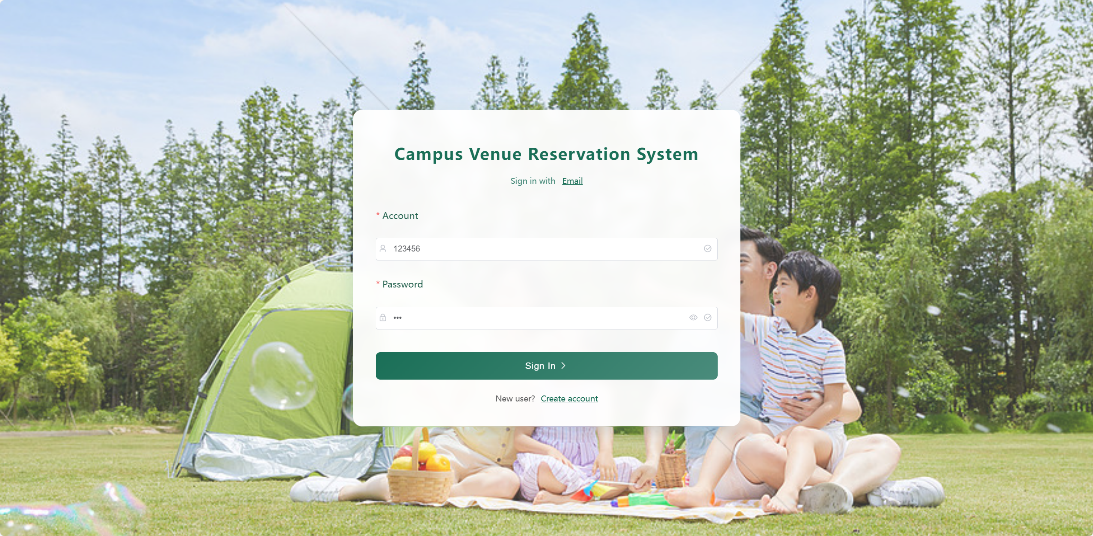
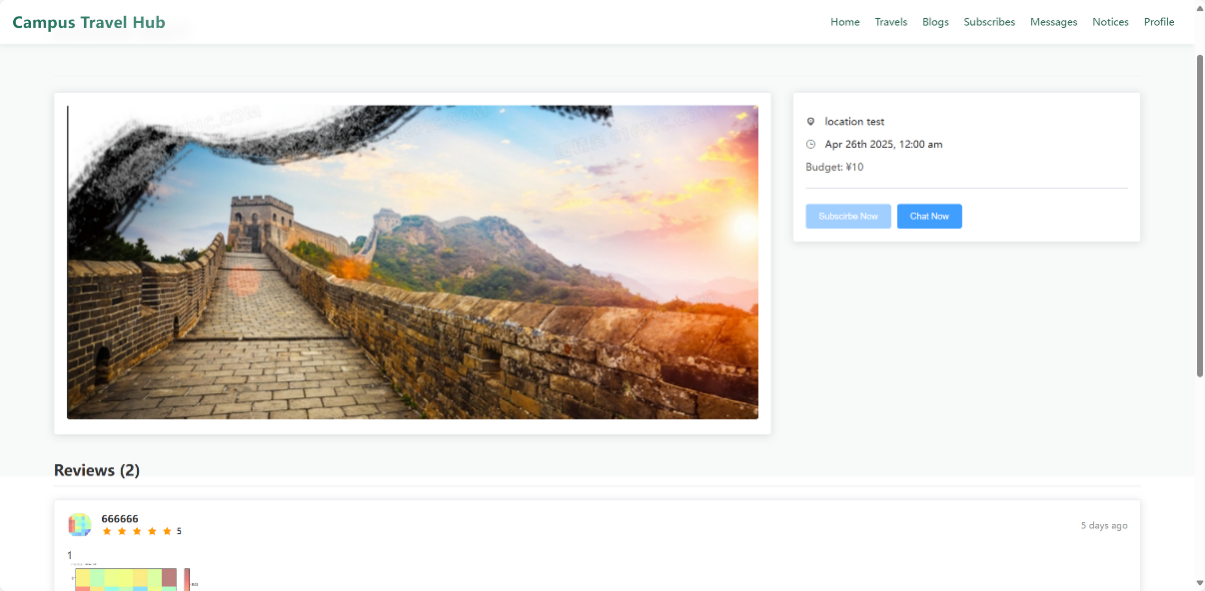
The registration process ensures data standardization based on field constraints in the User table: The email address must be unique, the password must be encrypted, and the username must be limited in length (varchar(255)). When the user fills in the form, the password input is protected by a mask. After submitting the form, the system detects whether the mailbox already exists (based on the unique index of the database). If it is repeated, the system prompts "The mailbox has been registered" and provides the "Login now" link (jump to Figure 9). New user data records the registration time via created\_time, providing an audit basis for subsequent behavior analysis, such as Travel Evaluation records.

Figure 10: Login Page

The login function is based on the User table (id, username, password) in the database. After clicking the "Login" button in the upper right corner of the interface, enter the registered email address (corresponding to the email field) and password (encrypted and stored in the Password field). The system verifies the password hash value (such as BCrypt algorithm) through Spring Security, and matches the email uniqueness constraint in the database to ensure the identity legitimacy. After successful login, users can access personalized functions such as Subscribe Info and Blog Info, and the foreign key user\_id ensures data ownership and permission isolation.

### Travel Detail Page

We designed the tour details page, where you can see the budget of the tour information and the evaluation of the people who have participated in the tour. At the same time, I gave the user the button to register and the button to communicate with the person in charge.

Figure 11: Travel Detail Page

### Blog List Page

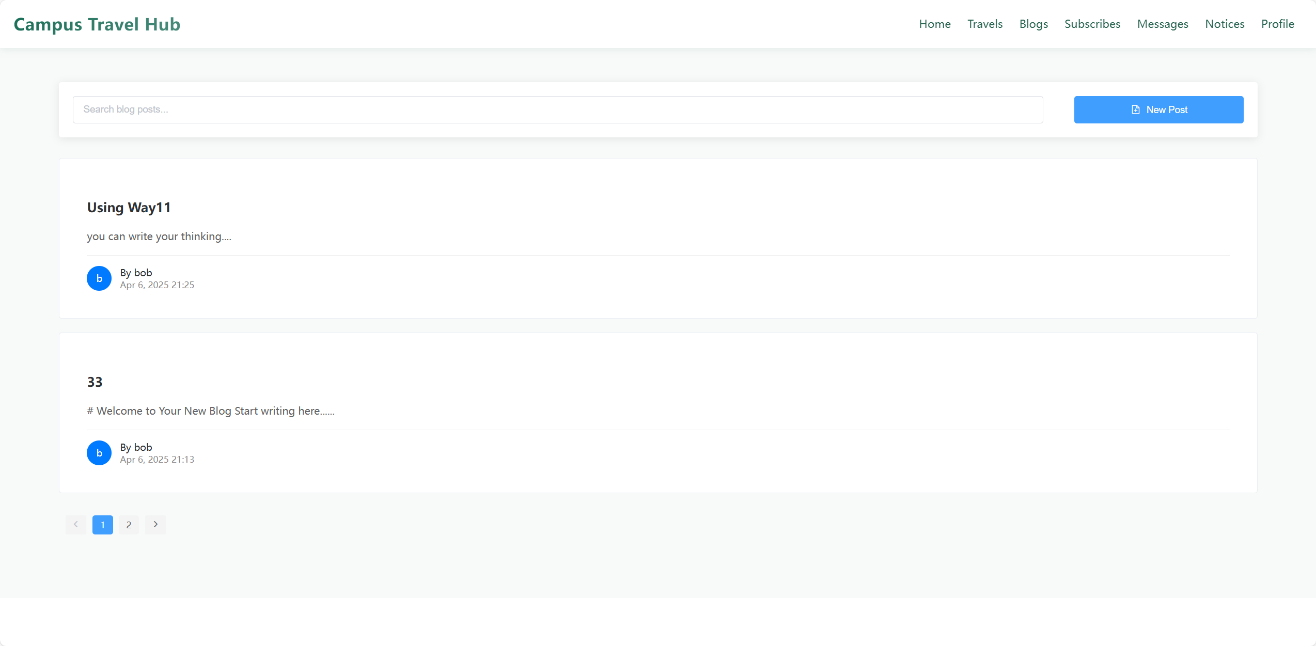
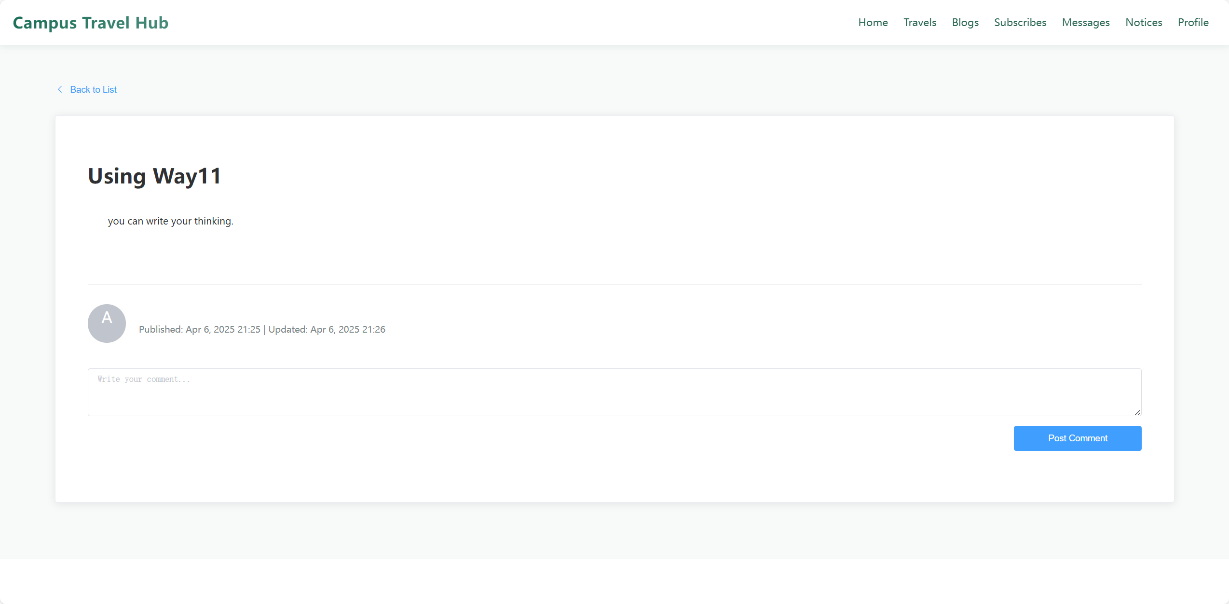
Users can see the title content and author information of the blog in the blog list page and can search the interested blog content by the blog title.

