Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 8, No. 1, 2024

Design and Implementation of a Student Attendance Management System based on Springboot and Vue Technology

Yixuan Liu *

College of Computer and Software Engineeringng, University of Science and Technology Liaoning, Anshan Liaoning, 114000, China
*** Corresponding author:** Yixuan Liu

Abstract: This paper describes in detail the development process of Student Attendance Management System based on Spring Boot, Vue.js and MySQL. The system aims to provide an efficient and automated solution for recording and managing student attendance to improve the daily management efficiency of educational institutions and reduce the administrative burden of teachers. The system adopts a modularized design, covering functional modules such as user management, student management, teacher management, class and course management, attendance record, leave management, statistical report and system settings. Through the practice of this project, we can have a deeper understanding of the powerful functions of modern Web development technology and its application prospects, and deepen our knowledge of Spring Boot back-end development, Vue.js front-end design and MySQL database operation in practical applications.

Keywords: Spring Boot; Vue.js; MySQL; Modular Design; Web Development.

1. Introduction

1.1. Project Background

With the growing demand for digital management in the education industry, schools have put forward higher requirements for the efficiency and accuracy of student attendance management. The traditional manual sign-in method is not only time-consuming and labor-intensive, but also prone to errors, which seriously affects the quality and efficiency of teaching management. Therefore, it is particularly important to develop an automated and intelligent student attendance management system. The system aims to automate the process of student attendance through modern information technology, reduce manual errors, improve the speed of data processing, and assist schools to better manage student attendance through data analysis.

1.2. Development Purpose

The development objective of this project is to provide an easy, efficient and accurate student attendance management system to improve the current attendance process in school education management. The system can help educational organizations to achieve the following goals:

(1) Improve attendance efficiency: an automated attendance system reduces the need for manual entry, making the attendance process faster and more accurate.

(2) Facilitate data statistics and analysis: the system can automatically generate attendance reports and support dynamic inquiries and statistics, helping teachers and administrators to quickly grasp students' attendance.

(3) Enhance student management: With accurate attendance data, teachers can keep abreast of students' attendance and pay proper attention to and manage absent students.

(4) Supporting decision-making: The data reports and analysis results provided can provide decision-making support for school management to optimize teaching resources and environment.

1.3. Main Technologies Developed

1.3.1. Spring Boot

Spring Boot is an extension to the Spring Framework that simplifies the process of configuring and deploying Spring-based applications [1].Spring Boot is designed to allow developers to get new Spring applications up and running quickly, reducing the complexity of building projects. It provides a wide range of default configurations (the idea that convention trumps configuration) while allowing developers to easily override and adjust these configurations. The core features of Spring Boot include embedded server support (such as Tomcat, Jetty, or Undertow), autoconfiguration, health checking, externalized configurations, and natural support for microservices architectures. This makes it well suited for building enterprise-class applications and microservices.

1.3.2. Vue.js

Vue.js is a lightweight front-end JavaScript framework for building interactive web applications [2]. It provides responsive data binding and compositional view components that enable developers to build modern single-page applications (SPAs) in an intuitive way.Vue.js is designed to implement responsive data binding and compositional view components. Its core library focuses solely on the view layer, which is easy to learn and integrate, but also integrates seamlessly with other libraries or existing projects.Vue.js is also very flexible, supporting everything from simple pages with quick interactions to complex single-page applications.

1.3.3. MySQL

MySQL is a widely used open-source relational database management system (RDBMS) that uses SQL (Structured Query Language) as its data access language [3]. It is based on a client-server model, where the server runs the MySQL software and manages access to the database, and the client performs data operations by sending requests to the server. MySQL is used by many websites and applications around the world, including many large and high-traffic websites. It is

known for its high performance, reliability, ease of use, and cross-platform support. MySQL is suitable for a variety of application scenarios, from small websites to large enterprise applications.

