

Design and Implementation of a Group Tour Information System for South of the Clouds Based on the SSM Framework

Ziang Fu *

Hubei University of Technology, Wuhan, Hubei, 430068, China

* Corresponding author Email: fuziangonline@gmail.com

Abstract: With the innovation of Internet technology, the tourism consumption pattern has ushered in a digital transformation. This research focuses on the design and development of the Caiyun Tourism Southern Grouping System, which is built upon the SSM (Spring, Spring MVC, MyBatis) framework. Through intelligent grouping and resource integration, an innovative platform of "tourism service+social interaction+e-commerce" is constructed. The core function of the system covers the whole process management of group assembling. Users can customize travel time, route preference and other parameters. The system relies on algorithms to achieve accurate matching of travelers, and improves the efficiency of group assembling through real-time progress push, so as to improve the success rate of group assembling in the traditional mode. The platform innovatively integrates tourism socialization and consumption scenarios: build a user-generated content (UGC) travel strategy community, and gather 3,000+high-quality travel notes to form a structured knowledge base; Embedded in the destination characteristic commodity e-commerce module, the consumption closed loop of "tourism experience-commodity purchase" is opened, and the value of tourism industry chain is extended. On the technical realization level, the advantages of SSM framework are brought into full play by adopting layered architecture design: Spring manages business components through IoC container, and AOP mechanism realizes unified log processing; SpringMVC adopts RESTful interface to improve the efficiency of frontend cooperation; The design of MyBatis dynamic SQL and secondary cache improves the performance of database query.

Keywords: SSM Framework; Tourism Group-booking; Information System; System Design; System Implementation.

1. Introduction

1.1. Research Background and Significance

With the rise in residents' consumption level and the evolution of their consumption concepts, tourism demand has shifted from standardized products towards diversified and personalized experiences. Tourists now prefer exploring niche destinations, immersing themselves deeply in local culture, or customizing personalized itineraries.

The South of the Clouds Group Tour System responds precisely to these trends. By leveraging digital means, it integrates tourism resources across Yunnan province, overcomes the limitations of traditional modes such as resource fragmentation and high costs. It packages resources such as lesser-known attractions and distinctive homestays into cost-effective group tour products, enabling tourists to enjoy personalized travel services at lower costs. The system achieves comprehensive and systematic integration of previously scattered and fragmented tourism resources, optimized their allocation across a broader scope[1].

The emergence and development of the system play a crucial role in driving innovation and revolution within the entire tourism industry. It opens up entirely new market spaces and business growth opportunities for tourism enterprises. By consolidating resources, it reduces operational costs and enhances resource utilization efficiency, thereby promoting the upgrading and structural optimization of the tourism sector[2].

1.2. Research Status at Home and Abroad

1.2.1. International Research Status

Internationally, tourism e-commerce is relatively mature,

and group tour modes are widely popularized[3]. Leading platforms such as Expedia and TripAdvisor have long established their presence in the group tour sector[4]. Through deep collaborations with tourism suppliers worldwide, they offer users a diverse range of products—from premium luxury tours to budget backpacker tours[5]. However, due to cultural differences and regional constraints, these platforms still fall short in addressing highly specific user demands and providing comprehensive localized services.

1.2.2. Domestic Research Status

In recent years, following the introduction of the social e-commerce concept into China, major players in the online travel industry have launched group tour products[6]. Leading Online Travel Agencies (OTAs) like Ctrip and Fliggy have devoted substantial resources to group tour operations, including destination-based grouping and theme-based grouping[7]. Nevertheless, the overall sophistication of domestic online travel platforms lags behind their international counterparts[8]. Additionally, the domestic group tour market faces several challenges that are in urge need for standardization and improvement. For example: high homogeneity of marketed products, severe information asymmetry, inadequate standardization and insufficient quality assurance mechanisms for services etc.[7][8]

1.3. Research Objectives

This study proposes the development of the South of the Clouds Group Tour Platform based on the SSM framework, with the following core objectives:

(1) To design and implement a comprehensive and user-friendly group tour platform that provides tourists with:

Access to domestic and international popular tourism destinations; Curated travel itineraries, hotel bookings, and attraction tickets; Multiple grouping modes with customizable grouping rules.

(2) To establish a dynamic tourism social platform facilitating: Interactive communication among tourists; Sharing of travel experiences and tourist guides; Exchange of group tour information; Formation of travel companion groups.

(3) To develop a robust backend management system enabling administrators to perform comprehensive overall management and monitoring of tourism resources, users, orders, and group activities.

