# 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: **DriverPass** (Owner: Liam; IT Officer: Ian).

 Purpose: Build a **cloud-hosted web system** that supports: online classes & practice tests, on-road lesson reservations, progress tracking, driver notes, activity/audit reports, and role-based administration.

 Access: From **any computer or mobile device;** allow **report export (e.g., Excel)** for offline review.

### 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?*

 Problem: High DMV test failure rates; DriverPass needs a platform for training, scheduling, and tracking.

 Components:

* Web app (student/secretary/admin/IT interfaces)
* Authentication & **role-based access control**
* Reservations module (lessons = 2-hour blocks; car/driver matching)
* Online class & practice tests (with scoring and status)
* Driver notes & lesson history
* **DMV update integration** (receive rule/policy/question updates + notifications)
* Reporting & **activity/audit logs**
* Payments (capture card details - use secure processor/tokenization)
* Database (customers, users/roles, cars, drivers, lessons, packages, tests, results, activity log)

### 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?*

 Enable customers to **register**, reset passwords, and **schedule/modify/cancel** lessons online or via secretary.

 **Match** each lesson to a car and driver without conflicts.

 Show online test **progress** (name, time, score, status: not taken/in progress/failed/passed).

 Maintain complete **audit trail** (who created/changed/canceled reservations).

 Provide **printable reports** and exports (CSV/XLSX).

 Allow admin to **enable/disable packages** (1: 6 hrs, 2: 8 hrs + in-person rules, 3: 12 hrs + in-person + online class/practice tests).

 Receive and surface **DMV updates** with notifications.

 Deliver within the **schedule** below.

## 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?*

 **Environment:** Cloud-hosted **web application** (responsive desktop & mobile browsers).

 **Responsiveness:** Pages load ≤ **2 s** under normal load; reservation creation ≤ **3 s** including conflict checks.

 **Uptime:** 99.5% monthly target.

 **Updates:** App releases at least **monthly; DMV updates** ingested and published **within 24 hours** of receipt.

#### 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?*

 **Browsers:** Latest Chrome/Edge/Firefox/Safari.

 **Server:** Linux container or PaaS;

 **Database:** Relational (e.g., PostgreSQL/MySQL) for scheduling constraints & audit logs.

 **Payments:** Integrate with a **PCI-compliant processor**.

 **Cloud storage/backup** handled by provider as requested.

#### 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?*

 Unique user IDs (email) and **case-insensitive** login; strong input validation.

 Reservation validator: prevents **double-booking** of **driver, car, student**, and **time**; enforces **2-hour** session length.

 Pickup & drop-off must match (per requirement); flag mismatch.

 **Admin alerts** on failed login thresholds, repeated payment failures, or schedule 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?*

 **Admin UI** to add/remove/modify users and assign roles **without code changes**.

 **Packages** managed via configuration (enable/disable; future modules pluggable).

 Compatible with cloud platform updates; infrastructure as code for repeatable environments.

 IT admin (Ian) has **full account control**: reset passwords, block ex-employees.

#### 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?*

 Auth: Email + password with **rate-limited login** and **account lockout** after N failures; **self-service password reset** via email.

 **TLS** for all traffic; **hashed & salted** passwords; **RBAC** for roles (Owner/Admin/IT/Secretary/Driver/Student).

 **Audit logging** of create/update/cancel actions (who/when/what).

 Sensitive data: use **payment processor tokens**; redact PAN in logs; follow **least-privilege** access.

### 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 users to **create accounts**, authenticate, and **reset passwords**.

 The system shall let **students** schedule, modify, and cancel **2-hour** driving lessons online.

 The system shall let **secretaries** create/modify/cancel lessons by phone or walk-in.

 The system shall **assign** a lesson to a **driver** and **car**, enforcing **no time conflicts**.

 The system shall **record** who created/modified/canceled each reservation and expose **activity reports.**

 The system shall manage **training packages** and allow admins to **enable/disable** packages.

 The system shall provide **online classes** and **practice tests**, storing **scores, time taken**, and **status**.

 The system shall display a **test progress** table (test name, time, score, status).

 The system shall store and display **driver notes** and **lesson times** for each student.

 The system shall **ingest DMV updates** and **notify** admins/users of new rules/policies/questions.

 The system shall export **reports** (CSV/XLSX) for offline analysis.

 The system shall allow **IT/admin** users to **reset accounts** and **block access** for terminated staff.

### 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.)?*

 **Users:** Owner (reports/overview), IT/Admin (user/role mgmt, resets), Secretary (scheduling), Driver (view schedule, add notes), Student (profile, schedule, classes/tests, progress).

 **Views/Pages:**

* Login/Reset Password; Profile
* **Scheduling** (calendar/list with conflict checks)
* **Student info input form** (name, address, phone, state, pickup/drop-off)
* **Packages** (select, enable/disable for admin)
* **Online class & practice tests** (take tests, see results)
* **Driver notes** table
* **Reports & audit logs**
* **DMV updates** notification center

 **Access:** Responsive **browser** on desktop & mobile.

### 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?*

 Year for dates is current project year (2025).

 Payments are processed by a **third-party gateway**; no storage of raw card data.

 DMV provides updates via feed/email/portal accessible by the system.

 Drivers = 10; each car has a dedicated driver (per interview).

 Scope excludes a full **content authoring system** - initial course/test content is provided 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?*

 Future **package customization by non-developers** is out of scope (can enable/disable only).

 Budget/time constrain advanced features (e.g., mobile app - using responsive web instead).

 Integration details with DMV may limit automation (depends on DMV interface).

 Small team availability affects parallelism; dependencies may shift dates if approvals slip.

### 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.*

*A graph with green squares

AI-generated content may be incorrect.*