

UNDERGRADUATE PROJECT PROGESS REPORT

|  |  |
| --- | --- |
| Project Title: | Java-Based Web Driving School Management System |
| Surname: | Liu |
| First Name: | Yuchen |
| Student Number: | 202118010217 |
| Supervisor Name: | Javier Jia |
| Module Code: | CHC 6096 |
| Module Name: | Project |
| Date Submitted: | 2024.12.27 |

Table of Contents

[1 Introduction 3](#_Toc185851891)

[1.1 Background 3](#_Toc185851892)

[1.2 Aim 4](#_Toc185851893)

[1.3 Objectives 4](#_Toc185851894)

[1.4 Project Overview 4](#_Toc185851895)

[1.4.1 Scope 4](#_Toc185851896)

[1.4.2 Audience 5](#_Toc185851897)

[2 Background Review 5](#_Toc185851898)

[3 Technical Progress 7](#_Toc185851899)

[3.1 Approach 7](#_Toc185851900)

[3.2 Technology 8](#_Toc185851901)

[3.3 Testing and Evaluation Plan 9](#_Toc185851902)

[3.4 Design and Implementation 10](#_Toc185851903)

[4 Project Management 11](#_Toc185851904)

[4.1 Activities 11](#_Toc185851905)

[4.2 Schedule 11](#_Toc185851906)

[4.3 Project Version Management 11](#_Toc185851907)

[4.4 Project Data Management 11](#_Toc185851908)

[4.5 Project Deliverables 12](#_Toc185851909)

[5 Professional Issues and Risk: 12](#_Toc185851910)

[5.1 Risk Analysis 12](#_Toc185851911)

[5.2 Professional Issues 12](#_Toc185851912)

[6 References 13](#_Toc185851913)

# Introduction

## Background

(1). Overview of Driving School Management Systems:

The management system of driving school is the key to the operating efficiency and benefit of driving school. They are the central hub for managing the complex logistical challenges inherent in the operation of a driving school, ensuring that all aspects of the operation run smoothly. The basic concept will always about automating tasks and reducing human errors.

(2). Challenges in Current Driving School Management:

Including the difficulty of dealing with student registration, which can cause a lot of trouble. How to properly schedule appointments is another problem, with students and coaches struggling to find their both convenient class times. The ability to maintain effective communication between coaches and students is also a challenge. These not only effect the smooth running of the school, but also have a negative impact on the student experience.

(3). Role of Technology in Modernizing Driving Schools:

In the field of web development and programming languages such as Java, new ways have been created to change the way driving schools’ management. These systems can automatically solve tasks that previously needed to be done by hands. A web-based system can provide access to scheduling and registration, allowing students to book and reschedule appointments. Java is extensible so it’s suitable for building system that can handle the management of the driving schools.

(4). Value of Integrated Data Analysis Tools:

Integrating data analytics tools into a driving school management system adds a layer of strategic insight that can greatly benefit the decision-making process. Integrated data analysis tools can optimize operations because they can highlight trends and anomalies that might otherwise go unnoticed. This ability enables driving schools to be proactive in developing programs and continuously improving services, ultimately leading to better outcomes for schools and students.

## Aim

The overall goal of this project is to develop a well-functioned online driving school management system to improve the management efficiency of driving schools, optimize the learning experience of students, and meet the needs of students’ self-driving.

## Objectives

To achieve this goal, the project will accomplish the following tasks:

1. Demand analysis: To understand of the management needs of driving schools, clear system functions.
2. System design: Set the structure of the system, design the front-end and back-end interaction.
3. Function development: To realize functions like the registration, login, announcement, coach management and etc.
4. Testing and Optimization: To perform a full-scale test, and optimize user experience and system stability.
5. Release: To deploy the developed system into a live environment, ensuring that all components are fully operational and secure.

## Project Overview

This project is a web-based driving school management system, which have multiple user roles like administrator, coach, student and consultant, and to realize online registration, appointment, coach management and other functions, the system will support the daily management of students and driving schools through efficient information flow, while providing data analysis tools, which can help in decision-making.