Figure 12: Blog List Page

### Blog Detail Page

In the blog's details page, users can see the title content of the blog and author information and can comment on the blog.

Figure 13: Blog Detail Page

### Subscribe List Page

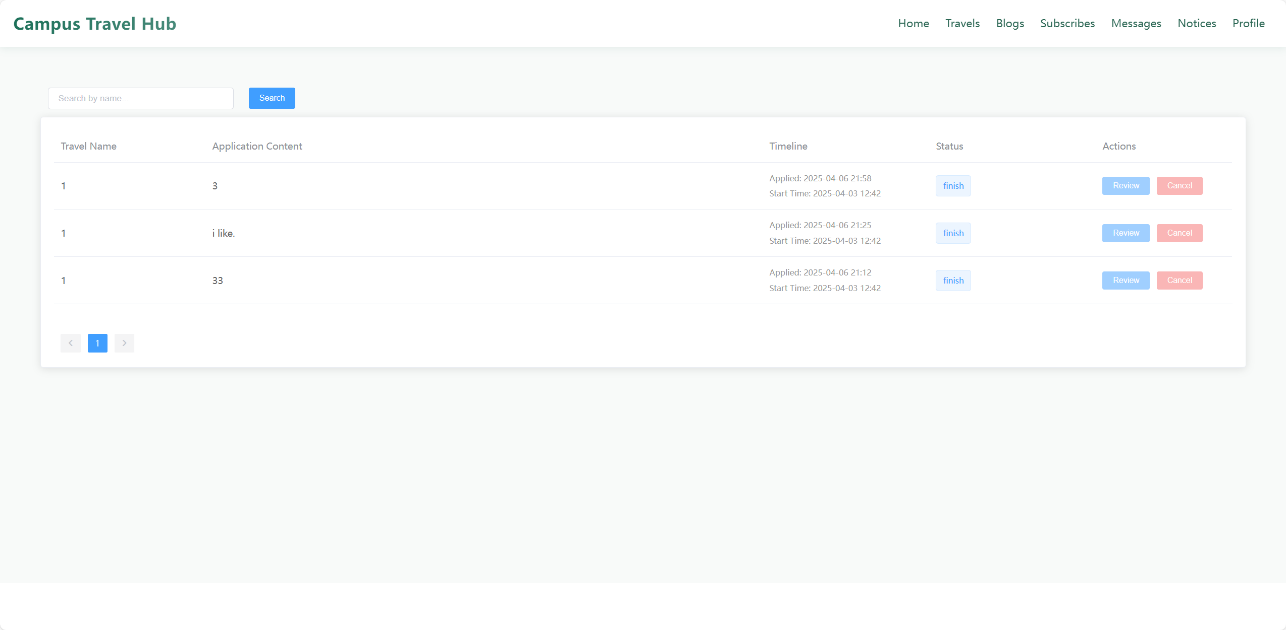
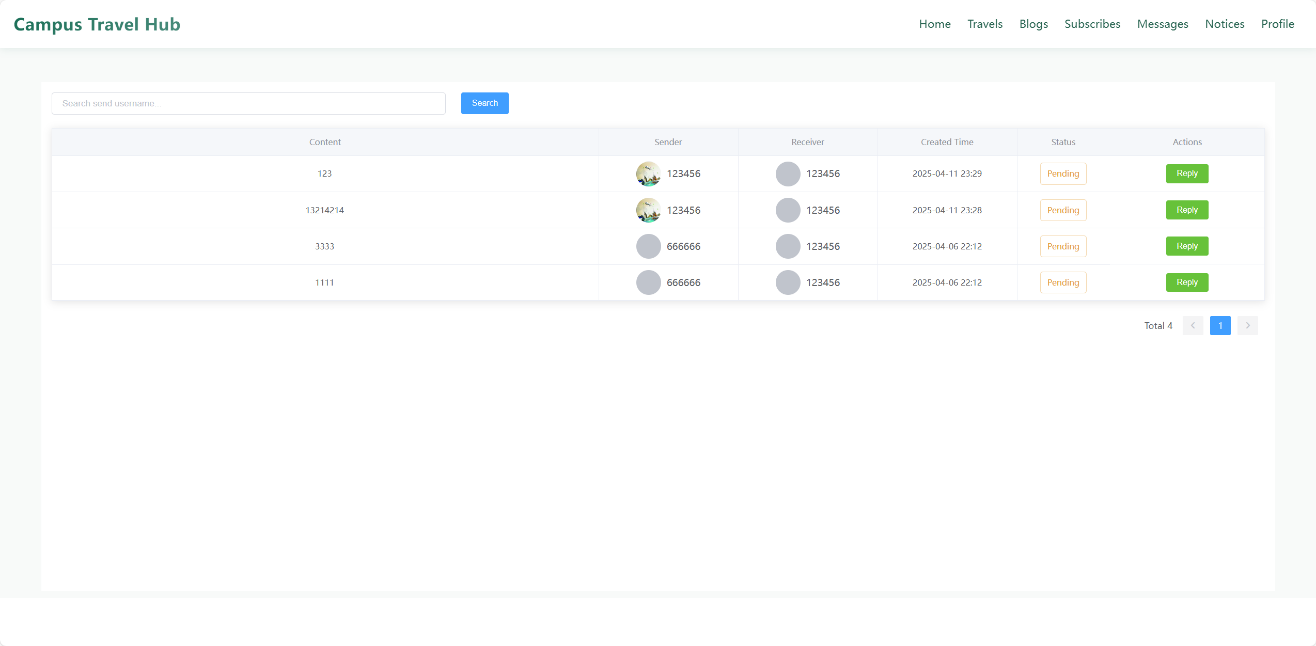
Users can withdraw their registrations on trips they have already signed up for, as well as review them. Due to the circumstances of the event, I do not allow cancellation of registration after the event is about to start, and I do not allow cancellation of registration after the event is over. Of course, only one evaluation is allowed after each trip.

Figure 14: Subscribe List Page

### Message List Page

The user can reply to the message in the message list or search for the message that needs to be replied.

Figure 15: Message List Page

### Notice List Page

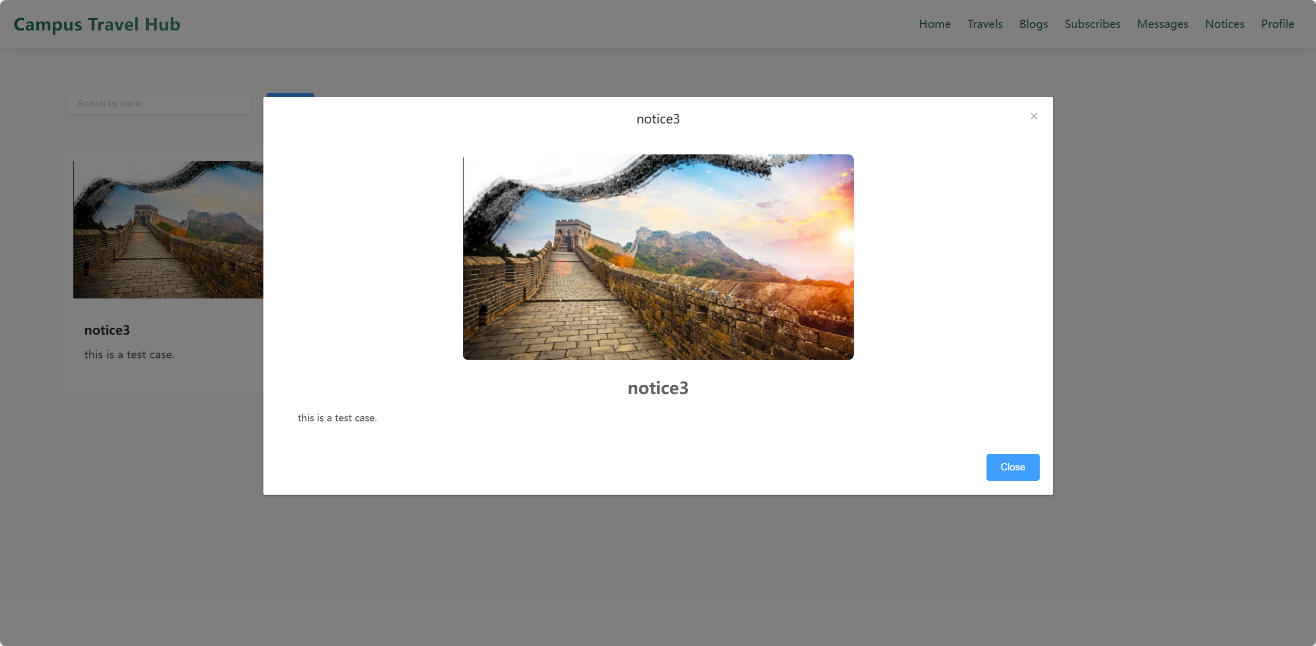
 The notification page can view the notification information of the website and can go to view the details of the notification.

