# CS 255 Business Requirements Document

## System Components and Design

### Purpose

* The purpose of this project is to develop a driver training and scheduling system for DriverPass. DriverPass is a company aiming to help customers prepare for their driving tests. The system will allow users to take online classes, complete practice tests, and schedule on the road driving lessons.
* The client is DriverPass, a company founded by Liam. He identified a market need for better driver training, as many people fail their driving tests at the DMV.
* Liam wants his system to book, modify, and cancel driving lessons online or through a secretary, track reservations, and assign drivers and cars. It must have secure role-based access, allow data downloads, and keep logs of user actions. Additionally, it should connect with the DMV for test updates, be web based, and cloud hosted for easy access and minimal maintenance.

### System Background

* DriverPass wants a driver training and scheduling system that allows customers to:
  + Take online courses and practice tests to prepare for their driving exam.
  + Schedule, modify, and cancel driving lessons online or through a secretary.
  + Track their progress, including test results and completed lessons.
  + Receive training with assigned drivers and cars based on their selected package.
  + Ensure security by restricting system access based on user roles.
  + Stay updated with DMV regulations by integrating with DMV updates.
* The main issue DriverPass addresses is the high failure rate of driving tests due to poor training. Many learners struggle with both the written exam and the practical driving test. Existing training programs are not structured well leading to poor preparation. DriverPass aims to fill this gap by providing a structured, accessible, and effective training system.

### Objectives and Goals

* The DriverPass system should provide seamless and efficient experience for both customers and admins by enabling:
  + Customer Features
    - Register and create an account
    - Book, modify, and cancel driving lessons online
    - Choose a driving package and make secure payments
    - Take online practice tests and view results
    - Track training progress
  + Admin and Staff Features
    - Assign instructors and vehicles to lessons
    - Manage user roles and access levels
    - Reset passwords and disable accounts
    - Track and log user activity
    - Generate and download reports for business analysis
  + System-Wide Features
    - Secure, cloud-based access from any device
    - Integration with DMV updates for test content and regulations
    - Automated notifications for customers and staff
    - Data security and backup management
* Measurable Tasks:
  + User Management & Security
    - Design user roles: Admin, IT officer, Secretary, Customer
    - Implement login, password reset, and role-based access control
    - Ensure data encryption and compliance with security standards
  + Lesson Reservation System
    - Create an appointment scheduling module with options for booking, modifying and cancelling lessons
    - Implement automatic instructor and vehicle assignment based on availability
    - Log and track all changes made to lesson bookings
  + Online Training & Testing Module
    - Develop an interactive online test system that tracks progress and cores
    - Store test results and lesson progress for each user
    - Enable lesson notes and feedback from instructors to be recorded and viewed.
  + Payment & Registration System
    - Implement a secure registration system collecting user details and payment information
    - Enable selection and purchase of driving packages
    - Ensure secure credit card transactions and data storage.
  + DMV Integration & Compliance
    - Design a system to fetch and update DMV regulations automatically
    - Set up notifications for administrators when DMV content updates are available
  + Reporting & Tracking
    - Create reporting tools to generate logs of user activity, payments, and lesson schedules
    - Allow admin to download customer and financial reports
  + Cloud-Based Web Interface
    - Develop a user-friendly web interface accessible on computers and mobile devices
    - Implement a dashboard displaying test progress, appointments, and instructor feedback

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

*What environments (web-based, application, etc.) does this system need to run in? How fast should the system run? How often should the system be updated?*

* The system shall run as a web-based and mobile friendly application, accessible through all modern browsers and mobile devices.
* The system shall support real-time updates for scheduling and notifications
* System updates shall occur at least once per month with critical bug fixes applied within 48 hours

#### Platform Constraints

*What platforms (Windows, Unix, etc.) should the system run on? Does the back end require any tools, such as a database, to support this application?*

