# CS 255 Business Requirements Document

Amelia Sivick

October 6, 2025

## 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?*

* Deliver a single, web based system that combines online practice exams, downloadable progress reports, and on road lesson scheduling
* Enabling multiple user roles like administrators, IT officers, secretaries, and students to access features through secure login
* Provide access from any computer or mobile device, with offline viewing of downloaded data (reports, test results)
* Flexible lesson packages, appointment management, and real time tracking of driving sessions in one centralized platform

### 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: Many students fail their DMV driving tests due to fragmented or inadequate preparation resources
* DriverPass offers a one stop training solution that pairs interactive online classes and practice tests with in car lessons guided by certified trainers
* Core components needed:

– User account module with role based permissions and password reset functionality

– Reservation system for booking, canceling, and modifying driving sessions, including package selection

– Session tracking interface to log start/end times and driver comments for each lesson

– Progress dashboard that displays test history, scores, and status (not taken, in progress, failed, passed)

– Reporting engine for activity logs, audit trails, and Excel compatible exports

– Integration layer to receive updates from the DMV on rules and sample questions

– Cloud hosted web front end and backend services to handle data storage, security, and scalability

### 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?*

* Account Management: – Allow users to register, authenticate, and reset passwords online – Grant IT officer full account control (create, modify, disable) and restrict student/secretary to their own data
* Lesson Reservation and Packages: – Implement three predefined packages with the ability to disable new registrations for any package – Enable students and secretaries to schedule 2-hour sessions by selecting date, time, driver, and pickup/drop-off location
* Session Tracking and Feedback: – Capture lesson start/end times, driver comments, and link each session to a specific car and trainer
* Progress Monitoring: – Display a table of online tests showing test name, time taken, score, and status flags – Offer offline download of progress reports in Excel format
* Reporting and Auditing: – Log every reservation change, test attempt, and account modification with user ID and timestamp – Provide printable activity reports to identify who performed each action
* DMV Integration: – Receive push notifications of policy or question updates from the DMV and surface alerts in the admin dashboard
* Security and Compliance: – Enforce HTTPS, role-based access control, and data encryption at rest and in transit – Ensure the system meets industry standards for user privacy and regulatory compliance
* Scalability and Availability: – Host on a cloud platform with auto-scaling, load balancing, and multi-zone redundancy to support growth and high availability

## 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?*

* Must run as a cloud-hosted, web-based application accessible via modern desktop and mobile browsers.
* Average page load and booking response time under 2 seconds during normal load.
* System maintenance and minor updates deployed weekly; major feature releases on a quarterly cycle.

#### 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?*

* Front end compatible with Windows, macOS, Linux, iOS, and Android browsers (Chrome, Safari, Firefox, Edge).
* Back end hosted in Docker containers on a cloud platform (e.g., AWS, Azure) using a relational database (PostgreSQL or MySQL).
* Requires web server (Nginx/Apache) and API framework (Node.js, Java, or .NET).

#### 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?*

* User identity distinguished by unique email addresses (case-insensitive) and case-sensitive passwords.
* Reservation engine must detect and prevent any overlapping bookings for trainers or vehicles.
* System logs and notifies administrators immediately if any database constraint or scheduling conflict occurs.

#### 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?*

* Administrators can enable, disable, or modify training packages via the UI without redeploying code.
* Modular microservices architecture to accommodate future platform upgrades or third-party integrations.
* IT officer granted super-admin privileges to configure system settings, manage service endpoints, and view logs.

#### 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?*

* Login requires a valid email and password; passwords hashed and salted (bcrypt or equivalent).
* All data exchanged over TLS/SSL; PCI-compliant handling for credit-card fields.
* After five consecutive failed login attempts, user account locks for 15 minutes and triggers an alert to the IT officer.
* ‘Forgot password’ workflow uses time-limited, single-use reset tokens sent via 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 validate user credentials during login.
* The system shall allow users to reset forgotten passwords through a secure, token-based email workflow.
* The system shall enable new user registration, capturing personal details and payment information.
* The system shall let customers and secretaries book two-hour driving sessions, specifying date, time, pickup, and drop-off locations.
* The system shall prevent double-booking of the same car or trainer.
* The system shall permit users to modify or cancel reservations up to 24 hours before the lesson.
* The system shall provide an administrative interface to enable or disable training packages on demand.
* The system shall display an online testing dashboard with test name, elapsed time, score, and status (not started, in progress, failed, passed).
* The system shall allow trainers to view their scheduled lessons and record start/end times plus free-form comments.
* The system shall integrate with a DMV content feed to receive rule and policy updates and notify administrators on arrival.
* The system shall log every create, update, or delete action on user accounts and reservations, capturing user ID and timestamp.
* The system shall generate and export activity reports in Excel or CSV format.
* The system shall support offline, read-only access to previously exported 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.)?*

* Responsive web UI accessible via desktop and mobile browsers.
* Customer portal: registration, profile management, booking calendar, test progress dashboard.
* Secretary portal: full reservation management (create/modify/cancel), customer lookup, lesson scheduling form.
* Trainer portal: view daily assignments, lesson details form for comments and time logs.
* Admin console: user and role management, package configuration, DMV update log, audit-report generator.
* All interfaces follow WCAG 2.1 accessibility guidelines and use consistent branding and navigation.

### 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?*

* All users have reliable internet access for online booking and testing.
* Customers and staff use modern web browsers; no legacy-only support required.
* DriverPass will provide or subscribe to a structured DMV update feed (e.g., API or RSS).
* Payment processing handled by a third-party gateway integrated via API.
* Secretary and trainers are comfortable using a browser-based UI for daily tasks.

### 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?*

* Offline mode is strictly read-only; no offline booking or profile updates.
* No native mobile or desktop client; functionality is limited to responsive web.
* Initial rollout supports only three standard packages; custom-package creation by non-developers slated for a later phase.
* Development constrained to a 15-week schedule and a small team, limiting scope of advanced analytics or machine-learning features.
* Integration limited to a single DMV content source; additional jurisdictions require separate development.

### 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.*