### Scope

This website will have these following functions:

1. User registration and login, support multiple identity and permission management.
2. Announcement issued for driving school information and coach recommendation.
3. Coach management, including information entry, scheduling and pass rate statistics.
4. Package ordering allows consultants to select the right package and pay online.
5. Student management, including booking practice, test and coach replacement functions.
6. Searching module provides information retrieval service for each user.

### Audience

**Driving School Administrators:**

Role: Oversee the overall operations of the driving school.

Needs: Require robust tools for managing student registrations, scheduling appointments, assigning coaches, handling payments, and monitoring the school’s performance.

Benefit: The system will streamline administrative tasks, reduce paperwork, and provide real-time insights into the school’s operations.

**Coaches/Instructors:**

Role: Provide driving lessons and track student progress.

Needs: Need an easy way to view their schedules, manage appointments, access student information, and record training outcomes.

Benefit: The system will help them organize their daily tasks efficiently and communicate effectively with students and administrators.

**Students:**

Role: Individuals enrolled in the driving school to learn driving skills.

Needs: Desire a user-friendly interface to register online, schedule or reschedule lessons, make payments, and track their learning progress.

Benefit: The system enhances their learning experience by providing convenience, transparency, and easy access to information.

# Background Review

In the realm of Driving School Management Systems (DSMS), several platforms have emerged to cater to the specific needs of driving schools. These systems aim to streamline operations, enhance student experiences, and improve overall efficiency. Here is a summary of existing DSMS platforms available in the market:

a. EzyDrive: This web-based platform is designed to simplify the scheduling process for both students and instructors. It offers features like real-time booking, class management, and student tracking, which are essential for efficient driving school operations.

b. DriveLogic: Known for its comprehensive set of features, DriveLogic provides solutions for billing, instructor management, and even student progress tracking. It is designed to handle the complex needs of driving schools, from financial management to educational oversight.

c. Teachworks: Although not specifically designed for driving schools, Teachworks is a versatile education management system that can be adapted to suit the needs of driving institutions. It offers features such as class scheduling, attendance tracking, and payment processing.

Technology Stack:

a. Technologies Used: DSMS platforms are built using a variety of technologies to ensure they are robust, scalable, and user-friendly. Common technologies include:

- PHP: A popular server scripting language for web development, known for its flexibility and wide community support.

- .NET: A framework developed by Microsoft for building a wide range of applications, including web-based DSMS, known for its performance and security features.

- Cloud-Based Solutions: Many DSMS are moving to cloud-based solutions to offer scalability, accessibility, and robust data storage and retrieval capabilities.

Data and Research:

To further substantiate the analysis of existing DSMS platforms, it is essential to refer to data and research that provide insights into their usage and effectiveness. For instance, market reports from Statista or Gartner can offer industry-wide data on the adoption of such systems and their impact on driving school management. Academic literature, such as articles from the Journal of Transportation Research or the International Journal of Educational Technology in Higher Education, can provide in-depth studies on the integration of technology in driving education and its outcomes.

In 2010, there were security experts, industry leaders, and several U.S. senators in order to reduce the drivers’ driving hazards, they all made the decision that to develop a driving supervision management solution [1]. For all the driving schools, first thing will always be management, a well functioned driving school management system can really make whole driving more efficient and the system is conducive to the real management of driving school. Therefore, foreign research on information system is relatively mature. UK driving school applicant can make driving test reservation online [2], they can decide whenever they want to take the exam, it is different from the arrangement of the Motor Vehicle administration and driving school in China, measures in UK can more outstanding the concept of convenience to the people. In the 1970s, car driving simulator training systems appeared in some developed countries such as the United States and Western Europe [3], and according to Manzura, South Korea's driving school management system is the student registration, registration, training, records, evaluation and examination of the entire process of computerized management, in cooperation with the government [4], we can all see these achievements of foreign driving school management systems had made.