Figure 16: Notice List Page

### Personal Center Page

Personal Center can modify the basic information of the user and can reset the password and log out. In addition, I also provide my reviews, My comments and My activities and My blog buttons to help users quickly edit and modify their website information. For example, users can use the My Travel button to jump to their posted Tours to edit and withdraw.

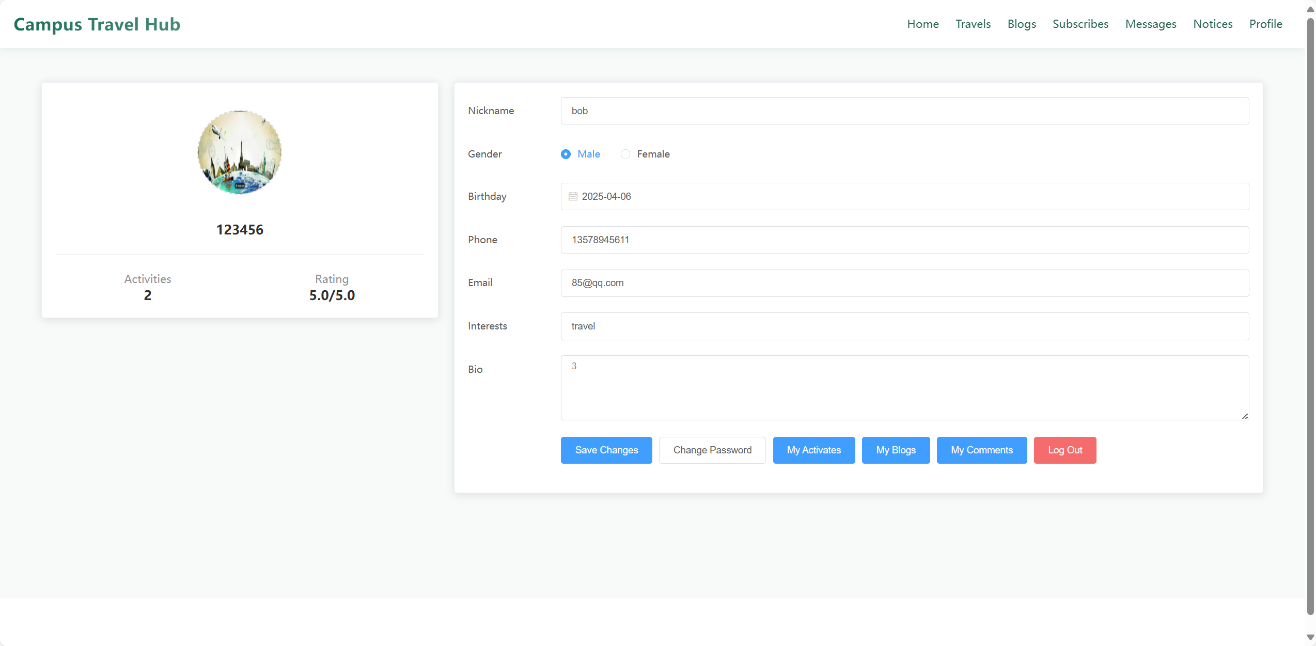


Figure 17: Personal Center Page

### User Chat Page

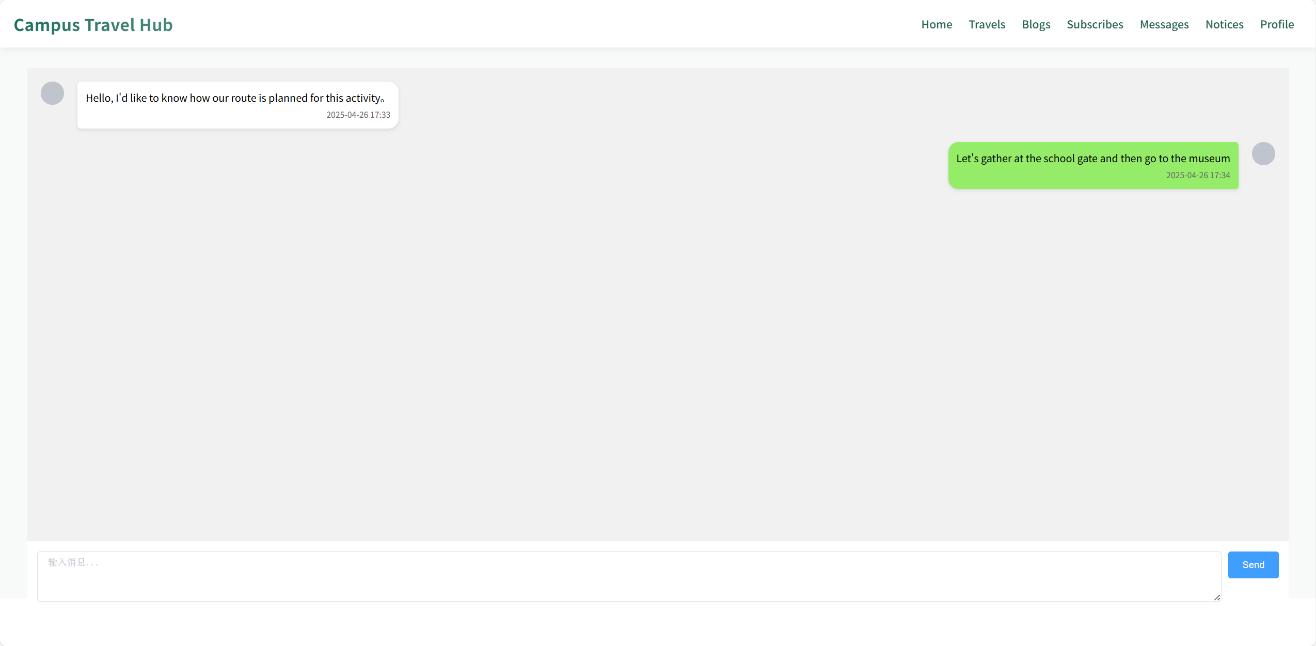
On the user conversation page, users can ask the initiator of the tourism activity for details of the activity. My interface model is the chat page of wechat, which is easy to operate and simple and clear.

Figure 18: User Chat Page

## Function Test

| Feature | Test Case | Test Precondition | Test Step | Expected Result |
| --- | --- | --- | --- | --- |
| Register | Enter information that meets the specifications | E-mail address not used | 1. Go to the home page of the site and click on the Register button to enter the registration page. Register Enter information that meets the specifications E-mail address used 2. Enter all the information according to the specification 3. Click the Register button | Jump to the login page and a prompt “Thanks for registering! You may login now.” |
| E-mail address used | The message “User with email xxx@xxx.com already exists!!!” appears. and the hyperlink “Login now!” appears. |
| Register | Enter the e-mail address in the wrong format | Other information conforms to the specification | 1. Go to the home page of the Travel guide site andclick on the Register button to enter the registration page 2. Enter your e-mail address in the wrong format. 3. Enter all other information according to the standard 4. Click the Register button | The message “Invalid email address.” |
| Enter the password in the wrong format | 1. Go to the home page of the Travel guide site and click on the Register button to enter the registration page 2. Enter your password in the wrong format 3. Enter all other information according to the specifications 4. Click the Register button | The message “Password must be 8 characters long and should contain letters, numbers and symbols.” appears. |
| Inconsistency between two password entries | 1. Go to the home page of the Travel guide site and click on the Register button to enter the registration page. 2. Enter all other information as specified 3. Enter a different password twice 4. Click the Register button | The message “Passwords must match” appears. |
| Login | Enter the correct e-mail address and password | Account is registered | 1.Go to the home page of the Travel guide site and click on the Register button to enter the login page  2. Enter your correct e-mail address and password. 3. Click the Login button | Successful login and jump to the Home Page of the Travel guide site |
| Enter an incorrect e-mail address or password | 1. Go to the home page of the Travel guide site and click on the Register button to enter the login page. 2. Enter the e-mail address and password that meets the specifications. 3. Click the Login button | The message “Email and password incorrect!!!” appears. |
| Enter a compliant e-mail address and password | Account not registered |  | The message “User with email xxx@xxx.com doesn't exist!” and the hyperlink “Register now!” appear. |
| Logout | Logging out of login from login status | Account is logged in | 1. Click on the user's avatar in the upper right corner of any page. 2. Click the Logout button | Jump to login page |
| Travel Management | Publish travel | Account not logged in | 1. Click on the product icons in any page 2. Click Add to Cart button | Jump to the login page with the message “You must login first!” and the hyperlink “Login now!”. |
| Account is logged in | Jump to the message alert page with the message “Publish successfully” |
|  | Withdrawing Travel | Travel has been published. | 1. Click on the withdraw in the upper right corner of any page. 2. Click on the Remove from List | Jump to the message alert page with the message “Withdraw successfully” |
| Blog Management | Publish a travel blog | User logged in | 1. Go to the blog management page  2. Fill in the title, body and label  3. Click the "Publish" button | Jump to the blog details page and the "Publish successfully" message is displayed |
| User not logged in | Jump to the login page, "Please log in first!" Prompt and "Sign in Now" link |
| Modify published content | User logged in | 1. Go to the blog edit page  2. Update the body content  3. Click "Save Changes" | Jump to the blog details page and the "Modified successfully" message is displayed |
| User not logged in | Jump to the login page, "Please log in first!" Prompt and "Sign in Now" link |
| Delete published content | User logged in | 1. Enter the blog management list  2. Click the target blog "Delete" button  3. Confirm the deletion operation | Jump to the blog details page and the "Delete successfully" message is displayed |
| User not logged in | Jump to the login page, "Please log in first!" Prompt and "Sign in Now" link |
| Notice Management | Publish a travel Notice | User logged in | 1. Go to the Notice management page  2. Fill in the title, content  3. Click the "Save" button | Jump to the Notice details page and the "Save successfully" message is displayed |
| User not logged in | Jump to the login page, "Please log in first!" Prompt and "Sign in Now" link |
| Modify published Notice | User logged in | 1. Go to the Notice edit page  2. Update the body content  3. Click "Save Changes" | Jump to the Notice details page and the "Modified successfully" message is displayed |
| User not logged in | Jump to the login page, "Please log in first!" Prompt and "Sign in Now" link |
| Delete published Notice | User logged in | 1. Enter the Notice management list  2. Click the target blog "Delete" button  3. Confirm the deletion operation | Jump to the Notice details page and the "Delete successfully" message is displayed |
| User not logged in | Jump to the login page, "Please log in first!" Prompt and "Sign in Now" link |

Table 3: Test Case List

## Performance Test

Performance testing is a common software testing methodology used to evaluate the

performance and responsiveness of a system under different load conditions. It focuses

on system throughput, response time, number of concurrent users and error rate, among

others. For Travel guide websites, performance testing can help the website to evaluate

the performance of the system during concurrent user access, high load and peak.