1.4. System Development Tools

IntelliJ IDEA, Apache Tomcat, MySQL

2. Feasibility Study

2.1. Economic Feasibility

Considering that most of the development tools and platforms used (e.g. Eclipse IDE, MySQL, etc.) are free or open source, and the hardware equipment can also use existing server resources, the initial investment cost is relatively low. In addition, this system is expected to greatly improve the efficiency of school attendance management, reduce human resource consumption, and save a lot of management costs in the long run, so it is economically feasible.

2.2. Technical Feasibility

The technology stack selected for this project includes Java as the back-end development language, Spring Boot framework to simplify the back-end development process, Vue.js for building an interactive front-end interface, and MySQL database for data storage. All of these technologies are currently popular and proven choices with extensive community support and rich development resources. In addition, the system is expected to be deployed on Apache Tomcat server, a widely used lightweight application server suitable for small and medium-sized projects. Therefore, from a technical point of view, it is completely feasible to develop the system.

2.3. Operational Feasibility

The system design will be based on the principles of user-friendliness and ease of operation. Through user training and continuous technical support, we will ensure that users (e.g., teachers, administrators, etc.) will be able to quickly get started and use the system effectively. At the same time, the system will be iteratively updated and optimized through continuous collection of user feedback in order to improve the operability and user satisfaction of the system.

3. Requirement Analysis

3.1. Requirement Description

The detailed functional requirements of this student attendance management system are described as follows:

1. User Management:

The system should support different roles of user login, including students, teachers, and administrators.

It provides basic functions such as user registration, login and password recovery.

2.Student Management:

Allow adding, editing, deleting and querying student information.

Capable of importing and exporting student lists.

3. Teacher Management:

Allows adding, editing, deleting and querying teacher information.

Teachers are able to view the student attendance records of the classes they are responsible for.

4. Class and Course Management:

Manage class information, including creating, modifying and deleting classes.

Manage course information, including adding, editing and deleting courses.

5. Attendance Records:

Automatically record students' attendance status (e.g. present, absent, late, leave).

Provide daily, weekly and monthly attendance reports.

6.Leave Management:

Students can submit leave requests online.

Teachers and administrators review leave requests.

7.Attendance Statistics:

The system automatically generates attendance statistics report, which can be queried by class, course or individual.

It supports outputting the attendance report to PDF or Excel format.

3.2. Performance Requirements

1. The system should be able to support at least 1000 users online at the same time.

2. Response time: The system should take no more than 5 seconds to load all pages under normal operating conditions.

3. Data processing: The system is capable of handling large amounts of data input, such as importing student and teacher data at the beginning of the semester.

4. System Design

4.1. Overall System Functional Module Design

The system has three roles: student, classroom and administrator. According to the roles of users, the system can be divided into the following three modules: student module, classroom module and administrator module. The overall system functional module diagram is shown in Figure 1.

4.1.1. User Management Module

The User Management Module is responsible for handling all user account-related operations, such as registration, login, privilege assignment and personal information update, to ensure that users of various roles can access the system securely and effectively. The Student Management Module and Teacher Management Module provide comprehensive management functions respectively, allowing administrators to add, edit and delete student or teacher information, and supporting batch data processing for quick information update and maintenance.

4.1.2. Class Management Module

This module allows users to create and maintain class information, including class creation, editing and deletion, as well as the course assignments associated with them. The course management module then focuses on adding, updating and deleting course information to ensure that course data is accurate and up-to-date. The Attendance Record module automatically tracks and records student attendance, provides teachers and administrators with real-time attendance data, and generates detailed attendance reports for subsequent analysis.

4.1.3. Leave Management Module

This module simplifies the leave process by allowing students to submit leave applications through the system, while teachers and administrators can approve these applications online, making the whole process more efficient. The Statistical Reporting Module provides powerful data

analysis functions to help administrators understand attendance patterns and trends by generating a variety of statistical charts to optimize teaching and management strategies. Finally, the System Setup module allows administrators to configure and customize system parameters, such as attendance rules and notification settings, to suit the specific needs of the school.