Management system of driving school in China had a late start than the foreign countries, thus the development of driving school management system in China is not as good as the system in foreign countries, but with the rapid development of the national economy, the domestic driving school management system is also on its way of development. Now most of the research of domestic driving school management system is mainly reflected in the function and technology, for example, Dr. Zhou and others pointed out that the driving school mainly realizes 7 functions, including information input, storage, browsing, query, data statistics, report output and data maintenance [5].

# Technical Progress

## Approach

The project employs a Waterfall software development methodology, which is characterized by a linear and sequential process. This approach ensures that each phase of the project lifecycle is well-defined and completed before progressing to the next. The methodology is particularly suited to this project as it allows for a structured approach to gathering requirements, designing, implementing, and testing the system. The phases include:

1. Project Initialization:

The project begins with defining its scope, objectives and deliverables. Stakeholder expectations are documented, and a project timeline is constructed.

1. Requirements Analysis:

A detailed analysis of the driving school management system’s needs is conducted. This phase identifies the essential system functionalities, such as user registration, appointment scheduling, and payment integration. The requirements are documented in a Software Requirement Specification (SRS) document.

1. System Design:

Based on the requirements, the system architecture and design are developed. This includes database schema design, defining API endpoints, and creating wireframes or mockups for the frontend interface.

1. Implementation:

The actual coding of the system is carried out, adhering to the design and ensuring modularity. Both the frontend (Vue.js) and backend (Spring Boot) are developed concurrently.

1. Testing:

Rigorous testing is performed to ensure the system meets its functional, performance, and security requirements. A Test-Driven Development (TDD) approach is adopted, ensuring that test cases are written alongside or before the implementation of code.

1. Deployment:

The system is deployed in a live environment, where its performance under real-world conditions is evaluated. Adjustments are made as necessary based on user feedback.

1. Maintenance and Closure:

Post-deployment, the system undergoes regular maintenance to address bugs, enhance performance, and incorporate new features if required. The project concludes once all deliverables are met.

Requirement Gathering Methods:

To ensure the system addresses real-world problems effectively, the following requirement gathering methods are employed:

1. Literature Review:

Academic journals, case studies, and industry reports are extensively reviewed to understand the current trends, challenges, and innovations in driving school management systems. This provides a foundation for identifying system functionalities and technical approaches that are both feasible and innovative.

1. Surveys:

Surveys are conducted with key stakeholders, including driving school administrators, instructors, and students. These surveys aim to identify pain points in the current manual or semi-automated management systems. For example, common challenges like inefficient scheduling and poor communication between students and coaches are highlighted.

1. Technology Forums:

Developer communities and technical forums are explored to gather insights into existing solutions and their limitations. These forums provide practical advice on selecting the appropriate technology stack and identifying potential challenges during implementation.

Testing and Evaluation Approach:

The project adopts a Test-Driven Development (TDD) approach, a software development process where tests are written before or alongside the development of functional code. This ensures that the system is developed with a focus on correctness, reliability, and stability from the outset.

The following testing strategies are employed:

1. Functional Testing:

Each module of the system is tested to verify that it performs its intended functions. For example:

* User registration: Ensures users can register with valid inputs and receive appropriate error messages for invalid inputs.
* Appointment scheduling: Verifies that users can book, modify, or cancel appointments based on system rules.
* Payment integration: Confirms that payments are processed securely and accurately.

1. Performance Testing:

The system’s performance is evaluated under different scenarios to ensure scalability and reliability. Specific tests include:

* Simulating concurrent users to assess how the system handles high traffic.
* Measuring response times for database queries and page loads during peak usage.

1. Security Testing:

Security testing ensures that the system protects user data and prevents unauthorized access. Key tests include:

* Authentication Testing: Verifies that only registered users can log in, and that strong password policies are enforced.
* Encryption Testing: Ensures that sensitive user data, such as passwords and payment information, are encrypted during transmission and storage.
* Vulnerability Testing: Simulates potential attacks, such as SQL injection and cross-site scripting (XSS), to identify and address security loopholes.

