Southern
New Hampshire
University

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 the DriverPass project is to develop a system for DriverPass, a company addressing the high failure rate (65%) of DMV driving tests by providing online practice exams, in-person lessons, and on-the-road training.
- The client, DriverPass, wants the system to manage online and office-based reservations for driving lessons, provide access to DMV-compliant practice tests, track user activities, and allow data access from any device, both online and offline (for viewing reports).
- The system aims to improve driver training and streamline business operations for DriverPass stakeholders, including customers, secretaries, admins, and IT officers.

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 seeks to fill a market gap by offering comprehensive driver training to reduce the 65% DMV test failure rate, as identified by Liam, the owner.
- The problem is the lack of effective training tools, which the system addresses through online practice tests, in-person lessons, and on-the-road training with 10 cars and drivers.
- Key components include:
    - A web-based interface for user interactions (e.g., booking, test-taking).
    - A relational database to store customers, appointments, and package data.
    - A cloud-based backend for accessibility and minimal maintenance.
    - Integration with the DMV for real-time rule and test updates.
    - Reporting tools for activity tracking and Excel-compatible data exports.

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 customers to schedule, modify, and cancel driving lessons online or via a secretary, supporting three training packages (6, 8, or 12 hours).
- It must track user actions (e.g., who made or canceled a reservation) and generate printable activity reports for accountability.
- The system should provide online practice tests with progress tracking (e.g., test name, score, status) and integrate with DMV updates for rules and questions.
- Measurable tasks include:
    - Supporting 10 cars and drivers for on-the-road training.
    - Allowing admins to disable training packages without code changes.
    - Enabling secure data access from any device, with offline report viewing.
    - Providing IT admin access for account management and password resets.

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 operate on a web-based platform, accessible via browsers (e.g., Chrome, Safari) and mobile devices (iOS, Android).
- User actions (e.g., booking a lesson) should be completed within 2 seconds for a seamless experience.
- The system should receive DMV rule updates in real-time or daily to ensure content compliance.
- The system must support concurrent access by multiple users (e.g., customers and secretaries) without performance degradation.

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 must run on cloud-based servers to minimize maintenance, supporting Windows, macOS, iOS, and Android platforms.
- A relational database (e.g., MySQL) is required to store user, appointment, package, and test data.
- The system must integrate with a DMV API or data feed for rule and test updates.
- Backup and security must be handled by the cloud provider, according to Ian’s request to focus on business operations.

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?

- Users are distinguished by roles (customer, secretary, admin, IT officer) using unique login credentials.
- Input fields (e.g., usernames, passwords) are case-sensitive to enhance security.
- The system must notify admins of issues (e.g., failed logins, system errors) via email or dashboard alerts within 5 minutes.
- Appointment data (e.g., date, time, driver) must be accurate to prevent scheduling 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?

- Admins can enable or disable training packages via a configuration interface without code changes, supporting Liam's flexibility requirement.
- The system must adapt to platform updates (e.g., browser or mobile OS updates) through regular maintenance patches.
- The IT officer (Ian) requires full access to manage accounts, reset passwords, and block users (e.g., terminated employees).
- The system should support future package customization with developer assistance, as noted by Sam.

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?

- Users must log in with a username and password, secured with encryption (e.g., HTTPS for web access).
- Data exchanges between clients and server use SSL/TLS to protect sensitive information (e.g., credit card details).
- Accounts are locked after three failed login attempts to prevent brute-force attacks, with admin notification.
- Customers can reset forgotten passwords via automated email verification with a secure link.
- Admin and IT officer accounts have elevated privileges, restricted by role-based access control.

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 allow customers to schedule, modify, and cancel driving lessons online or via a secretary, specifying date, time, and pickup/drop-off locations.
- The system shall track user actions (e.g., who made, modified, or canceled a reservation) and generate printable activity reports.
- The system shall provide online practice tests, tracking progress (e.g., test name, time taken, score, status not taken, in progress, passed, failed).
- The system shall integrate with the DMV to receive and apply updates to rules and practice questions, notifying admins of changes.
- The system should support three training packages (6, 8, or 12 hours, with varying features like online access or in-person lessons) and allow admins to disable packages.
- The system should store customer details (e.g., first name, last name, address, phone, credit card, pickup/drop-off location) and match customers with drivers and cars.
- The system shall allow admins to export reports (e.g., customer or activity data) in Excel-compatible formats for offline use.
- The system should enable IT officers to manage user accounts, reset passwords, and block access for security purposes.

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 must be web-based, accessible via browsers (e.g., Chrome, Safari) and mobile devices (iOS, Android).
- User roles include:
  - Customers: Schedule lessons, take practice tests, view progress (e.g., test scores, lesson notes), and manage account details.
  - Secretaries: Input customer data, schedule appointments, and view driver and car assignments.
  - Admins (Liam): Access activity reports, disable packages, and monitor system performance.
  - IT Officers (Ian): Manage user accounts, reset passwords, and block access.
- The interface includes:
  - A login page with secure credential validation.
  - A booking form for selecting packages, dates, times, and locations.
  - A test progress dashboard showing test name, time taken, score, and status (not taken, in progress, passed, failed).
  - A driver notes section displaying lesson times, start/end hours, and comments.
  - A contact page for customer inquiries and support.
- Interactions are via mouse/keyboard on browsers or touch on mobile devices, with responsive design for accessibility.

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?

- Users have access to modern browsers or mobile devices with reliable internet connectivity.
- The DMV provides an API or regular data feed for rule and test updates.
- The cloud provider handles backups and security, as requested by Ian.
- Customers are familiar with basic online interfaces for booking and test-taking.
- The project timeline (Jan 22–May 10, 2025) is sufficient for development and 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 system is constrained by the project timeline (ending May 10, 2025), limiting additional features like fully customizable packages without developer intervention.
- Budget constraints may restrict advanced security features (e.g., biometric authentication) or extensive testing.
- Dependency on DMV cooperation for rule updates may introduce delays if the API is unavailable.
- The system's offline capabilities are limited to report viewing in Excel, as online access is required for data updates to avoid redundancy.
- Limited resources (e.g., team size, as implied by named team members) may impact on scalability.

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.

[Insert chart]