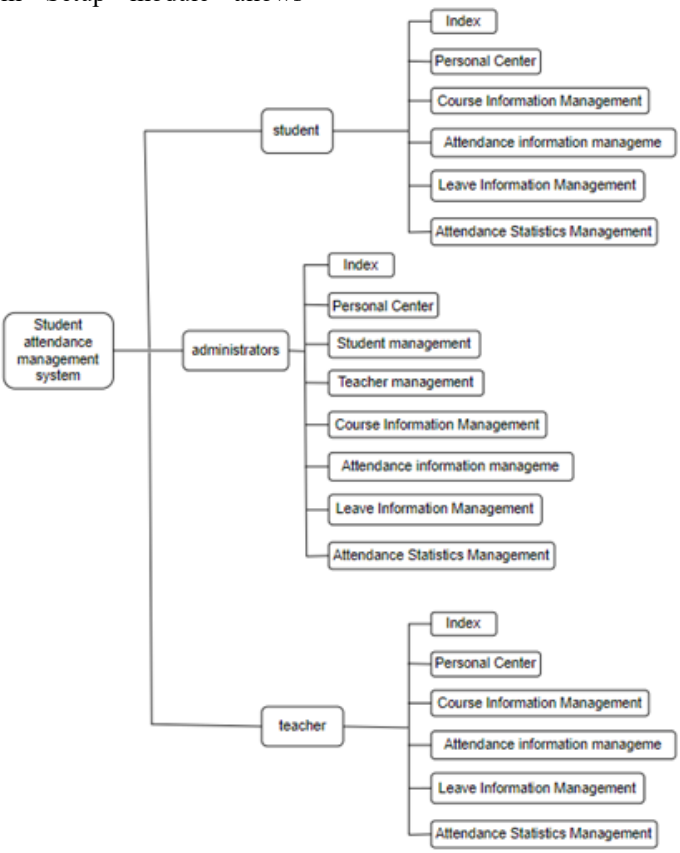

Figure 1. System Functional Structure

4.2. Database Design

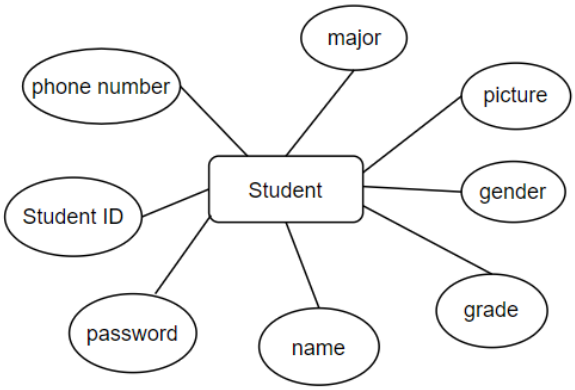

Figure 2. Student Entity Attribute Map

In the Student Attendance Management System, the core data entities and their relationships are carefully designed to meet the diverse needs of the system. The system mainly consists of entities such as student, teacher, class, course, attendance record, leave request, and user account. Among them, the student entity stores the student's personal information, such as name, student number and the class to which he/she belongs, as shown in Figure 2. The teacher entity, on the other hand, records the teacher's details, including name, work number and associated courses.

The class entity manages the basic information of each class, and the course entity describes the detailed information of each course, such as course name, course code, and instructor. The Attendance Record entity records student attendance, including date of attendance, course details and attendance status. The Leave Request entity handles the student's leave request and records the start and end dates of the leave and the reason for the leave.

The User Account entity is used to manage the login credentials and role permissions of system users. The Administrator has the highest privileges in the system and is responsible for managing all data records, including adding or deleting student and teacher information, and updating class and course information. Students and teachers, as general users, have access to specific data, e.g., students can view their attendance records and submit leave requests, while teachers can manage their own class attendance and approve students' leave requests.

In addition, the system is designed with supporting entities such as attendance statistics to provide more detailed and in-depth analysis of attendance data to help administrators and teachers better understand student attendance and trends. The overall entity-relationship diagram provides a clear blueprint for data manipulation in the system, ensuring accurate information and efficient data processing.

