# CS 255 Business Requirements Document Template

## 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, a company aiming to improve driving test pass rates for students.
* Purpose: To design and implement a system that provides online practice exams and on-the-road training scheduling.
* The system should support students, instructors, administrators, and integrate with external systems like the DMV.
* Goals include improving the quality of training, streamlining scheduling, and enabling secure online payments.

### 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: Over 65% of students fail their driving license exams due to inadequate preparation and reliance solely on memorizing old tests.
* DriverPass needs a platform that:
  + Offers online practice tests based on current DMV standards.
  + Allows students to schedule and manage driving lessons.
  + Provides instructors with tools for tracking and updating student progress.
  + Enables administrators to manage training packages, users, and payments.
  + Integrates with the DMV for regulatory updates and test requirements.
* Components needed:
  + Web-based user interface accessible via desktop and mobile devices.
  + Secure payment processing system.
  + Scheduling module for booking and modifying lessons.
  + Practice exam module with scoring and feedback.
  + Administrative dashboard for management tasks.

### 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?*

* Deliver a fully functional online and on-the-road training system within the agreed project timeline.
* Measurable objectives:
* Students can register and create an account.
* Students can schedule, modify, and cancel lessons online.
* Students can take practice tests and review scores.
* System sends notifications and reminders for appointments.
* Securely process online payments for training packages.
* Instructors can update lesson notes and track student progress.
* Administrators can manage training packages and view reports.
* System syncs with DMV to keep course material updated.

## 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 must be fully web-based and accessible via any modern browser (Chrome, Firefox, Safari, Edge) without requiring additional software installation.
* Pages should load within 2 seconds under normal network conditions.
* All transactions (e.g., scheduling lessons, processing payments) must process in under 5 seconds.
* The system shall perform daily automated backups to prevent data loss.
* Software updates, including DMV regulation imports and bug fixes, shall be applied within 48 hours of release to minimize downtime.

#### 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 shall run on Windows Server or Linux (Ubuntu/CentOS) hosting environments.
* The system back end must use a relational database management system (MySQL or PostgreSQL) to store user profiles, schedules, and test results.
* The application must be compatible with mobile devices via responsive design but will not require separate native iOS/Android apps for launch.
* The payment integration must support PCI-compliant third-party services (Stripe, Square).

#### 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 shall be uniquely identified by a system-generated user ID to prevent duplication, with email addresses serving as unique login credentials.
* Input for credentials (usernames, passwords) will be case-sensitive to improve security.
* The system must notify administrators of critical errors (failed payment transactions, DMV data import failures) via email and in-system alerts.
* Practice test scoring must have 100% accuracy, with immediate calculation and storage in the student’s profile.

#### 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 shall be able to add, remove, or modify user accounts, training packages, and test content without code changes, through a secure admin interface.
* The system must adapt to browser and operating system updates without requiring downtime.
* The IT administrator shall have full role-based access to adjust system settings, configure user roles, and manage API integrations.
* The system must be designed to scale, supporting up to 10,000 concurrent users without performance degradation.

#### 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?*

* All users must log in using unique credentials and multi-factor authentication (MFA) shall be available for high-privilege roles (Admin, IT).
* All client-server communication shall be encrypted using TLS 1.3 or higher.
* Accounts will be temporarily locked after five failed login attempts to prevent brute force attacks.
* Passwords must meet strong complexity requirements a minimum of 12 characters, mixed case, numbers, symbols. Password can be reset via a secure password recovery process.
* Payment processing must comply with PCI-DSS standards, and sensitive data such as passwords and payment details must be encrypted at rest and in transit.

### 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 when logging in, using secure authentication methods.
* The system shall allow students to register for an account by entering personal, contact, billing, and payment information.
* The system shall enable students to schedule, reschedule, or cancel on-the-road driving lessons.
* The system shall allow instructors to view, update, and manage their lesson schedules.
* The system shall deliver online DMV practice exams, automatically score them, and store results in the student’s profile.
* The system shall process payments for training packages through a secure, PCI-compliant payment gateway.
* The system shall send automated email or SMS notifications for lesson reminders, payment confirmations, and DMV updates.
* The system shall allow administrators to add, edit, or remove training packages and practice exam content.
* The system shall integrate with the DMV system to import regulation and question updates.
* The system shall allow office staff to manage student records, payment transactions, and instructor assignments.

### 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 accessible via modern desktop browsers Chrome, Firefox, Edge, Safari and mobile devices responsive design.
* Students: Must be able to schedule lessons, take practice exams, view results, purchase packages, and view payment history.
* Instructors: Must be able to view schedules, review student profiles, and enter lesson notes.
* Office Staff: Must be able to manage schedules, process payments, and update student or instructor records.
* Administrators: Must be able to manage all user accounts, adjust system settings, and update training packages.
* The interface shall include role-specific dashboards, so users see only relevant tools and information.
* Navigation should be simple and intuitive, with clear labels and consistent formatting across all pages

### 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 will have reliable internet access to use the web-based platform.
* Users will be able to access the system from a modern device (desktop, laptop, tablet, or smartphone) with an up-to-date browser.
* The DMV system will provide timely and accurate regulation updates in a compatible format.
* Third-party payment processors like Square and Stripe will be available for integration.
* Students will provide accurate personal and billing information during registration.

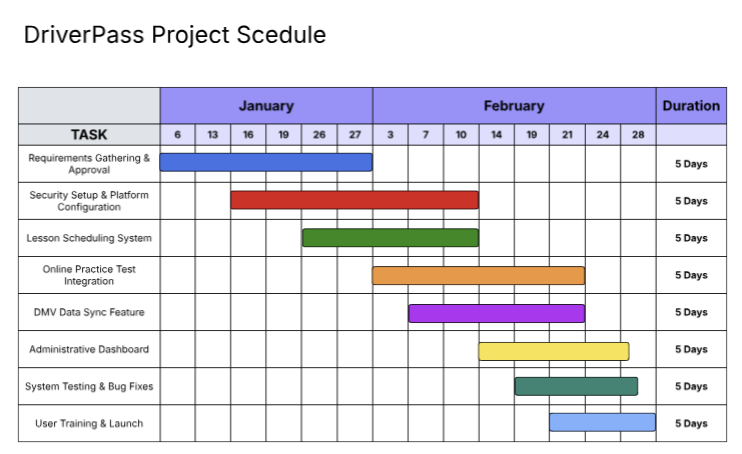
### 

### 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?*

* The system will not provide offline access to practice tests or scheduling features.
* On-the-road lessons require instructor availability, which may limit scheduling flexibility.
* The budget restricts the initial launch to English-language support only, with no multilingual features.
* The project timeline is limited to the agreed-upon development schedule, impacting on the number of additional features in the first release.
* DMV system integration depends on their update schedule and API availability, which may cause delays in applying changes.

### Gantt Chart

**