## Technology

|  |  |  |
| --- | --- | --- |
|  | Classification | Environment |
| Hardware | CPU | 11th Gen Intel Core i7-11800H @ 2.30GHz 8Core 16Threads |
| GPU | NVIDIA GeForce RTX 3070 Laptop GPU 8GB(VRAM) |
| Python Version | 3.10.7 |
| CUDA Version | 12.1 |
| CuDNN Version | - |
| Frontend Framework | voe |
| Backend Framework | Springboot |
| Database | Mysql |
| Development Tools | JAVA |

Table 1: Hardware and Software

OneDrive is used to upload the documents of each stage, update the task, and ensure that the tutor can timely understand the project progress and give comments and guidance.

The source code of the project is hosted on Github, a platform that provides code hosting, review, and management tools. This facilitates access and collaborative development.

## Testing and Evaluation Plan

Testing the system will involve the following steps:

1. Functional Testing
   * Test cases for user registration, login, and scheduling.
   * Example Test Case: Input invalid credentials to check error handling during login.
2. Performance Testing
   * Simulate 100 concurrent users booking lessons to test response time and server capacity.
   * Evaluate database query optimization using Redis caching.
3. Security Testing
   * Test the robustness of RSA encryption for user data.
   * Simulate potential attacks, such as SQL injection, to evaluate system defenses.

## Design and Implementation

The design and implementation of the system have been carried out in a structured manner to ensure modularity, scalability, and ease of use. Progress so far includes the following key areas:

1. Database Design

The foundation of the system lies in a robust and well-structured database, which ensures efficient data storage and retrieval. The following steps have been completed:

* Entity-Relationship (ER) Diagrams:

Detailed ER diagrams have been designed for all major system components, including users, appointments, packages, and feedback. These diagrams illustrate the relationships and cardinalities between entities, ensuring a logical data structure.

Key entities include:

* Users: Captures details for administrators, instructors, students, and consultants.
* Appointments: Tracks lesson bookings, cancellations, and rescheduling.
* Packages: Maintains information about driving lesson packages, pricing tiers, and features.
* Feedback: Stores user evaluations and complaints for system improvement.
* MySQL Schema Implementation:

A MySQL database has been designed and implemented based on the ER diagrams. The schema includes primary keys, foreign keys, and constraints to maintain data integrity.

Key tables include:

* User Table: Stores user details such as name, role, contact information, and encrypted passwords.
* Appointment Table: Records booking details, including user ID, appointment date, time, and status.
* Package Table: Lists available driving packages with pricing and descriptions.
* Transaction Table: Logs all payment-related data for audit and tracking purposes.



Figure 1: Database

1. Backend Development

The backend serves as the core of the system, managing data flow and implementing business logic. Significant progress includes:

* RESTful APIs:

APIs have been implemented using Spring Boot to handle essential functionalities, including:

* User Authentication: Secure login and registration APIs ensure role-based access control for administrators, coaches, students, and consultants.
* Scheduling and Appointments: APIs allow users to book, reschedule, or cancel appointments, with real-time updates to the database.
* Payment Processing: Payment APIs integrate with third-party gateways (e.g., Alipay and WeChat Pay) to enable seamless and secure transactions.
* Business Logic Implementation:

The backend handles various operations, such as generating appointment schedules, assigning instructors, and calculating package discounts.

1. Frontend Development

The frontend has been designed with a focus on user experience and accessibility, ensuring that users can easily navigate and interact with the system. Progress includes:

* Interface Design:

User-friendly interfaces have been developed using **Vue.js** to meet the needs of different user roles:

* **Registration and Login Pages:** Allow users to register with their details and log in securely.
* **Dashboard Interfaces:** Provide personalized dashboards for each user role (e.g., administrators can view analytics, students can manage bookings).
* **Appointment Booking Page:** Allows users to view available slots, book lessons, and track their schedules.
* **Admin Management Panel:** Enables administrators to manage instructors, packages, and student progress.
* Responsive Design:

The interfaces are designed to be mobile-friendly, ensuring seamless access across devices such as smartphones, tablets, and desktops.