This project uses the load testing tool JMeter for performance testing and uses Aggregate

Report to present the final results.

Aggregate Report parameters are explained in detail:

|  |  |
| --- | --- |
| Name | Description |
| Label | Name attribute of each JMeter element. |
| # Samples | the number of requests, that is, how many requests were issued in this  test . |
| Average | the average response time for a single request, in milliseconds |
| Median | the median, which is the response time for 50% of users |
| 90%Line | response time for 90% of requests |
| Min | Minimum response time |
| Max | Maximum response time |
| Error % | Error rate = number of incorrect requests/total number of requests |
| Throughput | Throughput is the number of requests completed per second. |
| KB/sec | amount of data received from the server per second |

Table 4: Test Label List

Our test case:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Test Item | Test Step | Expect Result |
| Responsive state of the site under light load | Testing the load capacity of an Travel guide site with a small number of users accessing the site at the same time | 1. Using JMeter to simulate 30 and 50 concurrent users to visit the website respectively. 2. Monitor and record the response time of the website after each user operation. 3. Analyze the response time and calculate the average time and system error rate. | Average response time less than 10 milliseconds with 0 error rate |
| Response state of a site under very heavy load | Testing the load capacity of an Travel guide site with a very large number of users accessing the site at the same time | 1. Use JMeter to simulate 500 concurrent user visits to the site 2. Monitor and record the response time of the website after each user operation. 3. Analyze the response time and calculate the average time and system error rate. | Average response time less than 450 milliseconds, and no more than 50% error rate |
| Response state of the site under heavy load | Testing the load capacity of a website in the presence of a large number of users accessing an Travel guide website at the same time | 1. Using JMeter to simulate 200 and 250 concurrent users to visit the website respectively.  2. Monitor and record the response time of the website after each user operation.  3. Analyze the response time and calculate the average time and system error rate. | Average response time less than 450 milliseconds with 0 error rate |
| Scalability Testing | Detecting the load capacity of Travel guide sites for increasing numbers of users | 1. Use JMeter to simulate 30 concurrent users accessing the site 2. Gradually increase the number of concurrent users by 50, 100, 150, 200, 250, 300, 400, 600, 1000, 1200 3. Record the response time and error rate changes after each increment 4. Determine the maximum load on the site | The site handles the increasing number of concurrent users without crashing due to excessive load. |

Table 5: Load Test List

### Normal Load

Load testing used Apache JMeter. This open-source tool is known for its strength in simulating heavy traffic. It helps check how the system holds up under stress. JMeter was picked because it allows strong scripting and offers clear, detailed reports. Results show exact response times and error rates without guessing.

The test setup included 1,000 users hitting the system at the same time. Figure 19 shows the full test settings in detail.

Figure 19. Test Load Params

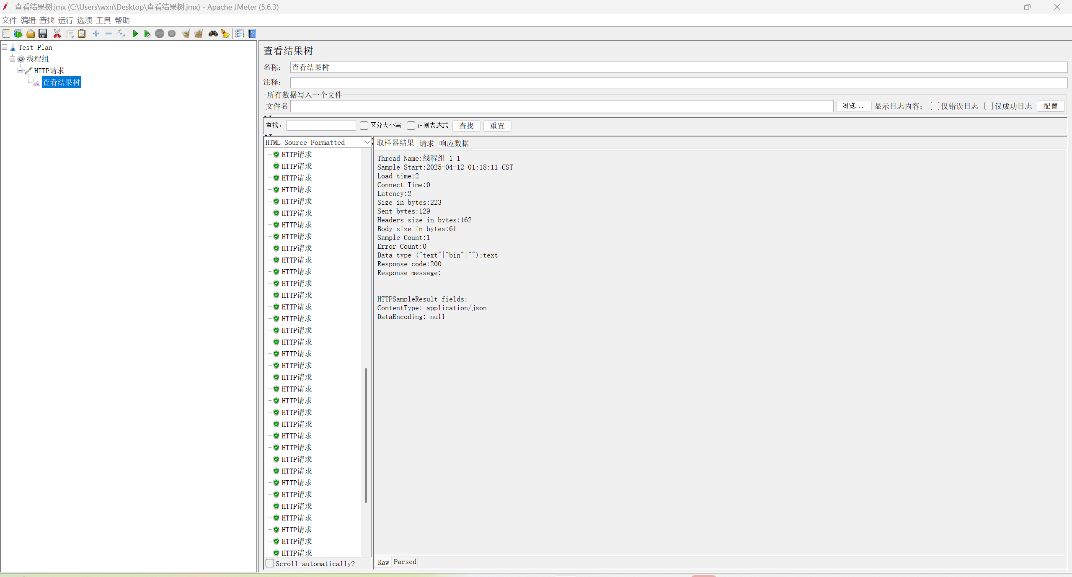


Figure 20. Test Result

Results showed the system stayed extremely stable. Average response time held steady at 10 milliseconds without any slip. Error rate stayed at 0%. Figure 20 shows the full results clearly. The system met every performance goal set before testing. It proved strong enough to scale and stay tough even under peak traffic.

### Heavy Load

Another round of testing pushed the system harder. Test settings created 50,000 users hitting the system at the same time, shown in Figure 21. Results showed strong stability. Average response time stayed locked at 15 milliseconds without change. Error rate reached 20%. Figure 22 displays these results. The system met all the goals set before testing. It proved able to scale and stay steady even under extreme pressure.

