# CS 255 Business Requirements Document Template

Complete this template by replacing the bracketed text with the relevant information.

This template lays out all the different sections that you need to complete for Project One. Each section has guiding questions to prompt your thinking. These questions are meant to guide your initial responses to each area. You are encouraged to go beyond these questions using what you have learned in your readings. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead, the goal is to complete each section based on your client’s needs.

**Tip:** You should respond in a bulleted list for each section. This will make your thoughts easier to reference when you move into the design phase for Project Two. One starter bullet has been provided for you in each section, but you will need to add more.

## 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 create a user-centered system that helps individuals prepare for and successfully pass their diving exams. This system aims to streamline the learning process by offering access to instructional materials, scheduling tools, and progress tracking features.

* Liam (Driver Pass boss) and Ian (IT Lead) want to build a system that improves driver training outcomes by combining digital and hands-on learning.
* The system should provide access to training modules, virtual classes, and optional on-the-road lessons to support different learning styles.
* DriverPass aims to help customers who struggle with DMV regulations and road rules by offering clear, accessible resources.
* The ultimate purpose is to increase the number of students who pass their driving test on the first try by offering structured, supportive training.

### 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?*

The client is a driving education company that wants to improve student outcomes and modernize how driving instructions is delivered.

* DriverPass wants to streamline driver education by offering centralized access to training materials, scheduling tools, and progress tracking.
* The current system suffers from poor structure, limited device compatibility, and weak access controls managed by only two team members.
* Students face difficulties navigating training packages and scheduling lessons due to inconsistent organization and lack of automation
* Key components needed to solve these issues include:

1. A user- friendly interface with clear navigation and structure
2. Robust scheduling tools that allow students to book lessons and track progress
3. Strong security and accessibility control features to protect user data and manage permissions
4. Cross- device compatibility to ensure accessibility from mobile devices, tablet, and desktop platforms

### 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?*

* Provide smooth and reliable scheduling tools for booking lessons, tracking progress, and managing training packages
* Offer 24/7 customer support through live chat or self- help portals for password resets, package assistance, and general inquiries
* Include DMV- related resources such as rulebooks, practice tests, and a DMV portal link to help users schedule their road test after completing training
* Track student progress through visual dashboards or progress bars to help instructors and students stay aligned
* Ensure the system is scalable and secure, with role-based access controls and audit logs for administrative oversight
* Improve customer satisfaction by reducing friction in booking, learning and testing processes- measurable through feedback surveys and pass rates

## 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 must run in web-based environments, accessible via modern browsers such as:

* Google Chrome
* Firefox
* Safari
* Microsoft Edge

It should also be compatible with mobile devices, including Android and iOS, through responsive design or a dedicated application. In addition, the system should support both desktop and tablet interfaces for the instructors and administrators. Like any expectations with loading pages, loading time period should be < 2 seconds with normal network conditions. User actions (e.g. navigation lessons) should respond within 1 second to ensure smooth interaction. With current updates for applications, the frequency is held every month due to updates on software security or added features. We will include:

* Minor updates (bug fixes, UI tweaks) should be deployed bi-weekly
* Major updates (new features, module additions) should be released quarterly, with minimal downtime
* System should support automated deployment pipelines to ensure fast, reliable updates

#### 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?*

This system should promote cross- platform and great compatibility with:

* Windows (used by most users and instructors)
* MacOS (mostly used by students using Apple products)
* Linux/ Unix (for sever-side deployment and developers)

The application should be accessible via modern web browsers to ensure broad usability. I recommend using a relational database, such as MySQL, to efficiently manage and store critical system data including user profiles, lesson schedules, progress tracking and administrative logs. We can also include an optional approach with Cloud Hosting for scalability and reliability.

#### 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?*

In order to distinguish between different users, they each will have a unique login ID with a secure password such as:

* Email
* Username

The system will use RBAC (role- based access controls) to distinguish between students, instructors and administrators. User sessions will be tracked with secure tokens to prevent mix-ups or unauthorized access. For example, to align with security standards commonly used in military systems, all user passwords must meet strict complexity requirements. By incorporating this practice, this ensures account protection and prevents unauthorized access. This approach ensures robust authentication and reflects best practices in high- security environments.

