Southern
New Hampshire
University

CS 255 Business Requirements Document

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 and implement an online system for **DriverPass**, a company that provides driver training and testing support for students preparing for DMV exams.
- DriverPass wants a system that allows students to access online practice tests, schedule on-the-road driving lessons, and track their progress toward passing the driving test.
- The goal is to create a secure, cloud-based platform that improves efficiency for both students and staff while reducing the high fail rate among DMV applicants.

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 identified that over 65% of students fail their DMV driving tests due to poor preparation and limited training resources.

- The company aims to solve this problem by offering a combination of **online education** and **in-person driving lessons**.

- The system will allow users to register, schedule lessons, take online practice exams, and monitor performance.

- Different user roles—such as **students, instructors, secretaries, the owner, and the IT officer**—will have unique access and permissions.

- The system will also support integration with DMV updates to ensure all tests and policies remain current.

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 students with a simple way to **register, purchase packages, and schedule lessons** online.
- Enable instructors to **view assigned lessons and enter lesson notes**.
- Allow the IT officer and management to **manage accounts, track activity, and generate performance reports**.
- Ensure that all user data is securely stored and only accessible to authorized users.
- Provide real-time access to **online practice tests** that automatically calculate scores and record results.
- Maintain a flexible architecture that supports adding or disabling training packages in the future.
- Deliver a cloud-based solution that minimizes technical maintenance for DriverPass staff.

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 shall operate as a **web-based application**, accessible from computers and mobile devices.
- It shall support multiple concurrent users without significant delays.
- System updates and DMV data synchronization shall occur regularly to ensure accuracy.
- The system shall provide responses (e.g., logins, booking confirmations) within a few seconds of user input.

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 shall run in all major browsers (Chrome, Safari, Edge, Firefox) on Windows, macOS, and mobile operating systems.
- The back end shall utilize a secure database management system to store user, schedule, and exam data.
- The system shall be deployed on a cloud platform to allow 24/7 online access and reduce local 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?

- The system shall uniquely identify users by username and password to ensure data accuracy.

- The system shall display precise information about test results, lesson times, and instructor notes.

- The system shall log every user action (such as creating, modifying, or canceling a reservation) to maintain accountability.

- If a system error or data conflict occurs, the admin will be notified automatically.

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 IT officer shall be able to add, remove, or modify user accounts and permissions without programming changes.

- The owner shall be able to disable or reactivate training packages as needed.

- The system architecture shall support easy updates for new DMV requirements or software upgrades.

- The design shall allow the system to scale as DriverPass adds more cars, instructors, or locations.

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 shall be required to log in with valid credentials before accessing the system.
- All communication between users and the system shall be encrypted using HTTPS.
- The system shall lock an account after three failed login attempts to prevent brute-force attacks.
- The IT officer shall be able to manually reset passwords or disable accounts when needed.
- Users who forget their passwords shall be able to securely reset them through an automated email link.
- The system shall comply with general data privacy and security best practices to protect sensitive information.

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 allow students to register for accounts online.
- The system shall validate user credentials during login.
- The system shall allow students to purchase one of three training packages.
- The system shall allow students to schedule, modify, or cancel driving lessons online.
- The system shall automatically match each lesson with an available driver and car.
- The system shall record lesson details, including time, duration, car used, and instructor comments.

- The system shall allow students to take online practice exams and view results.
- The system shall display test name, time taken, score, and pass/fail status.
- The system shall track all user activities, including reservations and cancellations.
- The system shall allow administrators to generate downloadable reports (Excel or PDF).
- The system shall integrate DMV updates into the practice exam database and notify administrators of changes.
- The system shall provide role-based access: owner, IT officer, secretary, instructor, and student.

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: Register, purchase packages, schedule driving lessons, take practice tests, and view their results and instructor feedback.
- Instructors: View daily and weekly schedules, record lesson notes, and confirm completion of sessions.
- Secretary: Enter student information, schedule appointments by phone or in person, and manage booking requests.
- Owner (Liam): View reports, enable/disable packages, and monitor overall system performance.
- IT Officer (Ian): Manage user accounts, reset passwords, block access, and ensure system security.
- The interface shall be browser-based and mobile-friendly with clear navigation and dashboards for each role.

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?

- Students have reliable internet access and devices capable of accessing the web application.
- DMV will provide updated test data through a digital feed or file import.
- DriverPass will maintain valid licenses for all instructors and vehicles used in training.
- Online payments will be handled through a secure, third-party payment processor.
- The DriverPass team will provide ongoing feedback during system testing.

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 initial version of the system will not include self-service customization of training packages.
- Time and budget constraints may limit advanced features such as mobile push notifications or built-in chat support.
- System reliability will depend on third-party hosting and DMV update availability.
- Integration with future mobile apps or DMV APIs may require additional development in later phases.

Gantt Chart

Include a **Lucidchart Gantt chart screenshot** here showing project phases and timing.

Use the following task list and schedule from the interview:

Task	Start Date	End Date
Collect Requirements	Jan 22	Feb 4
Create Use Case Diagrams	Feb 11	Feb 18
Build Activity Diagrams	Feb 15	Mar 9
Research User Interface Designs	Feb 27	Mar 7
Build Class Diagram	Mar 1	Mar 9
Get Customer Approval	Mar 10	Mar 11
Build Interface	Mar 12	Mar 24
Link Database to Interface	Mar 24	Apr 3
Build Business Logic Layer	Apr 5	Apr 27
Test System	Apr 27	May 7
Deliver System	May 8	May 9
Sign-Off Meeting	May 9	May 10