2. System Requirements Analysis

2.1. Functional Requirements

The system functionality is illustrated in Figure 1.

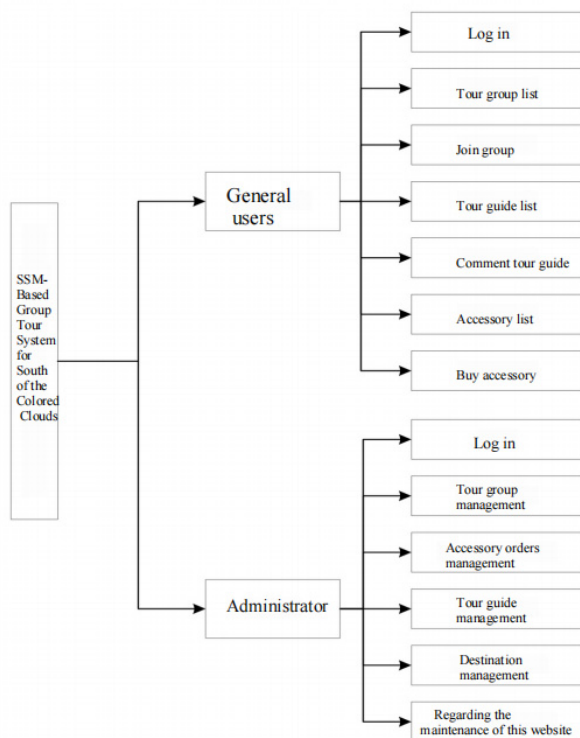


Fig 1. Overall system function diagram

2.2. Group Tour Functional Requirements

(1)Initiate Group Tour: Users can initiate group tours after selecting preferred travel products by providing group size, participation deadline, group title and other relevant details.

(2)Join Group Tour:Users can browse existing group tours in the activity list and join those of interest. The system shall validate: user eligibility for participation, whether the maximum group size has been reached etc.

2.3. Travel Guide Functional Requirements

(1)Edit Travel Guides: Supports publishing multimodal travel guides using texts, images and videos. When creating guides, users must specify destination, travel period, guide title and content body etc.

(2)Browse Travel Guides: Users can explore published guides through a categorized guide repository.

2.4. Administrator Functional Requirements

(1) User Information Management: Administrators can

view, edit, and delete registered user data such as: username, password, contact details, profiles etc. Administrators can also perform user classification e.g., regular users, verified users, restricted users etc., then assign differentiated permissions and operational restrictions to different user classifications in order to maintain the security and order of the platform[11].

(2) Product Review Management: Responsible for verifying travel products provided by suppliers, ensuring accuracy and authenticity of product descriptions, compliance with regulations, completeness etc. Product details such as product type, description, schedule, pricing, accommodation, transportation requires multi-tier verification protocols.

(3) Group Activity Supervision: Monitors all group tour activities with capabilities to track progress such as initiation status, participant count, remaining time etc. To detect anomalies such as fraudulent bookings, false advertising etc. And to intervene in violations via real-time administrative controls.

3. System Design

3.1. Hierarchical Architecture Design

The South of the Clouds Group Tour System adopts a layered architecture comprising four tiers: Presentation Layer, Business Logic Layer, Data Access Layer and Data Storage Tier.

Presentation Layer serves as the user-system interaction interface, it is responsible for accepting user requests, rendering system responses, implementing responsive UIs via HTML5, CSS3, and JavaScript. Using HTML for page structure, CSS for styling, JavaScript for interaction logic and Semantic UI framework for aesthetic layouts and enhanced user experience.

Business Logic Layer works as system core to processes critical business operations including:(1) Tour Product Management, handles search requests, filters product information based on requests, then sort and return it to user. (2) Group Tour Processing, validates user eligibility for initiating or joining groups, enforces grouping rules, calculating dynamic pricing and updates participant quotas. (3) Travel Guide Management, reviews user-submitted guides, processes view/like/comment operations and synchronizes data with database. (4) User Management, handles operation logic such as registration, login, profile updates etc., meanwhile validates input integrity and implements encrypted credential storage.

Data Access Layer manages database connectivity and persistence operations, provides add, delete, modify and query functions for tour products, users, groups, and guides layers. This layer utilizes the MyBatis framework to map JavaBeans to database tables through XML mapping files or annotations, enabling efficient database access.