Each password must include:

* At least one uppercase letter
* At least one lowercase letter
* At least one number
* At least one special character or symbol (e.g., !, $, ^, $)

The system should alert the admin when:

* There are repeats of failed login (5+ times from the same IP address)
* Schedule lesson fails to confirm or sync
* Critical system errors or crash occur
* Data integrity issues are detected such as missing progress records

#### 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 will support dynamic user management through an admin dashboard or RBAC interface. This will allow IT administrators to **add, remove or modify** users- including roles, permissions, and profile data- without altering the underlying codebase. Of those changes, it will reflect in real time across the platform. For platform update resilience, to ensure smooth adaptation to platform updates, the system will:

* Use modular architecture
* Implement automated testing and continuous integration pipelines to detect and resolve compatibility issues early on
* Follow sematic versioning and backward compatibility where it’s feasible

IT administrators will require:

* Secure login credentials
* Access to a web-based admin panel for managing users, viewing logs, and configuring settings
* Audit trail visibility to monitor changes and ensure accountability

#### 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?*

To incorporate proper security, users must provide:

* A unique username or email address
* A secure password that meets complex requirements
* Optionally, MFA (multi- factor authentication) for added protection

To protect data in transit (securing client- server communication):

* All communication will be encrypted using HTTPS
* Sensitive data will be hashed and salted before storage using algorithms
* Session tokens will be used for authenticated requests, with expiration and refresh mechanisms

If user forgets their password:

* They can initiate a secure password reset via email, SMS or secure personal questions
* A time-limited token will be sent to verify identity before allowing password change
* The system will enforce password history checks to prevent reuse of recent passwords

### 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 logged in
* The system shall allow students to register for DriverPass account using a valid email and password
* The system shall enable students to schedule driving lessons with available instructors
* The system shall display instructor availability based on location and calendar
* The system shall track student progress toward passing the written and driving exams
* The system shall provide access to practice written exams with automated scoring and feedback
* The system shall allow instructors to update lesson completion status and performance notes
* The system shall send automated reminders to students for upcoming lessons and exams
* The system shall allow instructors to view and manage their schedules
* The system shall enable secure messaging between instructors and students
* The system shall allow admins to approve or deny structure applications based on submitted credentials
* The system shall generate performance reports for instructors, students, and overall program metrics
* The system shall allow admins to manage pricing, lesson packages, and system settings
* The system shall log all user activity for auditing and troubleshooting purposes.
* The system shall support password recovery via email verification

### 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 system will provide a responsive, browser- based interface accessible via desktop, tablet, and mobile devices. The design will prioritize usability, accessibility, and role-specific functionality.

**Student interface (mobile and browser)**

Key Actions:

* Register and log in securely
* Schedule driving lessons with available instructors
* View upcoming lessons and test dates
* Take practice written exams and receive feedback
* Track progress towards licensing goals
* Communicate with instructors via messaging
* Access account setting and password recovery

Design Considerations:

* Simple navigation with dashboard overview
* Visual progress tracker and calendar integration
* Mobile- friendly layout for on-the-go access

**Instructor Interface (Browsers (desktop), mobile-compatible)**

Key Actions:

* Log in and manage schedule availability
* View student lesson bookings
* Update student progress and lesson notes
* Communicate with students securely
* Access performance metrics and feedback

Design Considerations:

* Calendar- based scheduling tools
* Quick access to student profiles
* Secure messaging and notification system

**Admin Interface (Browser (desktop preferred))**

Key Actions:

* Review and approve instructor applications
* Managed lesson pricing and system settings
* Generate reports on student and instructor performance
* Monitor system usage and troubleshooting
* Access audit logs and user activity

Design Considerations

* RBAC
* Data visualization for reporting
* Administrative dashboard with system alerts

### 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 users will be able to navigate the DriverPass platform independently on Android or Ios devices. However, the current design may not fully support users who are not tech-savvy or who do not regularly engage with mobile applications, potentially limiting accessibility and ease of use for less experienced individuals. Other user-related assumptions:

* Users understand basic English, as the platform’s content and instructions are currently only available in English
* Users will self-report issues or confusion via built-in feedback or support channels
* Users are motivated to learn and will engage with both the instructional content and practice tests without external enforcement
* Users have access to a stable internet connection and a device capable of running the DriverPass platform

Technology-Related Assumptions:

* The platform will be accessed via modern web browsers
* The system will integrate smoothly with certain databases for scheduling and test verification
* The DriverPass will be hosted on a secure, scalable cloud infrastructure with regular backups and uptime guarantees
* No assistive technologies (voice navigation) area currently supported, though accessibility improvements are planned

### 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 Design Limitations:

* The platform currently lacks personalized learning paths or adaptive testing, which can reduce effectiveness for users with varying skill levels
* There is limited accessibility support, which may exclude users with disabilities
* Integration with DMV systems for scheduling and verification is assumed but not yet implemented, which may delay or complicate real-world deployment
* The system does not yet include real-time instructor feedback or have live support, relying instead on static content and automated assessments

Resource, Time, and Budget Constrained:

* Development is constrained by a limited timeline, which restricts the scope of features and testing
* The project operates under assumed, modest budget, limiting investment in advanced technologies like AI-driven tutoring or full mobile appl development
* The team may have limited access to user testing groups, which may affect the accuracy if usability and engagement predictions
* Technical resources are focused on core functionality, leaving less room for enhancement like multilingual support or offline access

### Gantt Chart

A screenshot of a computer

AI-generated content may be incorrect.