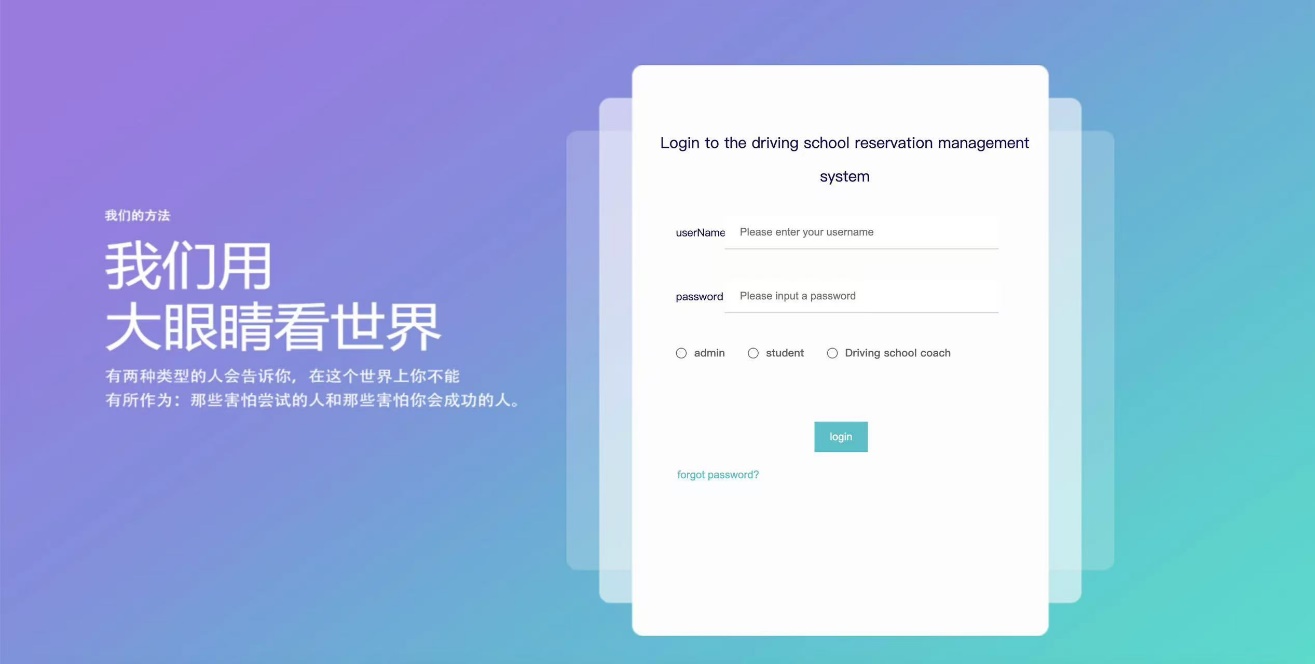


Figure 2: Login Page

1. Integration

Integration efforts ensure that all system components work cohesively. Key integration tasks completed include:

* Payment Gateway Integration:

Alipay and WeChat Pay APIs have been integrated to enable users to pay for lesson packages online. This ensures smooth and secure transaction processing.

Features include payment status tracking, error handling for failed transactions, and user notifications for successful payments.

* Backend and Frontend Integration:

RESTful APIs have been successfully integrated with the frontend interfaces, enabling dynamic data exchange between the client and server.

Examples:

* Registration forms send user data to the backend for validation and storage.
* Appointment pages retrieve available time slots from the database in real time.

1. Additional Progress

* Logging and Error Handling:

Comprehensive logging mechanisms have been implemented in the backend to track system activity and troubleshoot errors efficiently.