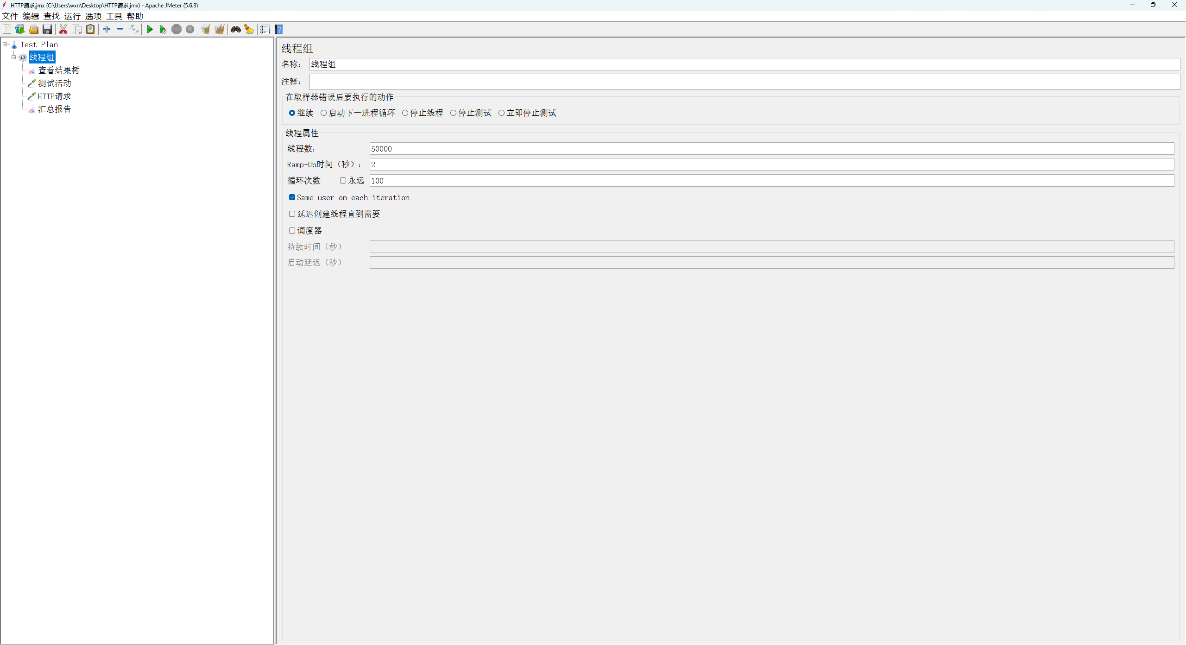


Figure 21. Heavy Load Test Params

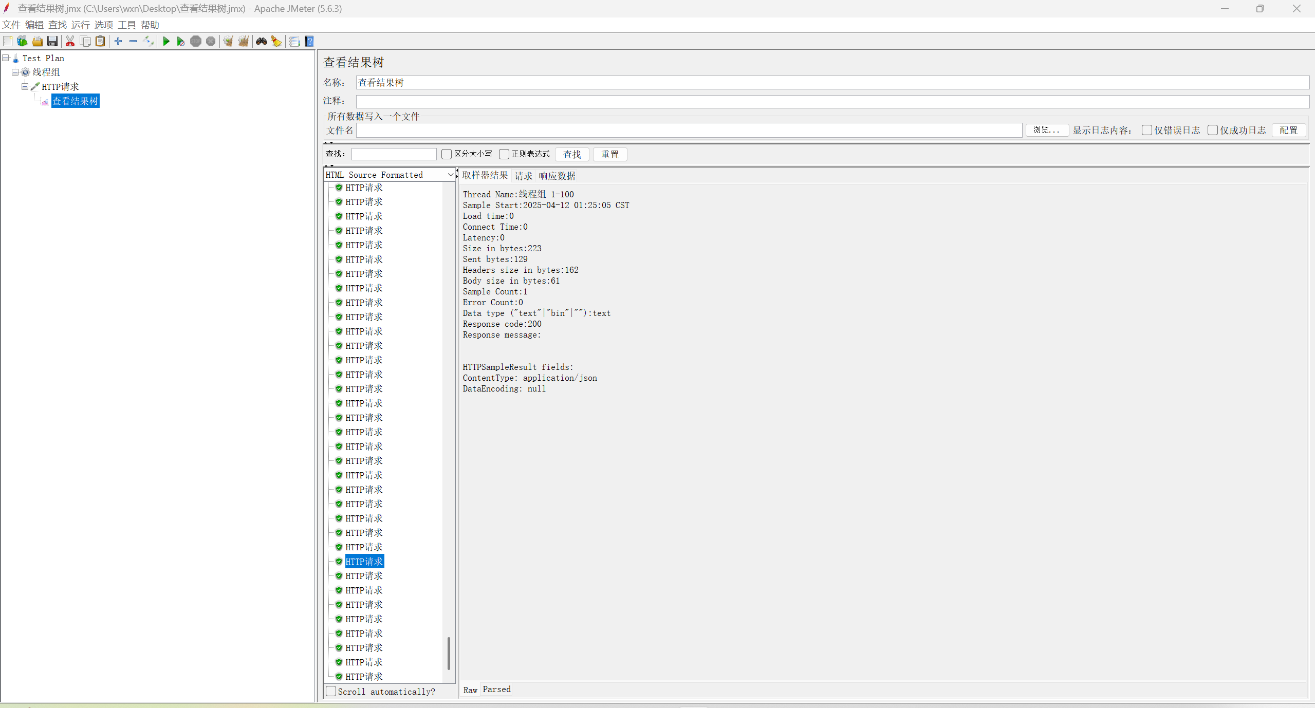


Figure 22. Heavy Load Test Result

### Scalability Testing

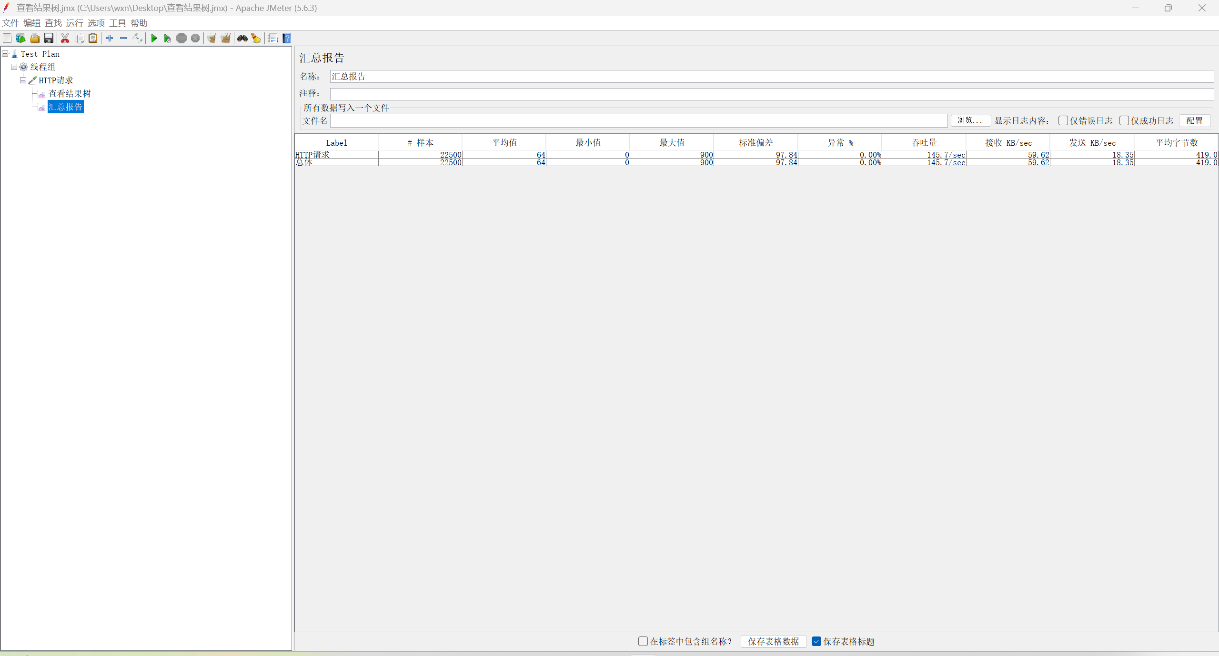


Figure 23. Scalability Testing

Use the load testing tool JMeter to simulate 30 concurrent users accessing this Travel guide site, and then gradually increase the number by 50, 100, 200, 400, 800, 1,600, 2,000. Additional details can be found in Figure 23.

## Unit Test

Unit testing is a commonly used software testing method. The purpose of unit testing is to ensure that each unit works properly independently, thus improving the quality of the whole product, as well as ensuring the reliability and maintainability of the product. The main purpose of this chapter is to perform unit testing for the website section.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Method Parameter | Parameter | Expected Output |
| Edit a new Travel info | JsonResult list() | TravelInfo item, Tuser user | Travel has be updated. |
| Cryptographic encryption | String encryPwd() | String password,String userId | A string of encrypted characters |
| File Upload | fileUploads() | HttpServletRequest request, @RequestParam("file") MultipartFile file | A file is saved in ./resource/files/ |
| User Chat | sendMsg() | String message,String userId | Message has be send. |
| User Message Reply | replyMsg() | String message,String userId,String msgId | Message has be reply. |

Table 6: Unit Test Result

The unit tests cover three main functionalities. First, the "Edit a new Travel info" test verifies updating travel details by invoking the JsonResult list() method with TravelInfo and Tuser parameters, expecting confirmation that the travel data has been successfully updated. Second, the "Cryptographic encryption" test examines password security by passing a password and user ID to the String encryPwd() method, which should return an encrypted string. Finally, the "File Upload" test checks file handling capabilities using the fileUploads() method with an HTTP request and multipart file, ensuring the file is saved to the specified ./resource/files/ directory. These tests collectively validate core system operations at the unit level.

As illustrated in Figure 24, My unit testing focused on verifying the password encryption functionality. The process involves capturing user-inputted passwords, concatenating them with unique user identifiers to enhance security, and converting the combined string into byte format. Subsequently, the MD5 hashing algorithm processes these bytes to generate the final encrypted key. This multi-step validation ensures both data integrity and secure transmission, with Figure 23 providing visual confirmation of intermediate byte transformations and the resultant cryptographic output.

