# CS 255 Business Requirements Document KING

## 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?*

* The purpose of this project is to develop a system for DriverPass that provides both online and on-the-road driver training services.
* The system aims to address the market gap where 65% of students fail DMV tests due to inadequate preparation.
* The client, DriverPass, wants to improve the pass rates of DMV driving tests by offering online classes, practice tests, and scheduled driving lessons.
* The system needs to support scheduling, user access control, report generation, and integration with DMV updates.

### 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?*

* DriverPass wants to solve the problem of low driving test pass rates by offering better preparation tools for students.
* The system shall allow students to register, schedule driving lessons, and access online learning content and practice tests.
* It must support appointment management, role-based user access (admin, IT, secretary, student), and secure data handling.
* The system should allow both online and in-office scheduling of lessons, support future customization of training packages, and provide progress tracking for each student.
* The system shall track assignments of students to specific drivers, cars, and lesson times for accurate scheduling.
* The system should allow drivers to upload comments about each lesson, which will be stored and displayed in a table alongside lesson times for administrative and student access.
* The system should support three predefined training packages: Package One (six hours of in-car training over three sessions), Package Two (eight hours of in-car training over four sessions plus an in-person DMV rules lesson), and Package Three (twelve hours of in-car training over six sessions, an in-person DMV rules lesson, and access to online classes with practice tests).
* Users should be able to reset passwords, receive DMV content updates, and access the platform from web-enabled devices.

### 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?*

* The system should enable 100% of students to register online with secure data storage within 5 minutes.
* The system should allow students to schedule, modify, or cancel driving lessons via their accounts in under 2 minutes per action.
* The system should track assignments of students to specific cars, trainers, dates, and times with 100% accuracy.
* The system should provide access to online training materials and practice tests, with progress tracking updated within 1 second of test completion.
* The system should allow admins and IT staff to manage user accounts, reset passwords, and monitor system activity within 1 minute per action.
* The system should generate downloadable activity and reservation reports within 10 seconds of request.
* The system should reflect DMV updates regarding test requirements or materials within 24 hours of notification.
* The system should allow administrators to disable training packages within 1 minute without code changes.
* The system should ensure secure login, encrypted data handling, and password recovery with 99.9% uptime.
* The system should validate that pickup and drop-off locations for lessons are identical, ensuring compliance with client requirements.

## 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 system should run as a web-based application hosted in the cloud to allow access from computers and mobile devices.

#### It should load pages and respond to user actions within 2 seconds under normal operating conditions.

#### The system should support simultaneous access by multiple users without performance degradation.

#### The system should handle up to 1,000 simultaneous users during peak times without performance degradation.

#### System updates, including security patches and content updates, should occur at least once a month or as needed to stay current with DMV changes.

#### 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 should be compatible with major web browsers (Chrome, Firefox, Edge, Safari) on both desktop and mobile devices.

#### It should operate across different operating systems including Windows, macOS, iOS, and Android.

#### The back end should be supported by a relational database to store user, appointment, and test data.

#### Cloud infrastructure should be used to handle hosting, data storage, and system scalability.

#### 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?*

#### The system should distinguish users based on unique login credentials such as username and password.

#### User inputs (e.g., names, dates, pickup locations) should not be case-sensitive, but must be validated for accuracy (e.g., correct format, no missing required fields).

#### The system should log and track all changes to records, including who made each change and when.

#### The system should notify admins via email or dashboard alerts within 1 minute of detecting issues like failed logins or reservation conflicts.

#### 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 should allow administrators to enable or disable training packages without needing to modify the code.
* The system should support role-based access so users can be added, removed, or modified as needed.
* The IT admin should have full access to manage user accounts, reset passwords, and block users when necessary.
* The system should be designed to adapt to platform updates and maintain compatibility with modern browsers and mobile devices.
* Future package customizations (adding or removing modules) should be planned for potential future development phases.

#### 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?*

* The system should require users to log in with a username and password to access their account.
* The system should support password recovery through a secure, automated reset process.
* The system should implement secure connections (HTTPS) to encrypt all data exchanges between the client and server.
* The system should lock a user account after multiple failed login attempts to prevent brute-force attacks.
* The system should use role-based access control to ensure users only access information relevant to their role.
* Sensitive data such as credit card information should be encrypted and stored securely.
* The system should comply with PCI DSS standards for secure storage and processing of credit card information (Payment Card Industry Security Standards Council, 2022).

### 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 should allow customers to create, modify, and cancel driving lesson reservations online.
* The system should allow secretaries to schedule appointments on behalf of customers.
* The system should allow administrators to manage user accounts, including password resets and access control.
* The system should track and log user actions, such as who made or changed a reservation.
* The system should enable customers to select from three training packages, each with specific in-car hours, in-person lessons, and online content as defined in the system background.
* The system should allow drivers to input and save comments for each lesson, accessible in a table format for administrators and students.
* The system should display test progress, scores, and status for each student in the online course section.
* The system should allow the input and storage of customer details, including name, contact info, and payment information.
* The system should generate downloadable reports and activity logs for administrative use.
* The system should support communication between users and DriverPass staff through contact forms.
* The system should sync with DMV updates and notify staff when new requirements or questions are posted.
* The system should validate that pickup and drop-off locations for driving lessons are identical.

