# CS 255 Business Requirements Document

## System Components and Design

### Purpose

*What is the purpose of this project? Who is the client and what do they want their system to be able to do?*

* **Client:** DriverPass
* **Goal:** Provide an online platform for students to access practice exams and on-the-road training to better prepare for driving tests.
* **System Needs:**
  + Allow students to register, book lessons, and take online practice exams.
  + Allow for management of driving lesson appointments.
  + Offer a system that is easy to use, intuitive, and secure.
  + Enable tracking of students’ progress in their learning and driving lessons.
  + Integrate online and offline components (e.g., practice exams and in-person lessons).

### System Background

*What does DriverPass want the system to do? What is the problem they want to fix? What are the different components needed for this system?*

* **Problem to Address:**
  + Many students struggle to pass their driving tests, indicating a gap in available driver education and preparation.
  + Lack of efficient scheduling and lesson management systems for students and instructors.
* **Solution Offered:**
  + DriverPass will create a platform for students to prepare for their driving tests with access to online practice exams and in-person driving lessons.
  + The system will enable users (customers, instructors, and admins) to easily interact with the scheduling, lesson materials, and performance tracking.
* **Components Needed:**
  + Online platform for access to practice exams and educational materials.
  + Scheduling and appointment management system.
  + User accounts with different levels of access (admin, secretary, and customer).
  + Tracking and reporting systems for student progress.
  + Secure login and role-based access controls.
  + Interface design for both students and admins with user-friendly navigation.

### Objectives and Goals

*What should this system be able to do when it is completed? What measurable tasks need to be included in the system design to achieve this?*

**Measurable Tasks to Include:**

* **Customer Management:**
  + Allow customers to register, log in, and manage their profiles.
  + Enable customers to book, modify, and cancel driving lessons online.
  + Provide students access to online classes and driving tests.
* **Admin Management:**
  + Enable admin users (Liam, Ian) to view and manage customer information, lesson schedules, and reporting data.
  + Admins should have the ability to reset passwords, manage accounts, and oversee system performance.
* **Lesson and Appointment Management:**
  + Allow secretaries to schedule, track, and modify driving lessons.
  + Provide functionality to track the status of lessons (completed, upcoming, canceled).
* **Reporting and Tracking:**
  + Implement a system to track customer progress through exams and driving lessons.
  + Track user activity (who scheduled, modified, or canceled lessons).
* **Security & Access Control:**
  + Different roles with access to specific features (admin, secretary, customer).
  + Ensure all user data is secure with role-based permissions.
* **Flexibility for Future Needs:**
  + Ability to add new training modules or update lessons as needed.
  + Integrate DMV updates and policies into the system for compliance.

## Requirements

### Nonfunctional Requirements

*In this section, you will detail the different nonfunctional requirements for the DriverPass system. You will need to think about the different things that the system needs to function properly.*

#### 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 DriverPass system will be web-based and accessible via browsers on both desktop and mobile devices.  
• The system should load key pages (dashboard, scheduling, tests) within 3 seconds under normal load.  
• The system should support at least 100 concurrent users without performance degradation.  
• DMV updates (rules, tests) should be reflected in the system within 24 hours.  
• Availability should be 99.9% with daily system health checks.

#### 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 platform will be hosted in the cloud (e.g., AWS or Azure) and be OS-independent from the user's perspective.  
• The back end will require a relational database (e.g., MySQL or PostgreSQL) for storing user info, lesson data, and reports.  
• The front end must work in modern browsers (Chrome, Firefox, Safari, Edge).  
• Backend tools may include frameworks like Node.js or Python, with REST API integration.

#### 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?*

• Users will be identified by a unique email address, and roles will define access levels.  
• All login inputs (usernames, passwords) will be case-sensitive.  
• The system will log all modifications and notify the admin of system errors, appointment conflicts, or failed logins after 5 attempts.

#### 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?*

• User management (add/remove/update roles) will be done through an admin interface without needing to change the code.

• The system will adapt to platform/browser updates by using responsive design and up-to-date frameworks.

• The IT admin will have full system access, including resetting passwords, unlocking accounts, and enabling/disabling features.

#### 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 must log in with a secure username and password.  
• All data transfers will use HTTPS with SSL encryption.  
• After 5 failed login attempts, the account will be locked and flagged for admin review.  
• A secure password recovery/reset feature will be available via verified email.

### 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 allow users to create an account with their contact and payment information.

• The system shall allow users to select and purchase one of three training packages.

• The system shall allow customers to schedule, modify, or cancel appointments.

• The system shall allow trainers to view their assigned appointments and leave comments on each session.

• The system shall allow customers to take practice tests and view their progress.

• The system shall track all system activity (scheduling, cancellations, comments) with time stamps and user ID.

• The system shall notify admins when DMV updates are received or when issues occur.

• The system shall allow secretaries to make changes to appointments via phone or in-person requests.

• The system shall allow reports to be generated and downloaded in Excel format.

• The system shall allow IT/admin users to manage access, lock/unlock accounts, and perform maintenance.

### 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 must be intuitive and responsive on both desktop and mobile browsers.  
• Different user roles will access different dashboards:

* Customers will register, view appointments, access online materials, and take practice tests.
* Trainers will view assigned sessions and leave lesson notes.
* Secretaries will have tools to manage customer schedules and registration.
* Admin/IT will manage users, reset accounts, and generate reports.
* Owner will access all reports and have read-only visibility over the whole system.  
  • Users will interact primarily via a web browser; mobile-friendly design is required.

### 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 have access to a stable internet connection and a modern browser.  
• The DMV will provide API access or some standard method to notify of updates.  
• Secretaries and trainers will receive basic training on using the system.  
• No in-person technical support will be required—the system will be intuitive enough for users to navigate independently.

### 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?*

• Limited time and budget may affect how many features can be built in the first release.  
• The team may not have access to real DMV data, so mock data may be used during development.  
• Real-time syncing with DMV systems may depend on external access or permissions.  
• Since the platform is entirely cloud-based, any internet outage may disrupt usage for all users.  
• No mobile app will be developed initially; only browser-based mobile access will be supported.

### Gantt Chart

*Please include a screenshot of the GANTT chart that you created with Lucidchart. Be sure to check that it meets the plan described by the characters in the interview.*

A graph with a pink rectangular object

AI-generated content may be incorrect.