Design of Campus Bicycle Rental Management System Based on SSM Framework

Wanxin Ping¹, Yuankun Chen² and Xiaoyan Hou³*

¹ School of Internet of Things, Hohai University, Changzhou, Jiangsu Province, 213000, China
² School of Information Management, Huazhong University of Science and Technology, Wuhan, Hubei Province, 430000, China
³ School of Internet of Things, Hohai University, Changzhou, Jiangsu Province, 213000, China
*Corresponding author’s e-mail: 15851961186@163.com

Abstract. In view of the chaotic management of bicycle rental in colleges and the high waste rate of resources, the purpose of this paper is designing a bicycle rental management system to make it easier and more efficient for students and leasing businesses to complete bicycle rental. According to the elaboration of the system framework, the design and implementation process of the database, the design concept and use principles of the system are introduced in detail.

1. Introduction
At present, there is no system for bicycle rental businesses in colleges and universities to update and share bicycle information in a timely manner, resulting in low bicycle utilization rate, confusion of rental and income management. [1] This paper aims to design a campus bicycle rental management system to solve this problem eventually. The system helps students efficiently access bicycle information, making it easier for businesses to rent bicycles and manage their income.[2]

In the campus bicycle rental management system, we can search all the rental bicycle information and rental records online and anytime, and can also score the rental experience. The administrator updates the user and bicycle information in a timely manner, improves the rental registration and completed orders, and checks the daily income. Super administrators manage administrator information and master all order and revenue information. In this way, management is more efficient and convenient.

2. Development Technique
The bicycle rental management system adopts the MVC mode, which realizes the separation of models and views, domain logic and database access, improving system flexibility and reusability and development efficiency. For the choice of system framework, we compared the common MVC framework, SSH framework and SSM framework. The decision was made to use the SSM integration framework (the integration of three open source frameworks, Spring MVC + Spring + Mybatis or SSM for short). The SSM integration framework decouples and simplifies development, enabling the design of powerful Web applications. At the same time, it is also a typical MVC framework. [3] It divides the entire system into four layers: the presentation layer (JSP and related JS), the control layer,
the service layer, and the database access layer.

MyBatis is an excellent data persistence framework. It writes the SQL in an xml file for easy management and optimization, avoiding almost all JDBC code and manually setting parameters and getting result sets. It also uses logical tags to control the splicing of dynamic SQL, and the result set of the completed query is automatically mapped with the java object.

![MyBatis Functional Diagram](image)

**Figure 1. MyBatis Functional Diagram**

### 3. Design of Function Module

The overall design of the system consists of five modules: user side, user management, administrator management, bicycle management, lease management and completed order management.

User (administrator) management module includes: user registration, user information modification, and user basic information query. The background administrator (super administrator) has the function of adding users (administrators), querying all users (administrators) and managing users (administrators).

![User Management Module](image)

**Figure 2. User Management Module**

User side module includes: user login registration, editing and adding user information, and completing the function of renting a bicycle.

Bicycle management module includes: bicycle information modification, bicycle basic information inquiry.

![Bicycle Management Module](image)

**Figure 3. Bicycle Management Module**

Rental management module includes: the registration of rental user information, bicycle rental...
information, rental time, deposit, and administrator.[4]

Completed order management module includes: the inquiry of rental information, return time, and revenue.

4. Design of Database

4.1. The design of database structure

The bicycle rental management system operates on bicycles, bicycle rentals, orders, and the like. The analysis abstracts the system into six entities: user, administrator, using, bicycle, rental registration, and completed order. The specific relationships between entities are as follows (enumeration only):

Users can rent multiple bicycles, which are rented by only one user and have a one-to-many relationship.

Bicycles can be managed by multiple administrators, and administrators can operate multiple bicycles in a many-to-many relationship.

A classification can only belong to one large classification, and a large classification can have multiple small classifications. This relationship is one-to-many.

