# 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?*

* DriverPass wishes to create a comprehensive training program for students preparing to obtain a driver’s license.

### 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?*

* Liam, the owner of DriverPass, believes there is a need for better driver training because “so many people fail their driving tests at the DMV.”
* DriverPass wants a platform for students to interact with various learning programs and schedule reservations for in-person driving lessons.
* The platform should include good logging practices for maintaining a record of user and administrator actions to resolve discrepancies.
* DriverPass wants a dynamic system that allows different modules to be added, removed, and disable as necessary. DriverPass is invested in creating a flexible, scalable platform to provide lessons of varying degrees.
* DriverPass offers three different packages for in-person lessons so that students can get as much or as little help as needed.
* DriverPass wants to focus on business operations, so they desire minimal technical issues.

### 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?*

* Student users (herein referred to as customers) have the following attributes:
  + The ability to manage reservations (establish, cancel, and modify appointments) for driving lessons. Customers should be able to choose a date, time, car, driver, and package for their session.
    - **Package One:** Six hours in a car with a trainer
    - **Package Two:** Eight hours in a car with a trainer and an in-person lesson where DriverPass trainers explain the DMV rules and policies
    - **Package Three:** Twelve hours in a car with a trainer, an in-person lesson where DriverPass trainers explain the DMV rules and policies – *plus* access to DriverPass’s online class with all the content and material, including practice tests.
  + Depending on their permissions (determined by the package the customer purchased), customers need to be able to access the online learning tools.
  + Enter the following information when purchasing and registering for a package:
    - First name, last name
    - Address, State
    - Phone number
    - Customer pick-up and drop-off location (should be the same location for both fields)
    - Credit card number, expiration date, and security code
  + Reset forgotten passwords
* The platform defines various roles for administrative users (employees of DriverPass)
  + Super admin (Liam) – full access over all accounts so they can be reset (forgotten passwords) or blocked (terminated employees)
  + IT Officer (Ian) – Maintains the system and modifies it as necessary. [Do not grant permissions for deleting things without super admit approval).
  + Secretary – Able to view reservations and create and change appointments.
* Platform functionality
  + Packages should have modularity so they can be disabled by the super admin or IT officer
  + The platform operates via the web, preferably through the cloud, so that users can access their data anywhere they have a connection with any computer or mobile device.
  + The system permits offline work by making various reports downloadable
  + Customer appointments for reservations have a CRM log that shows activity on the reservation.
  + Sends a notification when the DMV has updated policies, regulations, rules, etc.
  + Offers a “contact us” page for students to contact a DriverPass representative
  + It has an interface that shows student information, student learning progress, and driver notes from training sessions
  + Offers a page with an input form for students (or the secretary) to fill in student information, including a way for DriverPass to contact the student

## 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 DriverPass system should operate as a web-based application.
* The DriverPass system only needs to be fast enough to function comfortably for users, but extreme optimization is unnecessary. Therefore, please do not spend more on an asset than it is worth.
* The system should be updated on an as-needed basis determined by Liam, the DriverPass owner

#### 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 run on standard browsers such as Mozilla Firefox, Google Chrome, and Microsoft Edge. Ideal additional platforms include the Opera web browser and the mobile variants of all the previously mentioned browsers.
* The system’s backend will need a database to support various application functions, such as maintaining student information, learning modules, and scheduling.

#### 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 will be distinguished using a role-based access control matrix (RBAC). User classifications include admin (Liam), IT officer (Ian), secretary, drivers, and students.
* Input should be case-sensitive only for passwords. Case sensitivity in any other element is inconsequential.
* The system should inform the admin of a problem if scheduling conflicts, database errors, or mismatching drop-off and pick-up locations occur.

#### 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 ability to perform user changes should be built into the backend interface. Liam does not wish to handle the technical aspects of the program.
* The system should use up-to-date non-experimental features of the programming language chosen (most likely JavaScript, CSS, and HTML) to allow for maximum adaptability in the face of platform updates. Major updates that do not allow backward compatibility should notify the administrator and IT officer immediately.
* The IT admin needs permission to update and modify the system. However, the IT admin should not be able to delete anything without admin approval.

#### 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 requirements for a user to log in should include a username, password, and multi-factor authentication.
* Data exchanges should be secured through authentication, encryption, and SSL-enabled file transfer protocols (FTPS, HTTPS).
* A limit on failed logins should be implemented to reduce the threat of a brute force hack. After a predetermined number of failed attempts, the administrator should be notified.
* A standard “forgot your password” method should be implemented for end users to reset their passwords automatically. It should include sending a reset link to the user’s associated email that expires after a preset amount of time.

### 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 generate reports that can be exported to Microsoft Excel
* The system shall implement role-based permissions for different users (customer, admin, IT officer, secretary, driver)
* The system shall permit users to make reservations
* The system shall accept debit and credit transactions
* The system shall provide modular packages that can be enabled or disabled as desired
* The system shall require complete information to be entered for customers
* The system shall operate within a web browser
* The system shall maintain a database for users’ information
* The system shall include a notes section for drivers to record information about students’ lessons
* The system shall offer contact information for users to contact DriverPass representatives
* The system shall link with the DMV to provide up-to-date and accurate information

### 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.)?*

* Required sections of the interface as specified by Liam’s template:
  + Logo (top of page)
  + Online test progress
  + User information
  + Driver notes
  + Special needs
  + Driver photo
  + Student photo
* The interface should have a navigation bar for ease of access and a more straightforward design
* The interface should include a section for account settings, such as changing passwords
* There will be various users for the interface: students, drivers, and admins. The needs of each user are below.
  + Students
    - Read driver notes
    - Update personal information
    - Upload personal photo
    - Access learning modules
    - Request appointments
  + Drivers
    - Search individual students
    - Create and modify notes for students’ practice sessions
    - Upload driver photo
  + Admins
    - Search individual students
    - Modify user account status
    - Reset users’ passwords
    - Create, modify, and delete appointments for in-person training sessions
    - Generate and download reports
* Users will interact with the interface via a web browser. Therefore, mobile web browser functionality would be optimal but is not required.

### 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 a computer
* Students have a debit or credit card
* Users are familiar with technology
* Liam or Ian is comfortable with backend technologies such as Microsoft Azure or Amazon Web Services

### 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 cannot easily and reasonably be converted into a mobile application
* A small team limits the timeline for the project

### 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.*

Chart, timeline

Description automatically generated