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 this project is to fill the void in the market for training students for the driving test at the local DMV.
- The client is Liam, the owner of DriverPass, is the client.
- Liam wants a system to manage customer data, reservations, and driver assignments, accessible online and offline (for viewing), with robust tracking and reporting features and automatic updates to driver requirements.

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 aims to fix the problem of high DMV test failure rates by offering comprehensive training.
- The system will manage classes, tests, driving lessons, reservations, products/courses, and update requirements of DMV tests.
- components include a database that stores all information, an admin portal interface for managing all database information, and a consumer portal for managing all consumers' information.
- The system must be hosted remotely and must have web access, security, backups, and integration with DMV updates.

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 customer must be able to schedule, modify, and cancel all lessons booked online or through the phone.

- The secretary must be able to create, modify, and cancel all reservations for all customers.
- The system must provide telemetry on site use and statistics on completion/success rates.
- allow downloads of reports in Excel format from any device to admins
- the system must log all actions (e.g., modifications/cancelations/creations of reservations.)
- The system must have an access terminal for remote IT to manage the system.

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 as a web-based system hosted on the cloud for accessibility.
- The system must load pages and process reservations quickly (e.g., within 2-3 seconds).
- The system updates to DMV-related content (rules, tests) should occur as soon as notifications are received from the DMV.
- The system maintenance and backups should be handled automatically by the cloud provider with minimal downtime.

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 web service should be accessible from any platform that can run a web browser
- The system requires a database such as Postgres or mongoDB
- no specific OS constraints, but the system may be Unix/Linux-based for security and stability reasons.

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 unique login credentials (username/password) tied to their role (customer, secretary, IT officer, owner).
- Passwords should be case-sensitive for security; other inputs (e.g., names) can be case-insensitive for ease of use.
- The system should notify the admin (Liam or Ian) immediately if a user account is locked due to failed login attempts, if a reservation conflict occurs, or if there are any unidentified API calls.

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?

- User accounts (add/remove/modify) should be manageable via an admin interface without code changes.
- The system should use scalable cloud infrastructure to adapt to browser or OS updates automatically.
- IT admin (Ian) needs full access to all accounts, including resetting passwords and blocking access, plus system configuration rights.
- Packages can be disabled by Liam via the interface, but adding/removing them requires developer intervention in future updates.

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 log in with a username and password; customer passwords can be reset automatically via email.
- Data exchange must use HTTPS (SSL/TLS) to encrypt connections between client and server.
- Accounts should lock after 5 failed login attempts (potential brute force) and notify Ian for review.
- The cloud provider handles backups and security patches; customer credit card data must be encrypted.

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 customers to register with their personal info (name, address, phone, credit card, pickup/drop-off location).
- The system shall enable customers to schedule, modify, or cancel 2-hour driving lessons online or via the secretary.
- The system shall generate downloadable activity reports in Excel format for Liam.
- The system shall assign drivers and cars to customer reservations and track these assignments.
- The system shall log all user actions (e.g., reservations, cancellations) with timestamps and usernames for auditing.
- The system shall notify admins of DMV updates and allow manual content updates to reflect new rules.
- The system shall restrict data modifications to online-only access to prevent redundancy.
- The system shall provide access to online classes and practice tests, tracking progress (test name, time, score, status).

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.)?

- User Interface
  - The interface must be web-based and accessible via browsers on desktops, mobiles, or tablets.
  - Users and Needs:
  - Ian (IT Officer): Manage all accounts, reset passwords, block access, and configure system settings.
  - Customers: Schedule/cancel lessons, access online classes/tests, view progress (scores, status), and reset passwords.
  - Secretary: Enter customer info, manage reservations, view driver schedules.
  - Liam (Owner): Download reports, disable packages, and monitor activity logs.
  - The interface should include an input form for customer data, a progress dashboard for tests, a driver notes table (lesson time, comments), and a contact page.
  - The design should match Liam's sketch for test progress and driver notes layout.
- Potential for multi server interconnect approach for security and stability.

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?

- This assumes customers have reliable internet access and modern devices with web browsers.
- This assumes the DMV provides updates in a standardized, machine-readable format for integration.
- This assumes Liam's team (secretary, drivers) is trained to use the system with minimal technical support.
- This assumes the cloud provider handles all backups and security without additional configuration by DriverPass.

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?

- Offline data access is read-only; modifications require online connectivity, limiting flexibility in low-signal areas.
- Customizing packages (adding/removing) requires developer work, which is not feasible in real-time by Liam.
- Dependent on DMV's update frequency and format, which may delay content accuracy.
- Time constraints (delivery by May 2025) limit extensive feature additions beyond the initial scope.
- Budget not specified—assumes sufficient funding for cloud hosting and basic development.

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]