### 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 accessible via both desktop and mobile browsers.
* Users of the interface will include administrators, IT staff, secretaries, customers, and drivers.
* Administrators should have full access to manage accounts, monitor system activity, and generate reports.
* IT staff should require backend access for system updates, troubleshooting, and security management.
* Secretaries should have access to schedule, modify, and cancel appointments on behalf of customers.
* Drivers should have access to input and receive comments about each lesson, which will be displayed in a table with lesson times and other details.
* Customers should log in to view their selected training package details (e.g., in-car hours, in-person lessons, online content), schedule appointments, take practice tests, and track their progress.
* The interface should be simple, clear, and user-friendly, allowing customers to navigate training options and track their test results easily.
* Forms should be provided for data entry such as registration details, appointment scheduling, and contact submissions.
* The interface should clearly show package details, lesson times, driver comments, and test statuses.
* The interface should comply with WCAG 2.1 standards to ensure accessibility for all users (World Wide Web Consortium, 2018).

### 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?*

* It is assumed that all users will have access to a reliable internet connection to interact with the web-based system.
* It is assumed that users have access to modern web browsers that support the system's interface.
* The customer will provide current DMV policies and practice test updates regularly.
* It is assumed that users such as secretaries and administrators will receive proper training on how to use the system.
* The system will rely on cloud hosting for uptime, backups, and security.
* User authentication will be handled through standard email and password credentials, with password reset capabilities.
* It is assumed that credit card transactions will be processed using a secure third-party service.
* It is assumed that the third-party payment processor will maintain 99.9% uptime for credit card transactions.

### 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?*

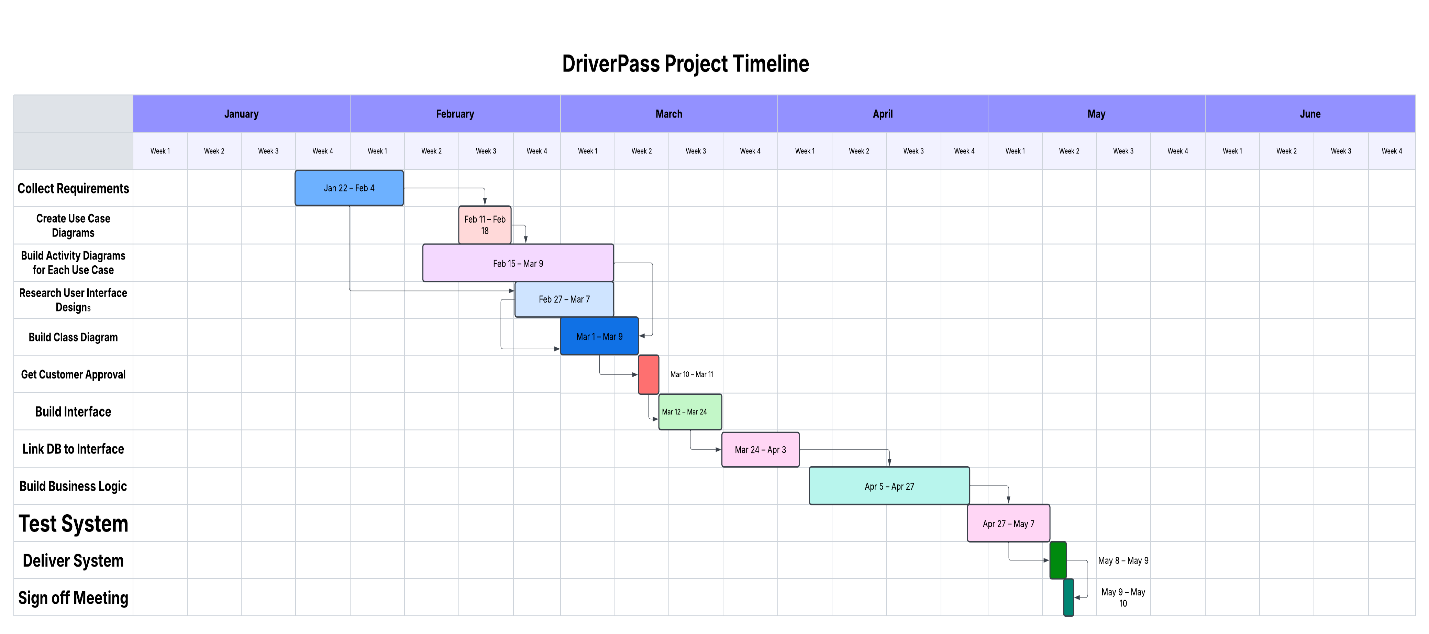
* The system cannot be updated or modified offline; changes require an active internet connection.
* Customization of packages (adding/removing features) requires developer intervention and cannot be done by nontechnical users.
* The initial version will not include real-time integration with the DMV for automatic updates; manual updates will be required.
* Budget constraints may limit advanced features like mobile apps or live chat support in the first release.
* The system is limited to web-based interaction and does not support standalone desktop applications currently.
* Any changes in DMV policy must be communicated manually unless future API integration is developed.
* Training for staff is assumed but not included in the current project scope.
* Limited time for user training may require a simplified interface to reduce the learning curve.

**References**

Payment Card Industry Security Standards Council. (2022). *PCI DSS v4.0.* Retrieved from <https://www.pcisecuritystandards.org/document_library>

World Wide Web Consortium. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1.* Retrieved from <https://www.w3.org/TR/WCAG21/>

**Gantt chart**



[Chart Link](https://lucid.app/lucidchart/5cb66ced-f4c3-4ac1-80f9-62ceb3a757cc/edit?viewport_loc=-1712%2C-537%2C4348%2C1352%2C0_0&invitationId=inv_39c8538c-12b7-44c1-8125-1577ee990a9f)