# 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 design a system for DriverPass that will help students prepare for their driving tests more effectively
* The goal is to create a comprehensive solution that combines online practice exams and on-the-road training to give students the tools and skills needed to succeed in both the written and practical driving exams.
* The client is DriverPass
* DriverPass wants the system to provide online practice exams, offer scheduling for on-the-road training, support user account management, and be accessible and user-friendly.
* The system must be secure, scalable, and adaptable to meet the needs of a diverse user base.

### 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 problem is a lack of comprehensive tools that combine practice exams, real driving experience, and progress tracking.
* DriverPass wants to fix this gap by creating a system that helps students become better prepared for both written and practical driving tests.
* DriverPass would need a user interface, online practice exam module, scheduling module, progress tracking module, account management system, and IT and security infrastructure for this system.

### 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?*

* Develop the online practice exam module, create a question database with detailed tagging for categorization and implement a results tracking system to log student performance.
* Build the scheduling module, develop a calendar system for managing instructor availability and implement a booking interface for students to select lesson times.
* Create the progress tracking system, a data logging feature to track practice test results and driving lesson completion.
* Implement the account management system, enable account creation and authentication with secure passwords and integrate role-based access for students, instructors, and administrators.
* Establish IT and security measures by implementing firewalls and intrusion detection to protect the system.
* Develop a student-friendly dashboard for easy navigation of practice tests, schedules, and progress.

## 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 would need to run in a web-based environment. Accessible via web browsers for both students and staff and responsive design to support desktops, laptops, tablets, and smartphones.
* Pages should load within 2-3 seconds under normal usage conditions.
* Responses to actions like submitting a practice test or booking a lesson should take less than 1 second.
* Perform minor updates weekly or biweekly, depending on the criticality of issues.

#### 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 would run on Windows which is compatible with modern browsers.
* The back end of the system will require several tools to support its functionality like MySQL to store structured data such as user accounts and roles, practice exam questions and results, and scheduling data for driving lessons.
* An application server, and optional cloud hosting services to ensure robust performance, scalability, and data management.

#### 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 will distinguish between different user types using role-based access control**.**
* Each user will have a role assigned during account creation, which determines their permissions and access to system features.
* Usernames or emails will be converted to lowercase during both account creation and login to avoid case-sensitivity issues.
* Passwords will retain case sensitivity to help with security.
* The system should notify the administrator if the system cannot connect to the database or experiences a query error or detection of unusual activities such as unauthorized data export or access to restricted files.

#### 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 be designed to allow administrators to manage users without requiring modifications to the codebase.
* The system will use modern development practices and tools to remain compatible with updates to underlying platforms.
* Use tools like BrowserStack to test compatibility with updated browsers and operating systems.
* IT administrators will require high level access to manage and maintain the system effectively, including tools for troubleshooting, monitoring, and securing the system.

#### 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 allow users to log in using their registered email or username.
* Users must enter a strong, case-sensitive password.
* A second authentication step, such as a one-time passcode sent to email or SMS, can be required for added security.
* All data exchanged between the client and server must be encrypted using HTTPS to protect against eavesdropping and man-in-the-middle attacks.
* Encrypt sensitive data both in transit and at rest.
* After a set number of failed login attempts, lock the account temporarily to prevent brute force attacks.
* Notify the account owner via email about suspicious login attempts.
* Provide a "Forgot Password" link on the login page.
* Require users to enter their registered email or username for verification.
* Send an email notification to the user confirming that their password was reset.

### 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.
* The system shall allow users to reset their passwords via a secure "Forgot Password" process.
* The system shall allow administrators to add, modify, and remove user accounts without changing the code.
* The system shall provide students with access to practice exams that mimic the real driving test format.
* The system shall display immediate feedback on practice test answers to help users learn.
* The system shall track and store students’ progress and performance on practice exams.
* The system shall allow students to view available driving lesson slots based on their location.
* The system shall enable students to schedule, reschedule, or cancel driving lessons.
* The system shall track the status of scheduled lessons, such as completed or pending.
* The system shall encrypt all user data both in transit and at rest.
* The system shall use HTTPS to secure all communication between clients and the server.
* The system shall notify administrators of any detected security breaches or unauthorized access attempts.

### 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.)?*

* Students would need to log in to their account, take practice driving exams and receive instant feedback, view and track their progress on exams and lessons, schedule, reschedule, or cancel driving lessons.
* Students would be able to access via a mobile app or web browser, see a dashboard showing their schedule, progress, and test history, and have buttons for scheduling lessons and starting exams.
* Instructors need to log in to view their teaching schedule, have access to student profiles to track individual progress, receive notifications for new or updated lesson bookings.
* Instructors would be able to access via mobile app or web browsers, view calendar interface for managing schedules, dropdown menus and forms to update lesson statuses.
* Admin would need to manage user accounts, monitor system performance through a dashboard, view audit logs for tracking critical actions, generate reports on student performance and system usage.
* Administrators would be able to use primarily browser-based interface with admin dashboards and dropdown menus and data tables for user management and reporting.
* IT would need access system logs for troubleshooting and to manage database backups and server configurations.
* IT would be able to use browser-based admin panel for day-to-day task and direct server access via SSH or monitoring tools.
* The user will interact with the interface with functionalities like scheduling, exam practice, and notifications. Intuitive navigation optimized for touchscreens on mobile devices.

### 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?*

* The design does not explicitly address whether the system will support multiple languages or localization for different regions.
* If DriverPass plans to charge for lessons or exams with payment handling.
* Assumes students, instructors, and administrators are familiar with basic technology and can navigate web or mobile apps without extensive training.
* Assumes all users have a reliable internet connection to access the web or mobile application.
* Assumes the system will follow industry-standard security protocols.

### 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 is designed for DriverPass’s current needs but may face limitations in handling a significantly larger user base without additional infrastructure investments.
* The system relies heavily on internet connectivity, limiting access for users in areas with unreliable or no internet.
* The initial design supports major platforms but may lack compatibility with less common operating systems or older devices.
* Limited number of developers, designers, and testers can slow down the development process.
* Tight deadlines might limit thorough testing, leading to potential bugs or security vulnerabilities in the final product.

### 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 screenshot of a calendar

Description automatically generated