# CS 255 Business Requirements Document Template

## System Components and Design

### Purpose

The purpose of this project is to develop a full solution for DriverPass, a company that specializes in helping people prepare for and pass their driving tests. DriverPass wants a system that allows customers to access online study tools, schedule on-the-road training sessions, and keep up with the latest driving regulations—all from their computers or mobile devices. This new solution will make it easier for DriverPass staff to manage and track appointments, handle customer data, and update driving packages as needed.

### System Background

The background of the proposed system is provided to explain why DriverPass needs a more efficient way to serve customers who are struggling with driving test preparation and scheduling. Currently, too many learners fail their tests due to limited access to practice exams and on-the-road training. By creating a cloud-based system, DriverPass aims to offer interactive practice tests, digital lesson materials, and real-time DMV rule updates. On top of that, the system should give users the ability to schedule, cancel, or reschedule lessons whenever they need, and it should let employees monitor and manage reservations through a secure interface.

In addition, DriverPass offers multiple driving packages ranging from basic single-lesson plans to more comprehensive bundles and the system must reflect these package options accurately. A tiered client level access structure will be essential, ensuring that administrators, driving instructors, and front-desk staff each have the appropriate permissions for their responsibilities. This role-based approach will protect sensitive data while keeping the system user-friendly for both employees and customers.

### Objectives and Goals

## Goals

## Improve Driving Test Success Rates

## DriverPass wants to boost the number of learners who pass their driving tests by offering enhanced study and practice resources.

## Provide Convenient and Flexible Scheduling

## Customers should be able to make appointments without calling in or visiting a physical location.

## Ensure Secure and Efficient Data Management

## The system must handle customer information safely, prevent unauthorized access, and integrate with DMV updates in real time.

## Objectives

## To achieve these goals, the system design will include the following measurable steps:

## Implement a Cloud-Based Reservation System

## Allow customers to schedule, modify, or cancel lessons online.

## Automatically notify instructors about changes.

## Enable Secure Role-Based Access

## Create tiered levels for different employee roles (administrators, driving instructors, etc.).

## Limit sensitive actions (e.g., viewing customer payment details) to authorized roles.

## Incorporate Real-Time DMV Updates

## Synchronize with DMV APIs to present new rules or testing requirements.

## Automatically notify customers of significant changes.

## Automate Key Customer Service Functions

## Let customers reset forgotten passwords securely.

## Offer built-in reporting tools for staff to track lesson usage and system activity.

## Provide Scalable, Flexible Features

## Allow for future expansions, such as additional packages or service tiers.

## Keep the system adaptable so it can grow with DriverPass’s evolving needs.

## Requirements

### Nonfunctional Requirements

**Performance Requirements**

* + Environment: The platform must be web-based and mobile friendly, allowing access on various devices.
  + Speed: Users should encounter minor loading delays, with a recommended max of 3 seconds for page loads on a standard broadband connection.
  + Updates: DriverPass must be able to push system updates without extended downtime, ideally using rolling releases or scheduled maintenance windows.

**Platform Constraints:**

* + Operating Systems: The application should function on modern browsers (Chrome, Firefox, Edge, Safari) across Windows, macOS, Linux, iOS, and Android.
  + Back End Requirements: A secure cloud-based server environment is required. The system must integrate with a relational database (e.g., MySQL or PostgreSQL) or a cloud-hosted database solution.

**Accuracy and Precision:**

* + User Identification: Each user (customer, staff, or admin) must have a unique username or ID. Case sensitivity may apply for passwords but not for usernames.
  + Admin Alerts: The system must notify the IT officer if data anomalies occur (e.g., duplicate reservations, suspicious login attempts).

**Adaptability:**

* + Modifying User Roles: IT admins should be able to grant or revoke roles without changing the system code.
  + Platform Updates: The system should gracefully handle updates from the cloud hosting provider or any software frameworks used in the application.

**Security:**

* + Authentication: Users must log in with unique credentials. Passwords should be encrypted at rest and in transit.
  + Connection Security: All communication between client and server should use HTTPS/SSL.
  + Brute Force Protection: After a set number of failed login attempts (e.g., five), the account is locked, and an admin is notified.
  + Password Recovery: Users should be able to reset their passwords through a secure, token-based mechanism.

