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 is to develop a comprehensive system for DriverPass that supports their new driver training business.

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 a system to help more people pass their DMV driving tests by offering online classes, practice tests, and in-person training. The system should let customers schedule and manage driving lessons online or through a secretary, assign drivers and cars, and support flexible training packages. It must track test progress, keep detailed driver notes, and provide secure, role-based access for employees. The system should log all user actions for accountability, allow data access online and offline, export reports, and automatically update with DMV changes. Finally, it should be a web-based, cloud-hosted platform that's easy to use and maintain.

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 completed system should let customers access training materials, schedule and manage lessons, and track test progress. Secretaries can handle appointments, and administrators control access and monitor activity. It must support training packages, assign drivers and vehicles, update automatically with DMV changes, and offer offline access and report exports. Key design tasks include user management, appointment scheduling, package control, progress tracking, activity logging, DMV integration, report generation, and a secure, cloud-based web interface. These features ensure the system meets all client needs efficiently.

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 needs to be web-based and cloud-hosted to allow easy access from any computer or mobile device without the need for local installations. It should provide a responsive and fast user experience, with quick loading times for pages, especially when scheduling lessons or accessing training materials, to ensure users can complete tasks smoothly. Updates should be applied regularly, particularly to incorporate the latest DMV rules and practice test changes, ideally with automatic syncing whenever new information is available. Additionally, routine maintenance and security updates should be scheduled periodically to keep the system secure and reliable without disrupting users.

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 designed to run on all major platforms since it is web-based, meaning it must work seamlessly across Windows, macOS, Linux/Unix, and mobile operating systems like iOS and Android through modern web browsers. On the back end, the system will require a robust database (such as MySQL, PostgreSQL, or a cloud-based database service) to store user data, appointments, packages, driver and vehicle information, and activity logs. Additionally, it will need server-side technologies and tools to handle authentication, data processing, and integration with external services like the DMV for automatic updates. Using cloud infrastructure will help ensure scalability, security, and ease of maintenance.

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?

Different users will be distinguished through role-based access control, where each user is assigned a specific role—such as customer, secretary, driver, IT officer, or administrator—with permissions tailored to their responsibilities. User accounts will require unique usernames or email addresses to ensure proper identification. The system's input fields, especially for usernames and passwords, should be case-sensitive to enhance security, while other inputs like names or addresses can be case-insensitive for usability. The system should inform the admin immediately if there are critical issues such as failed login attempts indicating possible security breaches, system errors that affect data integrity, unauthorized access attempts, or failures in syncing with DMV updates. Additionally, alerts should be sent for important events like password reset requests or system downtime to ensure prompt action.

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?

Changes to users—such as adding, removing, or modifying accounts and their permissions—should be possible through an admin interface or dashboard without needing to change the underlying code. This allows the IT admin to manage users dynamically and securely. To adapt to platform updates (e.g., browser, OS, or third-party service changes), the system should be built with modular, maintainable code and hosted on a cloud platform that supports easy updates and scalability. Regular testing and updates will ensure compatibility. The IT admin needs full access to the system for tasks like managing user accounts, resetting passwords, blocking or unlocking users, monitoring system health and security, applying patches or updates, and handling backups—essentially full control to maintain and secure the system effectively.

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 log in, users must provide a unique identifier (such as a username or email) and a password. The system should enforce strong password requirements to enhance security. To secure the connection and data exchange between the client and server, the system must use encrypted communication protocols like HTTPS (SSL/TLS) to protect data in transit from interception or tampering. In the event of a brute force attack—multiple failed login attempts—the system should temporarily lock the account after a set number of failed tries and notify the user and admin of suspicious activity, possibly requiring additional verification to unlock. If a user forgets their password, the system should offer a secure password reset process, typically involving sending a unique, time-limited reset link to the user's registered email, allowing them to create a new password safely.

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 login credentials.
- The system shall allow users to create and manage accounts.
- The system shall enable customers and secretaries to schedule, modify, and cancel lessons.
- The system shall assign drivers and cars to lessons.
- The system shall support multiple training packages and allow admins to enable or disable them.
- The system shall track and display test progress and scores.
- The system shall log all user actions for accountability.
- The system shall enforce role-based access control.
- The system shall provide secure password reset options.
- The system shall alert admins of suspicious activities.
- The system shall sync automatically with DMV updates.
- The system shall allow online and offline data access with report export.
- The system shall be web-based, cloud-hosted, and accessible on multiple devices.
- The system shall encrypt data to ensure security.
- The system shall have an intuitive, responsive interface.

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 needs to be user-friendly, intuitive, and responsive, supporting easy navigation for different user roles. It should clearly display relevant information such as lesson schedules, test progress, driver notes, and account settings. The interface must also ensure security features like login authentication and role-based access controls are seamless and unobtrusive.
The different users of the interface include:

- Customers who need to create accounts, schedule, modify, or cancel driving lessons, access online classes and practice tests, and view their test progress.
- Secretaries who manage appointments, input customer information, and assist with scheduling both online and by phone.
- Drivers who review their assigned lessons, schedules, and leave notes.
- IT Officers and Administrators who manage user accounts, control access permissions, monitor system activity, and handle security and maintenance tasks.

Users will primarily interact with the system via web browsers on desktops, laptops, tablets, and mobile devices, ensuring accessibility anytime and anywhere. The interface should be mobile-responsive to provide a consistent experience across 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?

Several aspects were not specifically addressed in the design, such as detailed error handling, user onboarding processes, accessibility features for users with disabilities, and the specific technologies or frameworks to be used for development. Additionally, there was no mention of backup and disaster recovery plans, system scalability for future growth, or how customer support will be integrated.