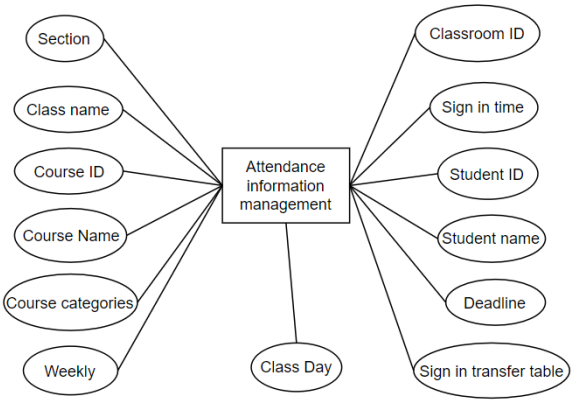

Figure 3. Attendance Information Management E/R Chart

5. System Design and Realization

5.1. System Implementation

Users log in on the login page by filling in the user name, password, selecting the corresponding role and other information.

Figure 4. Login Page

On the student registration page, you can register by filling in the information such as student number, password, confirmation password, name, grade, major, class, and cell phone.

Figure 5. Registration Page

In the class management page, you can view class information and perform operations such as detailing, modifying or deleting class management as needed.

Figure 6. Class Management

Administrator can log in and enter the system to view the home page, personal center, student management, teacher management, class information management and other functions. In the student management page, you can view index, student number, name, gender, grade, specialty, class, cell phone, photo, etc., and carry out details, statistical attendance, modification and deletion operations.

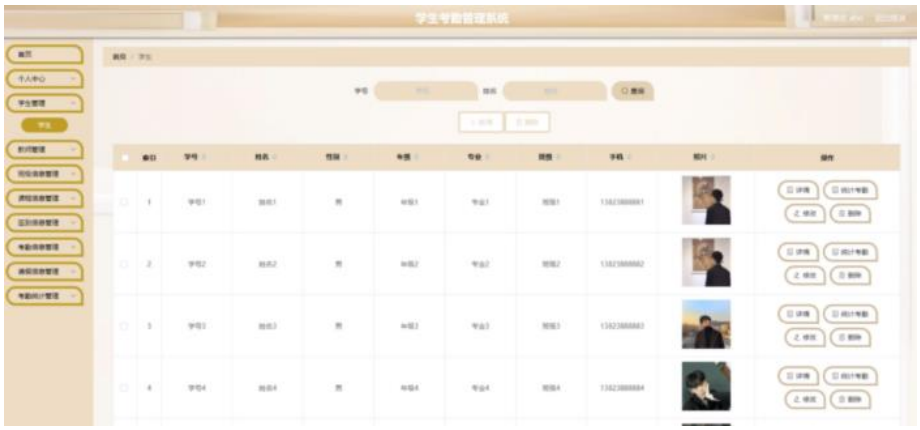

Figure 7. Student Management

In the class information management page, you can view the index, class number, class name, class picture, department, major, class teacher, number of students in the class, etc., and perform scheduling, modification and deletion operations.

Figure 8. Class Information Management

In the Attendance Information Management page, you can view the index, class name, course number, course name, course category, week, class day, session, teacher's work number, teacher's name, check-in status, deadline, student number, name, check-in time, etc., and perform modification and deletion operations.

