# CS 255 Business Requirements Document

## System Components and Design

### Purpose

Design a system for DriverPass that allows driver training and increases the success rate of students taking their driving tests. DriverPass has identified a gap in the market where students are failing their tests because they are only studying with previous exams. The client, DriverPass, wants a system that provides online practice tests and on-the-road training, allowing students to better prepare for their driving exams.

### System Background

DriverPass needs a system that offers both online learning and practical driving lesson scheduling. The main problem they want to solve is the high failure rate of students due to improper preparation. The system must allow users to take online classes, schedule driving lessons, track progress, and ensure secure role-based access. Some of the components needed for the system include:

* Online practice exams and course materials.
* System for scheduling driving lessons.
* Database with role-based access control.
* Feature to track and monitor student’s progress and system activity.
* Integration with DMV to update whenever new practices are available.
* Ability to enable or disable training packages from the administrative dashboard.

### Objectives and Goals

* Allow students to register for DriverPass services, choose a training package, and schedule driving lessons.
* Provide access to online course materials and practice tests.
* Enable users to book, modify, or cancel reservations for driving lessons.
* Maintain a secure login system with different access roles for students, instructors, and administrators.
* Track and report student progress, including completed tests, scores, and lesson history.
* Keep material up to date with DMV most recent questions and training materials by regular and automated updates.
* Offer reporting tools to allow management to monitor reservations, student progress, and system usage.
* Allow administrative control over packages and availability (enabling or disabling certain training packages).

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

* The system will be a web-based application hosted in the cloud to ensure accessibility from any device with an internet connection. The system should run in a way that pages and user’s actions, like scheduling a lesson or taking a practice test, should not take more than two seconds. The system must have a feature to download content, to allow working on it while offline. To make any changes to data it must be performed online to avoid duplicate data on different servers. The system should be updated once per day at off-peak hours to check for updates and new content from DMV.

#### Platform Constraints

* The system will run on any operating system, as it will be a web application, the only requirement is to have a web browser and internet connection. It will be compatible with major browsers like Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Brave, etc. The backend will require a relational database management system (RDBMS), like MySQL or PostgreSQL, to store all the information. The system will be built in a cloud-based environment, where AWS, Azure, or Google Cloud could be used for hosting, ensuring scalability.

#### Accuracy and Precision

* User authentication will be role-based, distinguishing between students, instructors, secretaries, and management. The input will be case-sensitive, allowing for most complex and secure usernames and passwords. Input validation will be implemented to ensure that accurate data is provided, preventing malicious code injections, and invalid inputs. The system should inform the admin of a problem whenever the system is taking more than two seconds to perform actions, a reservation conflict is happening, or when the system is not working (other than for an update or scheduled maintenance). As an extra layer of security, the admin should also be informed whenever a user has attempted to enter the password more than 3 times, as it might be a malicious user trying to break into the system.

#### Adaptability

* The system will allow management to add, remove, and modify users without changing code, using a user-friendly interface. However, changes to the code have to be made to add or remove schedules. The system should support automatic adaptation to platform updates, ensuring it works on web browsers every time they make an update. Management should have full control over user access, including the ability to reset passwords, deactivate accounts, and adjust permissions.

#### Security

* User login: Users must login with a username and password. Multi-factor authentication can be added for extra security.
* Secure data exchange: All data transmitted between the client and server will be encrypted using SSL/TSL protocols to prevent interception.
* Brute force protection: If an account is detected to be attempting to enter using a brute force attack, the system should lock the account and notify the user via email. The account can be enabled back by management after validating the user’s account is safe again.
* Password recovery: User who forget their passwords will be able to reset them via email or it can be recovered with help of management, after verifying the user’s identity.

### Functional Requirements

* The system shall validate user credentials when logging in, enforcing role-based access control.
* The system shall allow students to register for an account, select a training package, and schedule driving lessons.
* The system shall allow students to take online practice tests and track their progress, including test scores and completion status.
* The system shall allow users (students and secretaries) to book, modify, or cancel driving lesson appointments.
* The system shall assign students to available driving instructors, cars, and time slots, preventing scheduling conflicts.
* The system shall allow administrators to generate reports on student progress, lesson bookings, and system activity.
* The system shall allow the IT administrator to manage user accounts, reset passwords, and deactivate accounts when needed.
* The system shall be integrated with the DMV database to ensure training materials and practice tests are up to date.
* The system shall provide secure payment processing for students purchasing training packages.
* The system shall allow DriverPass management to enable or disable training packages without requiring system updates.
* The system shall offer a mobile-responsive web interface accessible from any device with an internet connection.
* The system shall allow users to download content to be able to use it offline, where any changes will be sent to the server when online.

### User Interface

**Students**

* Register, log in, and manage their accounts.
* Select a training package and make payments.
* Schedule, modify, or cancel driving lessons.
* Take online practice tests and view results.
* View lesson history, instructor feedback, and progress reports.

**Instructors**

* View their assigned students and lesson schedules.
* Leave comments and feedback for students after lessons.
* Update lesson completion status.

**Secretaries**

* Schedule, modify, or cancel lessons on behalf of students.
* View student details and lesson history.

**Management**

* Manage user accounts (add, modify, remove).
* Reset passwords and deactivate accounts.
* Maintain system security and monitor system activity.
* View reports on student progress, lesson bookings, and system performance.
* Enable or disable training packages as needed.

### Assumptions

* Users will have a laptop, personal computer, tablet, phone, or any other device that allows them to browse the internet.
* Users will have internet access.
* Users speak English.
* Users know how to enter a website.
* Users have a debit/credit card, PayPal, or any other valid form of payment.
* DMV will grant us access to their Application Programming Interface (API) to be able to recover and update the lessons and questions on our end.
* Users will use Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, or Brave to browse the internet.

### Limitations

* Real-Time updates: If DMV changes requirements or any other section that we are not retrieving in the API (like rules, requirements, or any other information that is not commonly found when transmitting data through an API), these changes would not be noticed unless an actual person is checking manually.
* Scalability: The system might be able to scale, however, if scaling is more than expected it will require infrastructure upgrades, making it more expensive to maintain.
* System downtime for updates: As we will update the system every day to check for updates with the DMV, there might be periods of time when the system is shortly unavailable.
* Dependency on third party services: As the system relies on third party services for payment and hosting services, if these services experience outages or problems, our system will be affected too.

### Gantt Chart

*A screenshot of a computer screen

Description automatically generated*