The design assumes that users have reliable internet access and are comfortable using web-based applications on various devices. It also assumes that users have basic digital literacy to navigate the interface and manage their accounts. On the technology side, it assumes that the cloud infrastructure will provide sufficient security, uptime, and scalability, and that integration with DMV systems is feasible and reliable.

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?

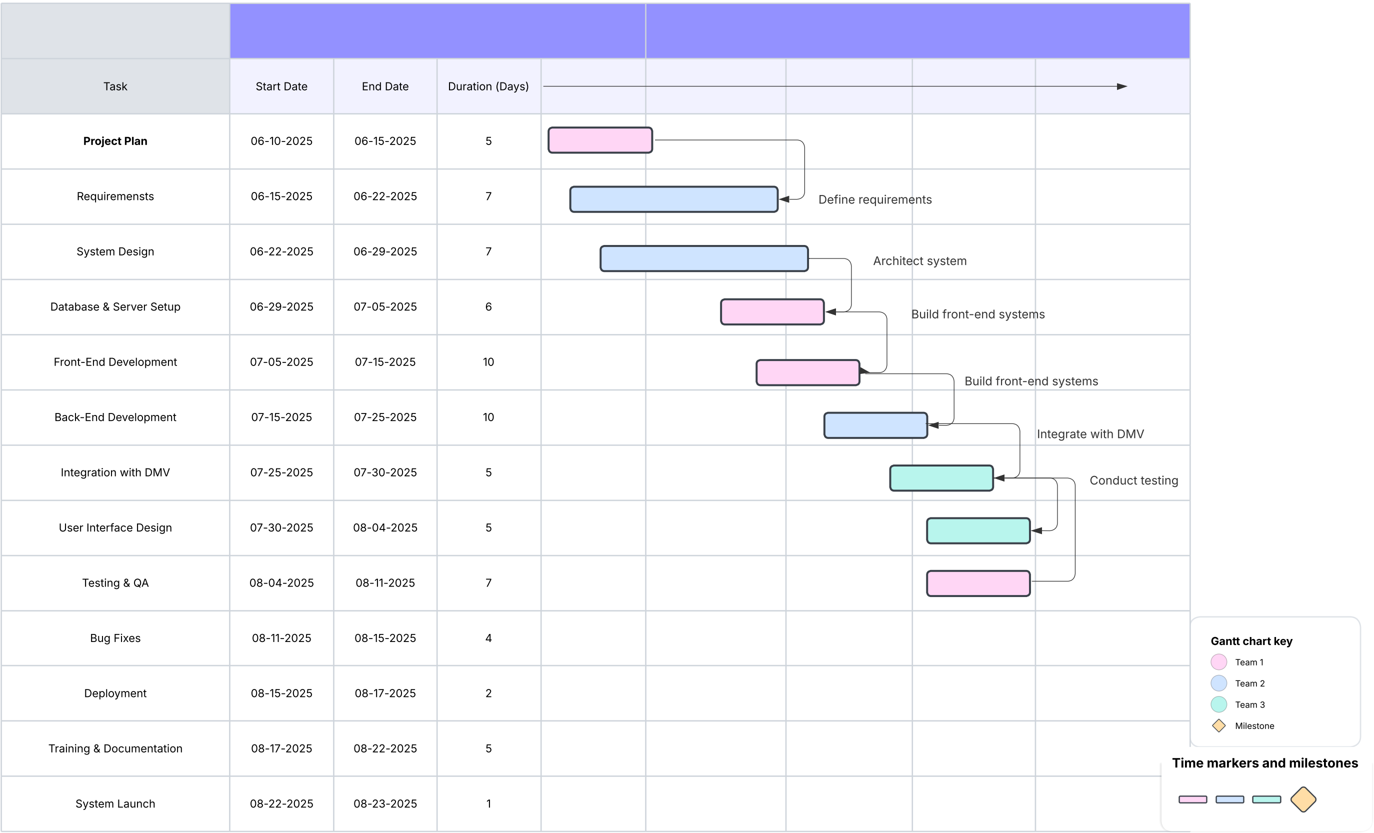
The system design has several limitations. Since the scheduling and package management features require future flexibility, the current design may not fully support dynamic package customization without developer intervention. Offline data access could be limited in functionality, potentially causing delays in syncing updates or user actions. Integration with the DMV depends on their system's availability and compatibility, which could affect timely updates. Security relies heavily on proper implementation of role-based access and encryption, but emerging threats may still pose risks.

Resource-wise, time constraints may limit thorough testing and refinement before deployment. Budget limitations could restrict the choice of advanced cloud services or third-party tools for enhanced features. Technology-wise, ensuring smooth performance across all devices and browsers can be challenging, especially with varying internet speeds or older hardware. Additionally, ongoing maintenance and support require dedicated personnel, which may strain the team if resources are tight.

Southern New Hampshire University

Gantt Chart

5-2 Project One



Task	Start Date	End Date	Duration (Days)
Project Plan	06-10-2025	06-15-2025	5
Requiremensts	06-15-2025	06-22-2025	7
System Design	06-22-2025	06-29-2025	7
Database & Server Setup	06-29-2025	07-05-2025	6
Front-End Development	07-05-2025	07-15-2025	10
Back-End Development	07-15-2025	07-25-2025	10
Integration with DMV	07-25-2025	07-30-2025	5
User Interface Design	07-30-2025	08-04-2025	5
Testing & QA	08-04-2025	08-11-2025	7
Bug Fixes	08-11-2025	08-15-2025	4
Deployment	08-15-2025	08-17-2025	2
Training & Documentation	08-17-2025	08-22-2025	5
System Launch	08-22-2025	08-23-2025	1