Figure 9. Course Information Management

5.2. Partial Source Code

```
@RestController
@RequestMapping("/kaoqinxinxi")
public class KaoqinxinxiController {
    @Autowired
    private KaoqinxinxiService kaoqinxinxiService;

    /**
     * Backend List
     */
    @RequestMapping("/page")
    public R page(@RequestParam Map<String, Object> params,KaoqinxinxiEntity kaoqinxinxi,
    HttpServletRequest request){
    String tableName = request.getSession().getAttribute("tableName").toString();
    if(tableName.equals("jiaoshi")) {

    kaoqinxinxi.setJiaoshigonghao((String)request.getSession().getAttribute("username"));
    }
    if(tableName.equals("xuesheng")) {

    kaoqinxinxi.setXuehao((String)request.getSession().getAttribute("username"));
    }
        EntityWrapper<KaoqinxinxiEntity> ew = new EntityWrapper<KaoqinxinxiEntity>();
    PageUtils page = kaoqinxinxiService.queryPage(params, MPUtil.sort(MPUtil.between(MPUtil.likeOrEq(ew, kaoqinxinxi), params), params));

        return R.ok().put("data", page);
    }

    /**
     * Front-end list
     */
@IgnoreAuth
    @RequestMapping("/list")
    public R list(@RequestParam Map<String, Object> params,KaoqinxinxiEntity kaoqinxinxi,
    HttpServletRequest request){
```

```
        EntityWrapper<KaoqinxinxiEntity> ew = new EntityWrapper<KaoqinxinxiEntity>();
    PageUtils page = kaoqinxinxiService.queryPage(params, MPUtil.sort(MPUtil.between(MPUtil.likeOrEq(ew, kaoqinxinxi), params), params));
        return R.ok().put("data", page);
    }

  /**
     * List
     */
    @RequestMapping("/lists")
    public R list( KaoqinxinxiEntity kaoqinxinxi){
        EntityWrapper<KaoqinxinxiEntity> ew = new EntityWrapper<KaoqinxinxiEntity>();
      ew.allEq(MPUtil.allEQMapPre( kaoqinxinxi, "kaoqinxinxi"));
        return R.ok().put("data", kaoqinxinxiService.selectListView(ew));
    }

    /**
     * Query
     */
    @RequestMapping("/query")
    public R query(KaoqinxinxiEntity kaoqinxinxi){
        EntityWrapper< KaoqinxinxiEntity> ew = new EntityWrapper< KaoqinxinxiEntity>();
      ew.allEq(MPUtil.allEQMapPre( kaoqinxinxi, "kaoqinxinxi"));
    KaoqinxinxiView kaoqinxinxiView = kaoqinxinxiService.selectView(ew);
    return R.ok("查询考勤信息成功").put("data", kaoqinxinxiView);
    }

    /**
     *  Backend Details
     */
    @RequestMapping("/info/{id}")
    public R info(@PathVariable("id") Long id){
        KaoqinxinxiEntity kaoqinxinxi = kaoqinxinxiService.selectById(id);
        return R.ok().put("data", kaoqinxinxi);
    }

    /**
     * Front End Details
     */
  @IgnoreAuth
    @RequestMapping("/detail/{id}")
    public R detail(@PathVariable("id") Long id){
        KaoqinxinxiEntity kaoqinxinxi = kaoqinxinxiService.selectById(id);
        return R.ok().put("data", kaoqinxinxi);
    }

    /**
     * Backend Save
     */
    @RequestMapping("/save")
    public R save(@RequestBody KaoqinxinxiEntity kaoqinxinxi, HttpServletRequest request){
      kaoqinxinxi.setId(new Date().getTime()+new Double(Math.floor(Math.random()*1000)).longValue());
        //ValidatorUtils.validateEntity(kaoqinxinxi);
        kaoqinxinxiService.insert(kaoqinxinxi);
        return R.ok();
    }

    /**
     * Front End Save
     */
    @RequestMapping("/add")
    public R add(@RequestBody KaoqinxinxiEntity kaoqinxinxi, HttpServletRequest request){
      kaoqinxinxi.setId(new Date().getTime()+new Double(Math.floor(Math.random()*1000)).longValue());
        //ValidatorUtils.validateEntity(kaoqinxinxi);
        kaoqinxinxiService.insert(kaoqinxinxi);
        return R.ok();
    }

    /**
     * Modify
     */
    @RequestMapping("/update")
    public R update(@RequestBody KaoqinxinxiEntity kaoqinxinxi, HttpServletRequest request){
        //ValidatorUtils.validateEntity(kaoqinxinxi);
        kaoqinxinxiService.updateById(kaoqinxinxi);//全部更新
        return R.ok();
    }

    /**
     * Delete
     */
    @RequestMapping("/delete")
    public R delete(@RequestBody Long[] ids){

        kaoqinxinxiService.deleteBatchIds(Arrays.asList(ids));
        return R.ok();
    }

    /**
     * Reminder interface
     */
  @RequestMapping("/remind/{columnName}/{type}")
  public R remindCount(@PathVariable("columnName") String columnName, HttpServletRequest request,
                       @PathVariable("type") String type,@RequestParam Map<String, Object> map) {
    map.put("column", columnName);
    map.put("type", type);

    if(type.equals("2")) {
      SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
      Calendar c = Calendar.getInstance();
      Date remindStartDate = null;
      Date remindEndDate = null;
      if(map.get("remindstart")!=null) {
        Integer remindStart = Integer.parseInt(map.get("remindstart").toString());
        c.setTime(new Date());

  c.add(Calendar.DAY_OF_MONTH,remindStart);
```

```
                remindStartDate = c.getTime();
                map.put("remindstart",
sdf.format(remindStartDate));
            }
            if(map.get("remindend")!=null) {
                Integer remindEnd =
Integer.parseInt(map.get("remindend").toString());
                c.setTime(new Date());

    c.add(Calendar.DAY_OF_MONTH,remindEnd);
                remindEndDate = c.getTime();
                map.put("remindend",
sdf.format(remindEndDate));
            }
        }

        Wrapper<KaoqinxinxiEntity> wrapper = new
EntityWrapper<KaoqinxinxiEntity>();
        if(map.get("remindstart")!=null) {
            wrapper.ge(columnName, map.get("remindstart"));
        }
        if(map.get("remindend")!=null) {
            wrapper.le(columnName, map.get("remindend"));
        }

        String tableName =
request.getSession().getAttribute("tableName").toString();
        if(tableName.equals("jiaoshi")) {
            wrapper.eq("jiaoshigonghao",
(String)request.getSession().getAttribute("username"));
        }
        if(tableName.equals("xuesheng")) {
            wrapper.eq("xuehao",
(String)request.getSession().getAttribute("username"));
        }

        int count = kaoqinxinxiService.selectCount(wrapper);
        return R.ok().put("count", count);
    }

}
```