* Version Control:

All development work is managed using GitHub. Separate branches are maintained for different modules (e.g., user-auth, appointment-management), ensuring clean and organized code.

# Project Management

## Activities

|  |  |  |
| --- | --- | --- |
| Objective | Completed Tasks | Pending Tasks |
| Requirement Analysis | Conducted surveys and literature review | Finalize user stories based on feedback |
| System Design | Database schema and API endpoints designed | Finalize frontend design |
| Function Development | User registration, login implemented | Complete scheduling and admin modules |
| Testing and Optimization | Testing and Optimization | Perform security and performance testing |

Table 2: Objectives

## Schedule

A Gantt chart is used to track progress:

* Completed Tasks: Requirement analysis, backend API development for registration and login.
* Pending Tasks: User testing, performance optimization, deployment.

## Project Version Management

The project employs GitHub for version control:

* Branching Strategy: Separate branches for each module (e.g., user-auth, admin-management).
* Commit Management: Frequent commits with clear messages to document progress.
* Collaboration Tools: Pull requests for peer code reviews and issue tracking.

## Project Data Management

Data is managed using:

* GitHub Repository: Stores project logs, design documents, and source code.

## Project Deliverables

|  |  |
| --- | --- |
| Deliverable | Status |
| Project Proposal | Submitted |
| Progress Report | In Progress |
| Final Report | Pending |
| Source Code/Software | Under Development |
| Test Report | Pending |
| Presentation Slides | Pending |

Table 3: Deliverable

# Professional Issues and Risk:

## Risk Analysis

|  |  |  |
| --- | --- | --- |
| Risk | Mitigation Strategy | Status |
| Delay in requirement gathering | Conduct weekly reviews to ensure progress. | Resolved |
| API Integration Failure | Test third-party APIs in isolation before merging. | Resolved |
| Future Risk: Scalability issues | Adopt cloud services like AWS for hosting. | Ongoing |

Table 4: Risk analysis

## Professional Issues

The project addresses several professional concerns:

* Legal: Compliance with data protection laws like GDPR (General Data Protection Regulation) to ensure user privacy.
* Social: Improving user experience by reducing manual administrative tasks and offering flexible scheduling.
* Ethical: Using secure payment methods to avoid fraud or misuse.
* Environmental: Promoting paperless workflows to reduce resource waste.

Professional conduct is guided by ACM and BCS codes, emphasizing system reliability, data security, and user-centric design.

# References

[1] Bowen, L. et al. (2020) ‘Factors underpinning unsafe driving: A systematic literature review of car drivers’, *Transportation Research Part F: Traffic Psychology and Behavior*, Volume 72, pp. 184-210. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S1369847820304046> (Accessed: 25/10/2024).

[2] LEARNER DRIVER ZONE, (2022) *How to Book A Driving Test.* Available at: <https://learnerdriverzone.co.uk/helpful-guides/booking-a-driving-test/> (Accessed: 25/10/2024).

[3] Martin-delosReyes, L. M. et al. (2019) ‘Efficacy of training with driving simulators in improving safety in young novice or learner drivers: A systematic review’, *Transportation Research Part F: Traffic Psychology and Behavior,* Volume 62, pp. 58-65. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S1369847818304406> (Accessed: 26/10/2024).

[4] Tairova, M. (2020) Getting A Driver’s License In South Korea: Complete Guide. Available at: <https://koreabyme.com/getting-a-drivers-license-in-south-korea-complete-guide/> (Accessed: 26/10/2024).

[5] Zhou, J. et al. (2016) ‘Design and Implementation of Driving School Information Management System Based on Dreamweaver’, International Conference on Sensor Network and Computer Engineering, Volume 6, pp. 510-514. Available at: <https://www.semanticscholar.org/paper/Design-and-Implementation-of-Driving-School-System-Zhou-Cai/add33d6ad4958cb9346824c062b370fe98022478> (Accessed: 26/10/2024).