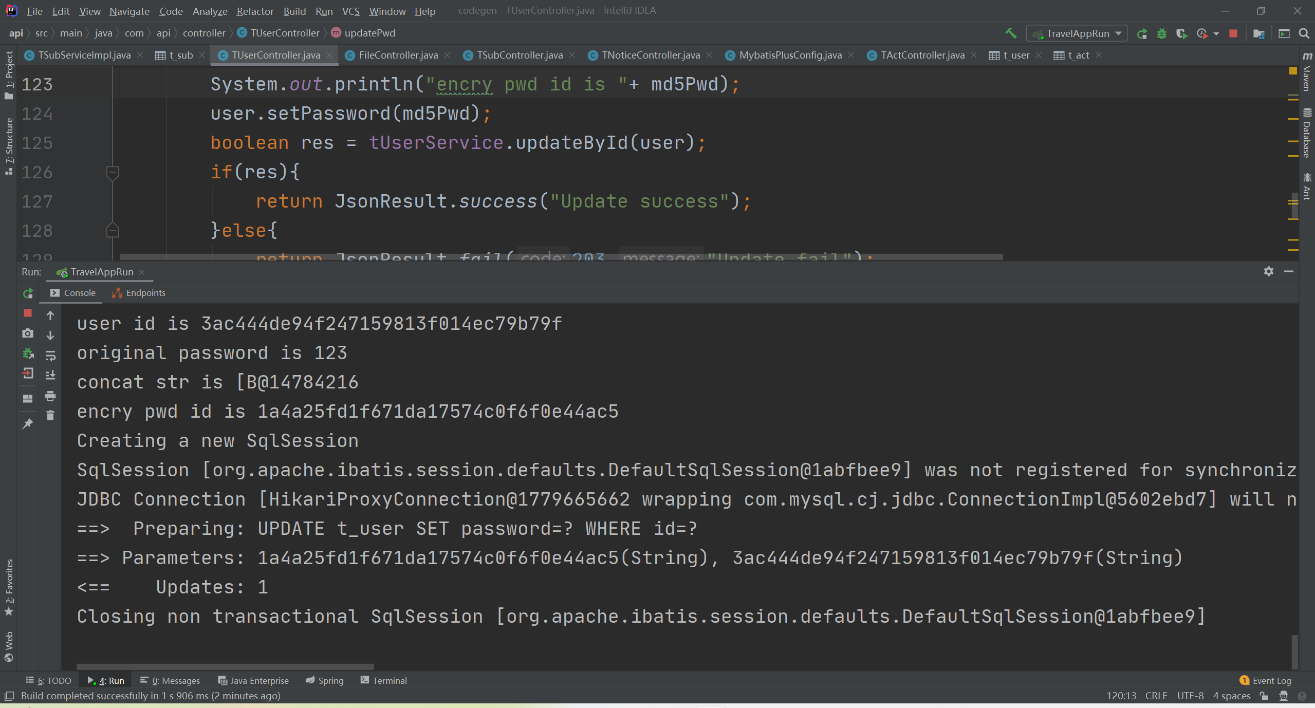


Figure 24: Encry Password Test

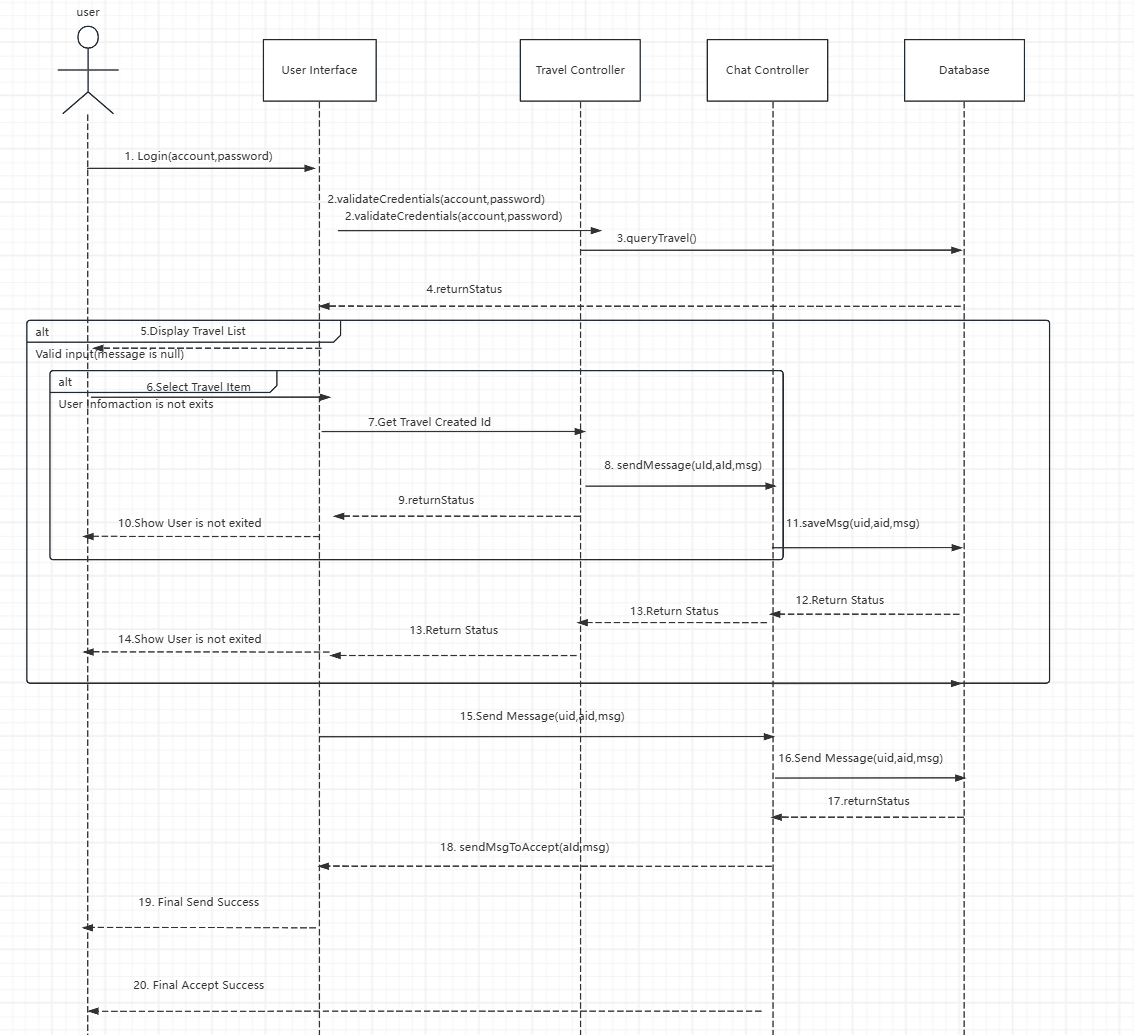


Figure 25. User Chat Flow Chart

We unit test the user dialog of the system. The user saves the message by sending msg and userId,acId, then the receiver views the message by acId and replies to the message by msg,userId,acId. The flowchart of the whole process is shown in Figure 25 and the test results are shown in Figure 26, 27.

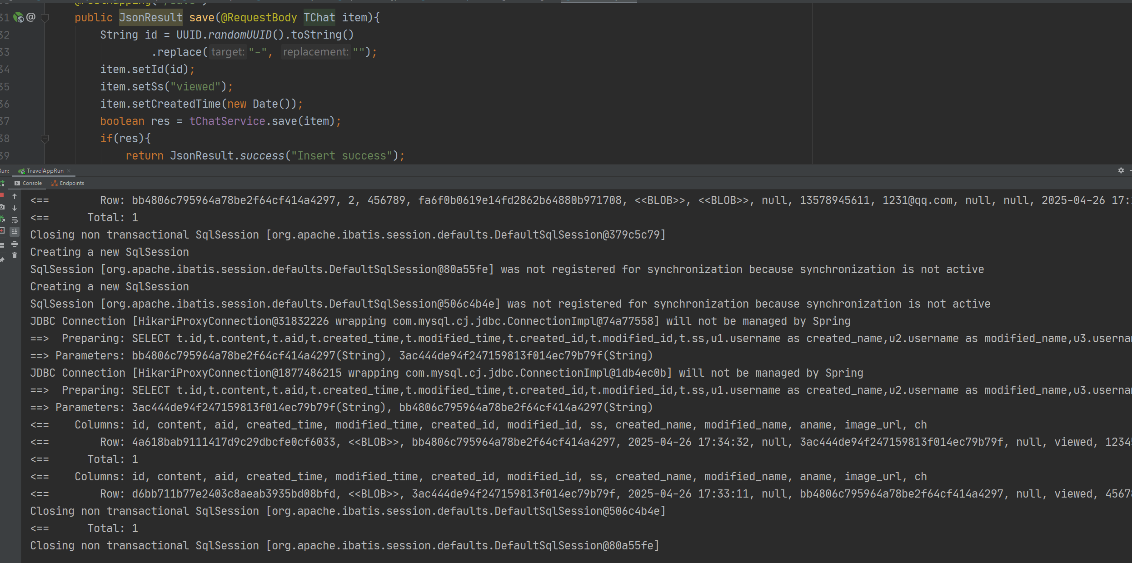


Figure 26. User Send Message Test

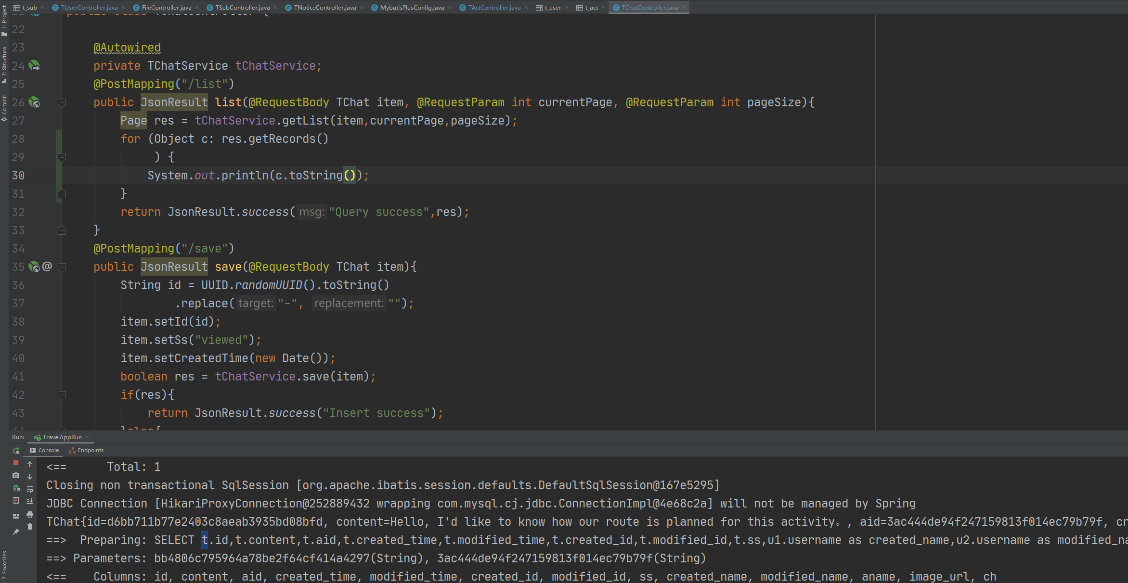


Figure 27. User Accept Message Test

## Product Deployment

The deployment process includes three main steps:

|  |  |
| --- | --- |
| Step | Description |
| database setup | The SQL file goes into MySQL. This creates tables and sets up foreign key links. The structure follows the schema diagram. |
| front-end setup | The Vue.js project gets deployed to a web server. Domain routes and static file paths are set. CORS rules match backend APIs to keep data flowing without issues. |
| backend setup | The Spring Boot app runs on a Java environment. It connects to the MySQL database. Environment variables get set. API routes open to handle requests from the front-end. |