![Database E-R Diagram](image)

Figure 4. Database E-R Diagram

4.2. The design of logical structure

The system factors of the DBMS should be considered during the logic design phase. The process of system logic design is to convert the E-R diagram designed in the conceptual design stage to the logic structure supported by the DBMS system used. This period mainly includes converting the E-R diagram into a relational model, model optimization, database schema definition and user sub-pattern design.

The E-R graph was transformed into the relational model as follows:

- User(user ID, user password, user name, user phone)
- Administrator (administrative ID, administrator password, administrator name, administrator phone)
- Bicycle(plate number, brand, model, rental price, status)
- Rental registration(order number, plate number, user ID, rental time, deposit, handling administrator ID)
- Completed order(order number, plate number, user ID, rental time, return time, total borrowing days, deposit, receivable amount, handling administrator ID)

(Note: the main code is underlined and the external code is marked with a pane)

5. Design of bicycle rental management system
The bicycle rental management system adopts the B/S three-layer mode, which consists of a front end and a background. The front end uses Flask framework + BootStrap + JQuery + HTML + CSS and JS encapsulation. The background is mainly SpringMVC + Mybatis + Spring and Shiro.

![System Administrator Management Process](image)

**Figure 5. System Administrator Management Process**

5.1. **DAO layer**
According to the basic framework of the bicycle rental management system, the DAO layer is a compilation of MyBatis. The DAO layer can be understood as data stored in a database or a hard disk that can be stored for a long time. The device does not disappear in the same way as it is in memory, that is, the data is stored on the persistent device. It is focused on the system. A relatively independent aspect of achieving data persistence.

The separation of data storage logic provides an abstracted data access interface. The separation of the underlying implementation of data access allows you to switch the underlying implementation without modifying the code. The separation of resource management and scheduling enables unified resource scheduling (such as caching mechanism) at the data access layer. Data abstraction that provides more object-oriented data manipulation.

5.2. **Control Layer**
The Control layer (belonging to the springMVC module) is mainly responsible for the specific business module flow control. It controls the business process by calling the interface of the Service layer. The configuration of the control is also in the Spring XML configuration file. There are different controllers for the specific business process.

5.3. **View layer**
View layer (belongs to the springMVC module), responsible for the display of the foreground jsp page, this layer needs to be combined with the Control layer to develop. It sends the request by jsp, the controller receives the request, the processing returns, and the jsp echoes the data.
6. Conclusion
This paper aims to design a campus bicycle rental management system to facilitate students to rent bicycles. Through the research of SSM framework, this paper determines the development of bicycle rental system under SSM framework. From the database design to the implementation description of each logical layer of the entire framework, the process of the entire system is introduced. Finally, the system will implement user management, bicycle management, bicycle rental management, order management and other functions to make it easier for college students to rent bicycles, enabling businesses to manage bicycle rentals more effectively.

Acknowledgments
This work was financially supported by Construction and Operation of campus bicycle rental management for College students' Innovation training Project.

References
[1] Baskaran, S., Shenbaga, V., Thenmozhi, D., & Shanmugapriya, S. Resource sharing platform for collaborative cloud campus. In 2014 International Conference on Computation of Power, Energy, Information and Communication, ICCPEIC 2014 (pp. 473–477).
[2] Hsu C Y, Chen Y C, Luo R C, et al. A Resource-Sharing Platform for Trading Biomedical Intellectual Property[J]. It Professional, 2010, 12(2):42-49.
[3] Zhang D, Wei Z, Yang Y. Research on Lightweight MVC Framework Based on Spring MVC and Mybatis[C] Sixth International Symposium on Computational Intelligence and Design. IEEE Computer Society, 2013:350-353.
[4] Zhang Y, Huang G, Zhang W, et al. Towards module-based automatic partitioning of Java applications[J]. Frontiers of Computer Science, 2012, 6(6):725-740.
[5] Brace W, Cheutet V. A framework to support requirements analysis in engineering design[J]. Journal of Engineering Design, 2012, 23(12):876-904.
[6] Zhang D, Wei Z, Yang Y. Research on Lightweight MVC Framework Based on Spring MVC and Mybatis[C] Sixth International Symposium on Computational Intelligence and Design. IEEE Computer Society, 2013:350–353