# 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?*

* Client Purpose – To take advantage of a void in the market when it comes to training students for the driving test at their local department of motor vehicles.
* Client Goal – Wants their students to be able to take online classes, practice tests, and provide them with on-the-road training if they wish.

### 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?*

* System Purpose – Online classes, practice tests, and scheduling on-the-road training.
* Problem to Fix – There is a need for better driver training. Many people fail their driving tests at the DMV.
* Components:
  + Cloud Servers, server backups and security should be provided/automated
    - System will be web based
      * The interface for user webpage, when logged in, should show:
        + Online Test Progress
        + Personal Information
        + Driver Notes
        + Special Needs
        + Driver Photo
        + Student Photo
      * Other pages include:
        + Registration Form, for students or the secretary to fill out
        + Contact Us
      * A way to contact the student, possibly a messaging system
  + Access system data from anywhere
    - SQL servers
    - At least two databases, user data and reservation data
  + Export data from the system for reports and analysis
    - SQL servers can support exporting to Excel
  + Role management for different users
    - Boss role, full access
    - IT, system access, no access to user information
    - Secretary, access to users and reservation system only
    - Students, personal data and reservation requests
  + Server logs that record events for security reasons, must be able to export the logs
  + A scheduling system
    - Real-time updates to track online reservations, cancelations, and modifications
    - The secretary can submit reservations on behalf of users over the phone or in person.
    - A separate log for the scheduling system, must be able to export the logs
    - Varying number selectable packages when making appointments
  + User accounts with sensitive identifying information
    - Registration begins online, or with a phone call, and then the customer gives us their information. This information would include their first name, last name, address, phone number, state, and their credit card number, expiration date, security code, drop-off and pickup location.
    - Self-help password reset
  + Server should be connected to the DMV so that they can update new rules, policies, or sample questions. A notification should be sent whenever they have an update.

### 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?*

* Completed Goal – The system, when completed, should allow students to create accounts on a website, over the phone, or in person to access online classes, practice tests, and schedule on-the-road training. There are 10 cars/drivers and three packages to pick from and assign to a student when they schedule on-the-road training. The main page, or dashboard, for the students should show the progress of their online tests and on-the-road training as well as their personal information and notes the drivers leave for them. There should be a way for the students to contact and be contacted by the DriverPass team. There should be access to the online classes and practice tests on the main page as well.  
  The owner also wants to be able to export server logs and data to Excel for reports as well as be connected to the DMV for new rules, policies, or sample questions.

DriverPass also wants to system to be web based and on the cloud for access to have server backups and security taken care of for them.

* Measurable Tasks:
  + User Account Management
    - Allow users to create accounts through the website, phone, or in-person
    - Store and encrypt personal data (GDPR/CCPA compliance)
    - Applicable security features such as, multi-factor authentication (MFA), and login failure limits
  + Online Learning/Testing Portal
    - Access to online classes & practice tests on the users’ main page, dashboard
    - Sync with DMV for updated rules/sample questions (API integration)
    - Track progress of online tests
    - Track data for course completion rates, and test passes and failures
  + On-the-Road Training Scheduling
    - Real-time updates to prevent double booking
    - Track data for scheduling/rescheduling as well as cancelations of drivers
    - Allow students to select drivers and packages
  + Student Dashboard
    - Show personal test progress
    - Allow students to edit their personal information
    - Show driver notes
  + Communication System
    - Email/Phone notifications
    - Messaging system between the students and DriverPass team, no student-to-student communication to help prevent phishing
  + Data Export
    - The owner can export logs and data to Excel
    - SQL servers can generate analytic information
  + Connection to DMV (API integration)
    - Auto sync rules, policies, and test/sample questions
    - Notify the DriverPass team when it updates
  + Cloud Hosting with Security
    - Web-based cloud hosting server
    - Automated backups at predetermined times
    - Security frameworks to prevent downtime and protect user data

## 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 should be hosted on a cloud platform and run only on web browsers from either a computer or any device capable of running a web browser (e.g., tablets, phones). Any data downloaded can be viewed offline, but it cannot be reuploaded.
* The system needs to be scalable to handle at least 50 or more users and run in real-time to provide immediate notifications and prevent conflicts in the reservation system. It needs to be that fast for user actions as well.
* The system needs to update as fast as the DMV API would update their changes.
* The system should be routinely updated for bugs and security, bi-weekly or monthly.
* The system should always be able to manually update in case of emergencies (e.g., security flaws).