* The system shall be compatible with Windows, macOS, Android, iOS, and major Linux distributions
* The backend shall be hosted on a cloud platform and shall use a relational database like MySQL to store user, appointment, and course data.
* The application will use JavaScript Framework on the frontend and Java or Node.js on the backend

#### Accuracy and Precision

*How will you distinguish between different users?* *Is the input case-sensitive? When should the system inform the admin of a problem?*

* Each user will be uniquely identified by a secure user ID or email address and authenticated via password or multi-factor authentication
* Input will be case sensitive
* The system will alert the admin when:
  + An error occurs in DMV integration
  + A lesson booking fails
  + A suspicious login attempt is detected
  + A customer account is locked due to failed logins

#### Adaptability

*Can you make changes to the user (add/remove/modify) without changing code? How will the system adapt to platform updates? What type of access does the IT admin need?*

* The system shall allow admins to add, remove, or modify users and their roles through the admin interface, without changing the source code
* The system shall be built with modular components to support updates in UI, business logic, and data storage with minimal disruption
* IT admins shall have access to the system dashboard, server logs, security audit tools, and deployment utilities to manage updates and maintenance.

#### Security

*What is required for the user to log in? How can you secure the connection or the data exchange between the client and the server? What should happen to the account if there is a “brute force” hacking attempt? What happens if the user forgets their password?*

* Users shall log in with unique credentials and optionally use Two Factor Authentication
* All data exchanged between client and server shall be encrypted using HTTPS
* Accounts with more than 5 failed login attempts within 10 minutes shall be temporarily locked, and an alert sent to the admin
* Users who forget their password can reset it securely using email or SMS verification process
* Role-based access control will restrict access based on user roles

### Functional Requirements

*Using the information from the scenario, think about the different functions the system needs to provide. Each of your bullets should start with “The system shall . . .” For example, one functional requirement might be, “The system shall validate user credentials when logging in.”*

* The system shall validate user credentials during login
* The system should allow users to register and create an account
* The system shall allow customers to book, modify, and cancel driving lessons
* The system shall automatically assign instructors and vehicles based on availability.
* The system shall store and display customer training progress and test results
* The system shall allow admins to manage users, reset passwords, and assign roles
* The system should provide online practice tests and track test scores
* The system shall integrate with DMV APIs to fetch regulation updates
* The system shall send automated notifications for upcoming lessons and schedule changes
* The system shall log all users and admins actions for auditing
* The system shall allow admins to generate and download users and financial reports.

### User Interface

*What are the needs of the interface? Who are the different users for this interface? What will each user need to be able to do through the interface? How will the user interact with the interface (mobile, browser, etc.)?*

* The interface should be web-based and responsive, designed for use on both mobile devices and desktop browsers.
* Customers shall use the interface to:
  + Register, log in, and manage their profile
  + Book, reschedule, or cancel lessons
  + Take practice tests and view results
  + Track their training progress and receive feedback
* Admins shall use the interface to:
  + View and manage customer schedules
  + Assign instructors and vehicles
  + Access reporting tools and downloading data
  + Manage user accounts and permissions

### Assumptions

*What things were not specifically addressed in your design above? What assumptions are you making in your design about the users or the technology they have?*

* Users will have internet access and a modern browser or mobile device
* DMV provides API access or a mechanism for data updates that the system can integrate with
* Users will require basic computer literacy to navigate the interface
* Instructors will use a mobile or web dashboard to track lessons and provide feedback

### Limitations

*Any system you build will naturally have limitations. What limitations do you see in your system design? What limitations do you have as far as resources, time, budget, or technology?*

* System functionality depends on the availability and reliability of cloud services and DMV integration APIs
* There may be limitations on real-time updates in regions with slow internet connections
* Resource constraints such as small development team or limited test9ing devices could delay certain features
* Initial development may focus on English only UI with localization added later
* User concurrency and load may be limited initially until further scaling is tested and implemented

### Gantt Chart