6. Conclusion

In this project, we developed a Student Attendance Management System utilizing Spring Boot, Vue.js, and MySQL. This system is designed to provide an efficient and automated solution for recording and managing student attendance, aimed at enhancing daily management efficiency within educational institutions and reducing the administrative workload on teachers.

The architecture of the system is based on a modular design, comprising various functional modules such as user management, student management, teacher management, class and course management, attendance recording, leave management, statistical reporting, and system settings. These modules are integrated to ensure comprehensive functionality and user-friendly operation across the platform.

Through this project, I have gained a profound appreciation for the capabilities and future potential of modern web development technologies. Utilizing Spring Boot for the backend development taught me how to leverage its auto-configuration features and extensive framework support to streamline the development of complex applications. Simultaneously, I honed my skills in using Vue.js to create dynamic front-end interfaces. The component-based approach of Vue.js significantly enhances development efficiency and improves the maintainability of web pages.

Employing MySQL as the database management system has deepened my understanding of relational database design and operations, particularly in managing large data volumes and ensuring data integrity. Moreover, the entire process of designing and implementing this project sharpened my project management skills, covering aspects such as requirements analysis, system design, problem-solving, and teamwork. This experience has not only enhanced my technical skills but also strengthened my ability to tackle real-world problems effectively.

References

[1] Xiao, P. W. (2018). Spring Boot -01- Quick Start (Graphic Tutorial). CSDN. Retrieved January 2024, from https://blog. csdn.net/ qq_40147863/article/details/84194493.

[2] Zhang, Y. (2022). "Step-by-step Vue.js 3 Front-end Development Practice". Tsinghua University Press.

[3] MySQL Tutorial. Runoob. https://www. runoob.com/ mysql/ mysql-tutorial.html.