#### 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 hosted on a cloud platform and run only on web browsers from either a computer or any device capable of running a web browser (e.g., tablets, phones).
* The back end should have a database with audit logs, package configuration (Liam should be able to enable/disable packages themselves), and user roles.
* Cloud services will manage the scaling, backups, and security for the client

#### 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?*

* Role-Based Access Control (RBAC), all users will have a primary ID linked to an email address (unique). The system will assign the ID in the database and give them their roles, usually a customer or driver. Other roles should be manually created, such as secretary, IT, or owner.
* User input should be case-sensitive for at least passwords. Emails, names, and addresses can be either.
* The system should notify the admins of a problem, via email/SMS when there are:
  + Security issues (e.g., brute-force attacks or unauthorized access attempts, admin password resets)
  + System failures (e.g., payment issues or API connection failures)
  + Reservation conflicts and packages forcefully enabled/disabled
  + Database connection fails

#### 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?*

* IT and owners can add/remove/modify users and disable packages.
* Platform, cloud servers, updates are automated and unnoticeable, and users handle their own browser updates.
* The IT role needs to be able to manage users, toggle packages, update driver and car information, view audit logs, and have access to other troubleshooting methods.

#### 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?*

* A user needs an email, a password, and possibly multi-factor authentication from a phone or app
* Data in transit and storage should be encrypted. APIs should limit requests made by each IP address.
* In case of brute force attacks accounts should have a limited number of attempts within a set timeframe. Accounts can be locked temporarily, permanently, or IP addresses can be blocked. Admins should get notified of accounts potentially getting brute forced or locked.
* If a user forgets their password, they should be able to send a time limited reset request to their email. After a password reset all sessions are invalidated and the user must re-login to their devices.

### 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 use Role-Based Access Control (RBAC)
* The system shall use five roles, Owner, IT, Secretary, Drivers, Customers
* The system shall track activities of users and allow downloadable reports to view offline
* The system shall use some form of appointment management
* The system shall let customers make appointments online or in person or over the phone with a secretary
* The system shall assign drivers and cars
* The system shall have three tiers of purchasable packages, with the ability to add more or disable them
* The system shall utilize user accounts to record names, addresses, phone numbers, states, payment details, pickup/drop-off locations, self-service portal for appointments
* The system shall have driver comments to review, and online classes and practice tests with the correct package
* The system shall use a DMV API to stay up to dates with new rules, policies, and practice questions
* The system shall utilize a cloud base web application
* The system shall let the cloud provider handle automated backups and security
* The system shall have a clean user interface with relevant information and easy navigation
* The system shall scalable to adjust to few or many users

### 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.)?*

* Needs:
  + A browser to access the site/user interface
  + Role-based access control (RBAC) and encryption to keep user and site data secure
  + Respond in real time over computer, tablet, or mobile devices
  + Option to download user logs and data
* All users need a login portal to access their unique user interface based on their role
* Users:
  + Customers:
    - Needs:
      * Set up driving lessons and pickup/drop-off destinations
      * Access online courses and practice tests, if enabled
      * View their test scores and lesson comments
      * Self-service portal for account issues
    - Interaction:
      * They will interact with a browser
      * They will interact with a dashboard that shows test progress, upcoming lessons, lesson comments, and driver and car details
  + Secretaries:
    - Needs:
      * Ability to manage appointments for customers
      * Ability to create new users on behalf of a customer
      * Ability to view driver schedules and available cars
      * Ability to contact customers
    - Interaction:
      * A format to view schedules (e.g., something calendar-esque like a weekly view)
      * Search for users in the database
      * Schedule and create accounts for customers
  + Drivers:
    - Needs:
      * Ability to view their own lesson schedules
      * Ability to submit comments/feedback on lessons
      * Ability to contact their students or secretary
      * Ability to report issues
    - Interaction:
      * A calendar format similar to secretary but less information
      * Will submit lesson notes
      * Will be notified of schedule changes
      * Will view their schedule
  + IT:
    - Needs:
      * Complete access to accounts
      * Ability to manage permissions and roles of users
      * Ability to conduct maintenance and updates
      * Ability to monitor APIs for updates
    - Interaction:
      * An administrator dashboard with system tools
      * Audit log viewing
      * Manage users and roles
  + Owner:
    - Needs:
      * Generate reports
      * Disable packages
    - Interaction:
      * A dashboard that can view the cloud sever metrics (e.g., financials, customer numbers)
      * Package activation and deactivation
      * Export data into reports

### 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?*

* Under the assumption that everyone has a smart phone or access to a device able to use a browser
* If the owner and IT lose access to their accounts how will they retrieve them?

### 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?*

* This has already been addressed but, packages can be disabled and enabled but not added currently
* Unless DMV API and the system is automated to update there is a chance for outdated information
* It would be hard for the Secretary and IT roles to carry their tasks out on a mobile device

### 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.*