Data Storage Tier utilizes MySQL, a relational database management system, as the storage solution. It enables the centralized storage of information within the system. The database schema, including appropriate table structures, is meticulously designed based on system requirements. Furthermore, relevant indexes and constraints are implemented to ensure data integrity, consistency, and efficient storage performance.

3.2. Database Design

The Entity-Relationship Diagram (ERD) is a fundamental

tool for database design and information system modeling. It plays a crucial role throughout the entire lifecycle of the system, from initial business requirements analysis to development and subsequent maintenance. Serving as both

the blueprint for the database schema and the foundational "genetic map" of the system, the ERD effectively captures the core data structure and relationships. The specific ER diagram for this project is presented in Figure 2.

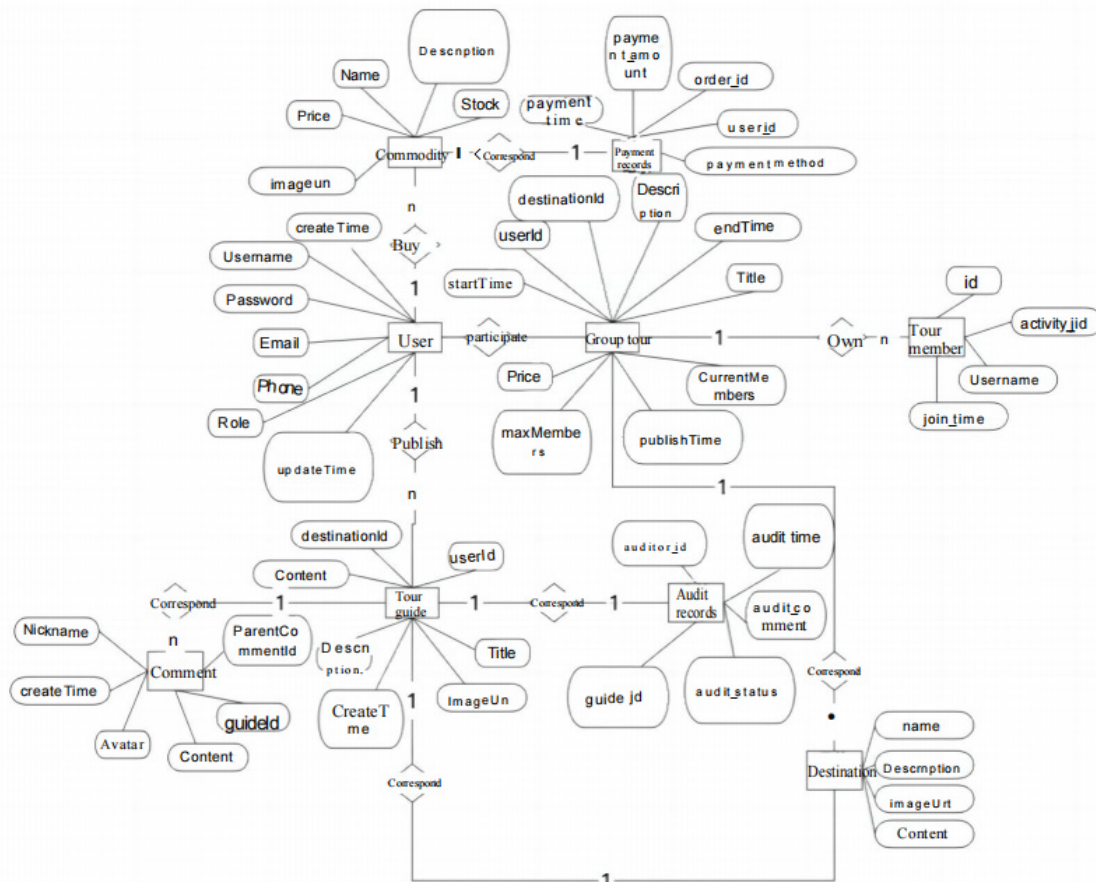


Fig 2. System ER diagram

The primary database tables are presented in Figures 1, 2, and 3.

Table 1. Group tour database

Field Name	Field Type	Description
id	int	Group tour ID
destinationId	Int	Destination ID
Title	vchar	Group tour title
Description	vchar	Group tour introduction
Price	vchar	Pricing
maxMembers	Int	Max group member quantity
CurrentMembers	Int	Current applicants quantity
startTime	Date	Start time
endTime	Date	End time
publishTime	Date	Publish time
userId	Int	Publisher ID

4. System Implementation

4.1. Entity Layer Coding Implementation

Within the entity layer, corresponding Java entity classes are created based on the system database design. For instance, the attributes of the User entity class directly correspond one-to-one with the fields in the database's user table. Taking the TravelProduct entity class as an example, in addition to its

basic attributes, it may also include associations with other entity classes. For example, a private `List<GroupBuying>` groupBuyings; declaration signifies the list of group buying activities associated with that tourism product. This approach establishes object-relational mapping (ORM) between entity classes, facilitating data processing and manipulation within the business logic layer[12].