**Functional Requirements:**

* Each requirement starts with “The system shall . . .” to define critical functions:
* User and Reservation Management:
* The system shall allow users to create, update, or delete driving lesson reservations.
* The system shall validate payment details (card number, expiration date, security code) before finalizing reservations.
* The system shall track which driver and car are assigned to each reservation.

**Package Management:**

* The system shall enable or disable available driving packages based on admin input.
* The system shall allow adjustments to package pricing or features without redeployment.

**DMV Integration:**

* The system shall connect to the DMV’s API to retrieve updated rules or testing guidelines.
* The system shall notify customers if new rules affect their upcoming reservations or study materials.

**User Roles and Security:**

* The system shall enforce role-based access to limit administrative actions to authorized staff.
* The system shall automatically lock an account after five failed login attempts.

**Logging and Reporting:**

* The system shall maintain an activity log capturing who created, modified, or canceled a reservation.
* The system shall generate reports for administrators, such as daily reservation summaries and system usage metrics.

**User Interface:**

* Stakeholders:
  + Customers: Need a clear interface to book lessons, take practice tests, and track progress.
  + Secretary/Front Desk Staff: Must see a calendar view of scheduled appointments, quickly edit reservations, and handle phone requests.
  + Owner/IT Officer: Should access administrative tools for enabling/disabling packages, checking logs, and handling user accounts.

**Interface Design:**

* + Should be intuitive and responsive on both desktop and mobile.
  + Include forms for capturing user data (first name, last name, address, credit card details).
  + Offer visual dashboards (e.g., a calendar for lesson scheduling, progress bars for online tests).

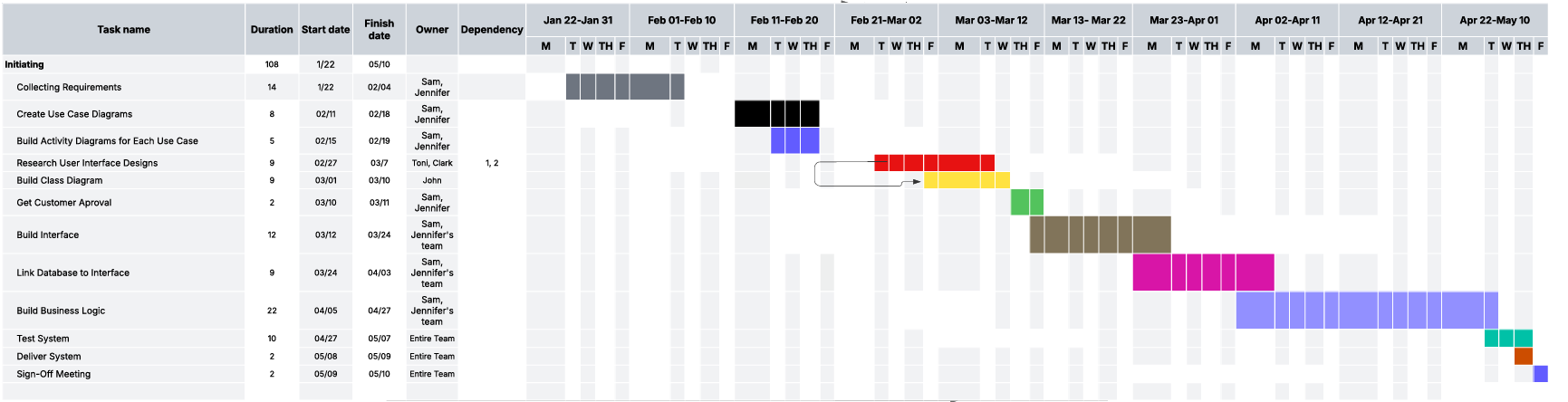
**Assumptions:**

* Users have a stable internet connection to use online functionality and real-time updates.
* DriverPass will provide valid credentials for connecting to the DMV’s system.
* Users have at least a basic smartphone or computer device that can run a modern web browser.
* Credit card payments will be handled by a secure third-party service (e.g., Stripe, PayPal) to ensure compliance with payment industry standards.

**Limitations:**

* Offline Usage: While some data can be downloaded, true read/write functionality requires an internet connection.
* Budget Constraints: Advanced features like voice-activated assistance may be deferred until future development cycles.
* Time Constraints: The system must be delivered in the timeline specified in the Gantt chart, which may limit the scope of additional, nonessential features.
* Maintenance: Complex modifications to the system (e.g., adding entirely new packages with unique rules) may require developer intervention.

### Gantt Chart

**