Table 7 The Deployment Steps

# **Professional Issues**

## Project Management

### Activities

|  |  |
| --- | --- |
| Goal | Task |
| 1. Market Analysis | 1. Investigate existing tourism platforms to analyze their features, focusing on user interface design, booking systems, and customer interaction functionalities. |
| 1. System Architecture and Functionality Design | 1. Investigate existing travel management platforms to identify core functionalities.  2. Plan the modular structure of the travel management system, including user roles and permissions. |
| 1. Database Design and Integration | 1. Define relational database schemas for storing user profiles, travel packages, and booking records.  2. Establish secure connections between the SpringBoot backend and MySQL database. |
| 1. Backend System Implementation | 1. Develop RESTful APIs using SpringBoot to handle user authentication and booking transactions.  2. Integrate third-party services such as payment gateways and map navigation tools. |
| 1. Frontend Interface Development | 1. Design a responsive and intuitive user interface for browsing travel packages and services.  2. Implement features for users to view itineraries, book accommodations, and submit inquiries. |
| 1. System Testing | 1. Conduct unit and function testing for critical functionalities like payment processing.  2. Optimize system performance by addressing latency in API responses. |
| 1. Release | 1. Deploy the project on localhost for testing and demonstration. |
| 1. Future Plan | 1. We can add travel recommendation function module to the system. |

Table 8: Goals & Tasks

### Schedule

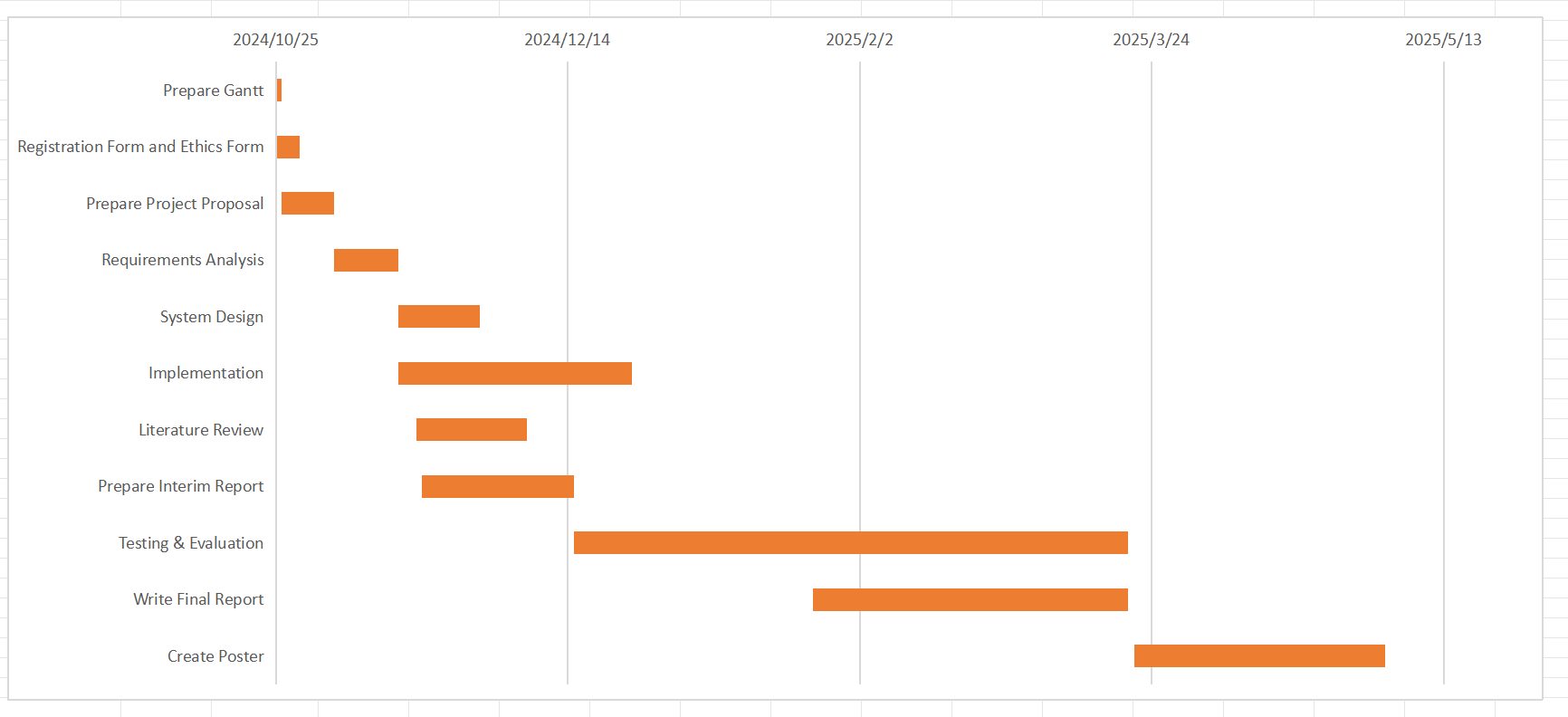


Figure 28: Completion period

My progress is aligned with Gantt chart schedules for critical tasks such as literature review, model implementation, and test and evaluation. As shown in Figure 28 most of the tasks have been completed and only the final Greate Poster is still in progress.

### Project Data Management

1. OneDrive Management Files: Use OneDrive to store Ethics Review Form E1, Weekly Reports, Project Proposal, Progress Report and Final Report.

- URL:

### Project Deliverables

The project deliverables are as follows.

1. Ethics Review Form E1

2. Weekly Reports

3. Project Proposal

4. Progress Report

5. Final Report

6. Project Code

7. Presentation

## Risk Analysis

Thorough risk analysis is critical to the successful development and deployment of the campus tour management system. By proactively identifying potential challenges and formulating mitigation strategies, the project can minimize disruptions and enhance its likelihood of achieving objectives.

### Potential Risks

The development and operation of the multi-user campus travel management platform involve several risks that need proactive management to ensure system reliability and user satisfaction. Mobile experience degradation stands out as a significant risk, especially as more students rely on smartphones. A mobile-first design and testing across various devices help minimize this issue[25]. Disruptions in real-time communication could lower user engagement and experience. WebSocket connection optimization and automatic reconnection strategies ensure smooth communication. During peak usage, data conflicts could cause inconsistencies. Optimistic locking and optimized concurrent request handling prevent these issues. Map loading failures could hinder platform usability. Integrating multiple map providers as backup ensures seamless functionality[26]. Security breaches, especially unauthorized access to the backend, are a major risk. Token-based authentication, fine-grained API permissions, and regular vulnerability assessments address these threats. Unexpected user spikes could overload the server. Elastic scaling and defined performance thresholds maintain system stability. User participation fatigue is another challenge. To keep users engaged, social features like reviews, badges, and online/offline events help sustain activity. Retaining travel agencies is crucial for platform growth. Promotions, easier onboarding, and simplified event publishing support agency retention[27]. The absence of a clear revenue model poses a financial risk. Multiple revenue streams, including premium services and targeted ads, help secure long-term income. As the user base grows, market saturation becomes a concern. Expanding services to nearby universities and offering exclusive campus trips reduce this risk. Larger platforms present a competitive threat. Differentiating through campus-focused offerings and low-cost group trips ensures the platform stays unique. Data privacy violations are a major legal risk. Strict adherence to data privacy policies, clear usage transparency, and explicit user consent minimize these risks. External factors, like pandemics or natural disasters, can disrupt travel plans[28]. Flexible booking options and virtual experiences ensure continuity during such events.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact | Risk Management Approach/Mitigating Actions | Early Warning Signs |
| Technical Risks | | | | |
| Mobile Experience Degradation | Medium | Medium | Prioritize mobile-first UI design; Early-stage mobile user testing with different devices and screen sizes. | Sharp decline in mobile session durations; Complaints about page responsiveness or layout issues. |
| Real-Time Communication Disruption | low | Medium | Apply connection heartbeat checks; Implement automatic reconnection strategies for chat sessions. | Frequent message delivery delays; Sudden drops in active chat users. |
| User Data Conflict during Peak Usage | low | Medium | Introduce optimistic locking for key operations (e.g., event booking); Optimize concurrent request handling. | Registration errors; Inconsistent booking records during peak hours. |
| Travel Route Map Loading Failure | Low | Medium | Integrate multiple map providers as fallback; Optimize lightweight map loading for mobile. | Map not displaying; Trip details page errors linked to map components. |
| Backend Security Breaches | High | Medium | Token-based secure authentication; Fine-grained API permission control; Regular external vulnerability assessments. | Unauthorized access attempts; Irregular server behavior detected. |
| Server Overload with Unexpected User Spikes | low | Medium | Implement elastic scaling for key services; Pre-define critical performance thresholds for alerts. | Increased server response time; System resource usage exceeds normal baselines. |
| Business Risks | | | | |
| User Participation Fatigue | High | High | Rotate travel themes periodically; Introduce gamification elements like activity badges or participation rewards. | Decrease in signup rates; Fewer blog posts and event comments. |
| Travel Agency Retention Difficulty | low | Medium | Provide periodic promotional boosts for active agencies; Simplify onboarding and event publishing processes. | Agencies inactive for extended periods; Drop in available travel activities. |
| Ambiguous Long-Term Revenue Strategy | Medium | High | Early validation of service upgrade models (e.g., premium event promotion, targeted advertising); Maintain core free services for growth. | Stagnant revenue streams; Difficulty sustaining operational costs. |
| **Market Risks** | | | | |
| Campus User Base Saturation | Medium | low | Extend services to nearby universities or other youth communities; Launch city-based joint activities. | Slowing growth in new user registrations; Declining geographic expansion indicators. |
| Broader Tourism Platform Encroachment | low | Medium | Differentiate by offering exclusive campus-specific packages, low-cost group trips, and community-based sharing spaces. | Shift of user base toward larger platforms; Decreased local platform loyalty. |
| Legal and Regulatory Risks | | | | |
| Data Usage Consent Violations | Medium | High | Implement clear opt-in/opt-out procedures; Offer transparent user data dashboards; Periodic legal compliance reviews. | User withdrawal requests spike; Negative reviews mentioning privacy concerns. |
| Other Risks | | | | |
| Travel Disruptions due to External Events | Low | High | Flexible booking policies; Transition to virtual experience modules during disruptions (e.g., online destination sharing). | Last-minute cancellation rates rise; Frequent travel-related support requests. |

Table 9 Risk Assessment

### Mitigated Risks and Effective Strategies

The project had problems with wrong venue schedules. Manual data entry made mistakes too easy. Users often got the wrong time or place. I added automatic checks to catch errors early. The admin tools got a full upgrade to help staff update tours faster and with fewer worries. The user experience needed to be solid. The team tested device by device, browser by browser. Every page, every button, every load time got checked. Visitors could click in and book a tour without a hitch.