Table 2. Tour guide database

Field Name	Field Type	Description
id	int	Tour guide ID
destinationId	Int	Destination ID
Title	varchar	Tour guide title
Description	varchar	Tour guide introduction
Content	varchar	Tour guide content
CreateTime	Date	Start time
userId	Int	Publisher ID
ImageUrl	varchar	Tour guide Diagram

Table 3. User database

Field Name	Field Type	Description
id	int	User ID
Username	varchar	User name
Password	varchar	User password
Email	varchar	User email
Phone	varchar	User cellphone number
Role	varchar	Character
createTime	Date	Create time
updateTime	Date	Update time

```
public class User {

    private int id;           // User ID
    private String username; // User name
    private String password; // Password
    private String email;    // Email
    private String phone;    // Phone number
    private String role;     // User character
    private Date createTime; // Create time
    private Date updateTime; // Update time

    // Getters and Setters
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

4.2. Data Access Coding Implementation

```
public interface UserMapper {

    // Add user
    void addUser(User user);

    // Update user information
    void updateUser(User user);
}
```

The Data Access Layer utilizes MyBatis to establish

mappings between entity classes and database tables. For instance, the UserMapper interface defines operations for user data manipulation, with field correspondences to the User entity class configured via annotations or XML files[13]. Data access methods are declared within Mapper interfaces; for example, the UserMapper may contain User selectById(Long userId); to retrieve user information by ID.

To eliminate duplicate data, techniques such as adding unique constraints in SQL statements or using the DISTINCT keyword are employed. The implementation of methods in Mapper interfaces is achieved by writing corresponding SQL statements. For example, the getUserById method in UserMapper would be implemented with SQL statements: SELECT * FROM user WHERE id = {id}.

During implementation, special attention is paid to SQL statement optimization, including strategic index creation, to enhance the efficiency of data querying and manipulation operations.

```
public interface UserMapper {

    // Add user
    void addUser(User user);

    // Update user information
    void updateUser(User user);
}
```

4.3. Data Business Layer Coding Implementation

```
public interface UserService {
```

```
// Get all users
List<User> getAllUsers();
User      getUserByUsernameAndPassword(String
username, String password);
}
```

Data Business Layer functions as the critical hub for core system logic processing, it primarily undertakes responsibilities for handling business rules and workflows. The UserService interface defines a series of business methods related to user operations, such as User getId(int userId); for retrieving specific user information by user ID.

Within the UserServiceImpl implementation class, the dependency injection feature of the Spring framework is leveraged to inject the UserMapper interface from the data access layer, thereby enabling concrete business logic implementation. For instance, in the getUserByUsernameAndPassword method, username and password inputs from the frontend are received, then the select method of UserMapper is invoked to query user information in the database, then the corresponding User object is returned based on the query result.

4.4. Implementation of the Data Control Layer

Built upon the Spring MVC framework, the data control layer primarily handles frontend request reception, invocation of business logic layer methods, and response delivery. Corresponding controller classes e.g., UserController, TravelProductController are created. Within these controllers, request-handling methods are defined using annotations. For instance: @RequestMapping("/user/login") public String login(User user, Model model) {... } is a method processes user login requests, where the User object parameter captures frontend-submitted credentials (username/password), and processing results are stored in the Model object for frontend view rendering.

Controller methods delegate business processing by invoking service interfaces from the business logic layer. For example, the login method in UserController calls UserService.login() for authentication, sets prompt messages based on the verification outcome, and redirects to appropriate pages.

Additionally, the data control layer manages request parameter binding (e.g., @RequestParam specifies parameter names/defaults).Data format conversion (e.g., @ResponseBody serializes returns to JSON for AJAX requests).Exception handling (e.g., @ExceptionHandler centralizes error processing, returning user-friendly messages).

Through well-designed implementation, this layer ensures effective frontend-backend interaction and data transfer, enabling accurate responses to diverse user requests while enhancing user experience.

```
@Controller
public class UserController {
    @Autowired
    private UserService userService;

    @GetMapping("/view")
    public String viewUser(@RequestParam("id") int id,
```

```
Model model) {
    User user = userService.getUserById(id);
    System.out.println(user);
    model.addAttribute("user", user);
    return "user/view";
}
}
```

5. System Testing

5.1. Integration Testing

Integration testing is conducted by progressively combining subsystems according to the system architecture and specific business workflows. For example, when a user initiates a tour group activity on the frontend, this operation involves multi-tier interactions spanning the user interface layer, tour group business logic layer, data access layer, and database. The testing verifies each step from inputting tour group details to successful activity creation, ensuring correct execution of all processes, accurate persistence of information at the data layer, and valid data transfer integrity across architectural layers.

5.2. Web Platform Test Report

Functional Test Results: Comprehensive functional testing confirms most modules operate per design specifications. The tourism product display and search functionality accurately presents product information with precise filtering by keywords, destinations, and price ranges, returning qualified product lists meeting all criteria.

Performance Metrics: Under simulated multi-user concurrency scenarios, core performance parameters including response latency, throughput capacity, and concurrent user load thresholds were empirically evaluated. Under normal load conditions (100-500 concurrent users), the system maintains an average response time below 1.2 seconds with rapid page loading and smooth operation flow. When concurrent users exceed 1000, the average response time increases by no more than 3 seconds while throughput remains at high levels.

6. Conclusion and Outlook

Developed and implemented on the SSM (Spring, Spring MVC, MyBatis) framework, the "South of the Colored Clouds" group tour system demonstrates the following functional and technical advantages: Functionally, the system centers on group tour services as its core capability, effectively addressing and enhancing users' practical needs in tourism. Technically, adopting the SSM framework ensures the system aligns with fundamental software application requirements. Specifically, Spring manages system components, dependency injection (DI) decouples functional layers, significantly improving system flexibility, the loosely coupled backend architecture enhances code maintainability and scalability.

Building upon this robust SSM foundation, the system holds strong potential for continuous optimization and expansion. It is positioned to play an increasingly vital role in future tourism markets, delivering enhanced group tour experiences to travelers while propelling Yunnan's tourism

industry to new heights of development.

References

- [1] **Jia, W. Q., Liu, X., & Fu, P.** (2024). Design and implementation of enterprise record management system based on Spring Boot + Vue framework. *Industrial Control Computer*, *37*(10), 151–152.
- [2] **Denise, S. E., & Michael, C. H.** (2025). Integrating smart cities and tourism systems: A critical review. *International Journal of Public Sector Management*, *38*(2), 196–212.
- [3] **Gao, J., Fang, G., & Liu, L.** (2023). The design and implementation of the tracking management system for the post practice of college students. *Advances in Computer and Communication*, *4*(1).
- [4] **Anouar, D., & Seddik, B.** (2017). Design of electronic ticket system for smart tourism. *Advances in Computing*, *7*(1), 4.
- [5] **Kalliontzis, D., & Kotzamanis, V.** (2023). Addressing geometric and material nonlinearities in fluid-structure interaction with the ALE-SSM framework. *Engineering Structures*, *295*, 116851.
- [6] **Piao, L. S., & Chen, P. X.** (2024). Design and implementation of student homework management system based on SSM framework. *Technology Information*, *22*(8), 233–237, 243.
- [7] **Wang, J. G., Zhang, Z., Tang, L. F., et al.** (2024). Design and implementation of student information management system based on SSM framework. *Modern Information Technology*, *8*(7), 30–33.
- [8] **Guo, J. T., Chen, T., & Xiang, Y.** (2024). Design and implementation of a campus dormitory management system based on SpringBoot framework. *Computer Knowledge and Technology*, *20*(7), 37–40.
- [9] **Liu, F. L.** (2023). Design and implementation of tourism big data analysis system based on SSM framework. *China New Technologies and Products*, *21*, 22–25.
- [10] **Zhao, J.** (2023). Enterprise compliance management system based on SSM + VUE framework.
- [11] **Feng, Z. C., Mi, W., & Guo, X. L.** (2022). Design of logistics company information management system based on SSM architecture. *Computer Knowledge and Technology*, *18*(4), 24–26.
- [12] **Lv, N.** (2019). Design of Hexigten tourism destination marketing information system. *Inner Mongolia Science Technology & Economy*, *6*, 69–70, 72.
- [13] **Bai, S., Zhao, Y. N., Tuo, Y. X., et al.** (2018). Design and implementation of tourism platform based on BS architecture. *Computer Knowledge and Technology*, *14*(34), 48–50, 61.