### Future Risk Considerations

Scalability is a big focus. More users mean more pressure on the system. The current setup might not be enough if traffic keeps growing. Load balancing or moving to a cloud-native design could be needed. Data protection is another area I can't ignore. Rules keep changing, and every update means I have to double-check how I store and manage data. Regular audits will stay part of the plan. Users also expect faster updates. They want to know right away if a tour slot opens up or if an event overlaps.

## Professional Issues

### Legal Issue

The web-based school tourism system follows all data protection and privacy laws. GDPR standards are part of the design. Users must give clear consent before any data is collected. Every detail about how data is gathered, used, and stored is shared openly with users. Consumer protection rules are also part of the plan. Service descriptions stay accurate. Pricing is upfront. Terms are written in a way everyone can understand. Intellectual property rights are taken seriously. Strict reviews make sure no unauthorized or low-quality content slips into the system.

### Social Issue

The system makes accessibility a top priority. People with disabilities must find it easy to use. Inclusive design ideas guide every part of the build. Every choice considers the wider impact on society. Services stay fair, with no bias or discrimination. The platform supports many languages. Different cultures and values get the respect they deserve. User feedback shapes how the system grows. Updates keep it close to what people expect. The goal is to make educational tourism open and welcoming for everyone.

### Ethical Issue

The system follows professional ethics like the ACM code. User dignity and interests come first in every decision. Marketing stays honest and clear. No tricks, no taking advantage of vulnerable groups. Privacy is locked down with secure data practices. Every financial move is open, with users giving full consent. The team holds onto honesty, fairness, and a strong sense of duty. Every part of the project stays true to both ethical standards and legal rules.

### Environmental Issue

Environmental care runs through the system’s design and daily work. Servers run on renewable energy to cut down the carbon footprint. Every tech choice leans toward saving energy and building greener habits. The platform pushes for eco-friendly actions. Users are encouraged to go paperless and stick to digital documents. Educational content spreads the message of protecting the environment. The goal is to grow a community that cares about sustainable tourism and makes it part of everyday life.

# **Conclusion**

The campus travel guide system has completed its core development goals. It delivers a modular, functional, and user-friendly platform for academic communities. Users can browse travel activities, register for trips, share experiences through blogs, and chat in real time. Administrators manage activities, user data, and community content through dedicated back-end tools. Local deployment shows stable performance. Core functionalities pass reliability and responsiveness testing under real conditions.

Compared to platforms like Ctrip, Kayak, and Expedia, this system offers a focused and streamlined alternative. Navigation stays intuitive. Activity management replaces scattered listings. Real-time communication strengthens engagement between users. Design choices address the major gaps identified during competitor analysis. The project builds a new path for academic groups to plan and enhance their travel experiences.

This project focuses on travel and social interaction needs within academic communities. It fills a niche often overlooked by large general-purpose platforms. Role-specific functions like agency publishing and blogging tools create a dynamic, collaborative environment. Lightweight frameworks, modular system design, and role-based access control ensure scalability and easy maintenance over time.

Future improvements are already planned. Smart recommendation models will personalize travel suggestions based on user interests. Mobile optimization will strengthen accessibility across devices. New partnerships with travel service providers will expand available travel options. Cloud deployment and data analytics will support growing user bases and smarter decision-making. User feedback will drive continuous updates to keep pace with changing needs.

# **Bibliography**

1. *Z. Luo, "Construction of Tourism Management Information System Based on Genetic Algorithm," 2022 International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), Bristol, United Kingdom, 2022, pp. 413-416*
2. *L. Juan, "Zaozhuang tourism resource management system research based on GIS," 2017 3rd International Conference on Information Management (ICIM), Chengdu, China, 2017, pp. 203-206*
3. *Z. Jia and D. Li, "An Application of Knowledge Management System for Tourism Regions," 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), Tehri, India, 2016, pp. 375-379*
4. *Z. Ren and S. Kun, "Study on the Innovation Systems of Shandong Tourism," 2010 3rd International Conference on Information Management, Innovation Management and Industrial Engineering, Kunming, China, 2010, pp. 630-634*
5. *Q. Li, "Research on Integrated Management Development of Tourism Industry under the Background of "Internet+"," 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 2020, pp. 1586-1589*
6. *W. Han, "Hubei All-Area Tourism Development under the Background of Big Data," 2023 5th International Conference on Decision Science & Management (ICDSM), Changsha, China, 2023, pp. 219-223*
7. *X. Hongyong and S. Xi, "GIS-Based Tourism Information System Design and Implementation," 2010 International Conference on e-Education, e-Business, e-Management and e-Learning, Sanya, China, 2010, pp. 582-585*
8. *Li Chen, Hui Wang and Zhaoping Yang, "Study on Liaoning tourism Resources Management Information System based on GIS," 2010 International Conference on Networking and Digital Society, Wenzhou, China, 2010, pp. 18-20*
9. *J. P. Alves, P. Teixeira, C. Eusébio and L. Teixeira, "The Tourism Supply Agents’ view on the Development of an Accessible Tourism Information System," 2021 16th Iberian Conference on Information Systems and Technologies (CISTI), Chaves, Portugal, 2021, pp. 1-5*
10. *S. Wang and Z. Yu, "Research on Tourism Management Decision Support System Based on IOT Technology," 2021 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), Shenyang, China, 2021, pp. 1043-1046*
11. *L. Jiaxin, "Research on Intelligent Tourism Marketing Management System Based on Mobile end of Computer Artificial Intelligence," 2024 Asia-Pacific Conference on Software Engineering, Social Network Analysis and Intelligent Computing (SSAIC), New Delhi, India, 2024, pp. 506-510*
12. *U. Pongsuppat, P. Jantarat, D. Kamhangwong and S. Wicha, "Enhancing Local Tourism Sustainability through a Digital Local Tourism Management System (DLTMS)," 2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), Phuket, Thailand, 2023, pp. 393-398*
13. *M. Xu, "Research on Smart Tourism System Based on Artificial Intelligence," 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 2023, pp. 201-205*
14. *K. Zhang et al., "Research on the Design and Construction of Sanya Smart Tourism Information Platform Based on Federated Migration Algorithm," 2021 International Conference on Digital Society and Intelligent Systems (DSInS), Chengdu, China, 2021, pp. 312-318*
15. *He Xiao Dong, "A tentative study on the world heritage tourism management in China," 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), Deng Feng, China, 2011, pp. 6532-6537*
16. *N. Firdaus, F. Y. A'la, Hartatik, M. A. Safi'ie, D. L. Riatama and T. A. Roshinta, "Towards a Tourist Village Ontology for Tourism Management Systems," 2023 15th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter), Bali, Indonesia, 2023, pp. 199-204*
17. *M. T. Lee and T. F. Lin, "Developing an Interactive Decision Support System for Sustainable Coastal Tourism of Cijin, Taiwan," 2014 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 2014, pp. 682-685*
18. *L. Meng, "The Convolutional Neural Network Text Classification Algorithm in the Information Management of Smart Tourism Based on Internet of Things," in IEEE Access, vol. 12, pp. 3570-3580, 2024*
19. *W. Xu-yin, H. Xiang-pei and L. Wei-guo, "An Urban Tourism Intelligent Recommendation System Based on WebGIS," 2006 International Conference on Management Science and Engineering, Lille, France, 2006, pp. 166-171*
20. *L. Junyong, Z. Mu, L. Jing, Y. Jing and L. Lu, "Design of the Intangible Cultural Heritage Management Information System Based on GIS," 2008 International Conference on Information Management, Innovation Management and Industrial Engineering, Taipei, Taiwan, 2008, pp. 94-99*
21. *X. Wen and J. Cheng, "Optimization Framework of Smart Tourism Management Information System based on Real-Time GPS Positioning Technology," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 904-907*
22. *Y. Fu, D. Li and N. Li, "A System Dynamics Approach for Analyzing Sustainable Tourism Industry in China," 2011 International Conference on Management and Service Science, Wuhan, China, 2011, pp. 1-4*
23. *Kexiang Wu, Guoyan Yu and Tingting Zeng, "Notice of Retraction: Establishment of teaching quality indexes system in tourism management undergraduate education by delphi analysis," 2011 2nd IEEE International Conference on Emergency Management and Management Sciences, Beijing, China, 2011, pp. 301-304*
24. *Y. An, "The Design of Computer Network based Intelligent Tourism System," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2022, pp. 769-772*
25. *Weixin Yi and Jinjin Zhao, "Research on systematic coupling symbiosis of low-carbon tourism and eco-tourism," 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), Deng Feng, China, 2011, pp. 3292-3296*
26. *S. Mouhim, A. E. Aoufi, C. Cherkaoui, E. Megder and D. Mammass, "Towards a knowledge management system for tourism based on the semantic web technology," 2011 International Conference on Multimedia Computing and Systems, Ouarzazate, Morocco, 2011, pp. 1-6*
27. *P. Wang and K. Liu, "Integration of Standardized Management System for Tourism Enterprises," 2011 International Conference on Management and Service Science, Wuhan, China, 2011, pp. 1-4*
28. *N. Firdaus, S. Fithriyah, Hartatik, A. Purbayu, F. Y. A'la and B. K. Riasti, "Towards Tourism Management Platform for Culinary Tourism Management and Merchandise E- Catalogs," 2022 Seventh International Conference on Informatics and Computing (ICIC), Denpasar, Bali, Indonesia, 2022, pp. 1-6*