# CS 255 Business Requirements Document Template

## System Components and Design

### Purpose

* This project aims to create a robust system tailored for DriverPass, a driver training company, with the goal of remedying the shortage of efficient tools for students to ace their driving exams. The system will offer online practice exams and practical on-road instruction to enhance students' readiness for obtaining their driver's licenses.

### System Background

* DriverPass endeavors to enhance student success rates in passing their driving tests through online classes, practice exams, and firsthand road training. The system will enable users to access their data seamlessly, both online and offline, while facilitating reservation management, activity tracking, and adherence to the Department of Motor Vehicles (DMV) regulations. Additionally, it will cater to diverse user roles with customizable access levels, all within a user-friendly interface.

### Objectives and Goals

* Furnish virtual classes and simulated tests to aid in driving license exam readiness.
* Conduct practical road sessions led by seasoned drivers.
* Empower users to schedule, adjust, and cancel driving lesson bookings.
* Monitor user engagement and furnish comprehensive reports for oversight and accountability.
* Remain current with DMV regulations, policies, and sample inquiries.
* Uphold data integrity and implement diverse user roles with pertinent access privileges.
* Develop an intuitive, aesthetically pleasing interface for seamless user navigation and engagement.

## Requirements

### Nonfunctional Requirements

#### *Performance Requirements*

* **Swift Responsiveness**: The system is expected to promptly respond to user interactions, ensuring a seamless user experience.
* **Robust Scalability**: The system must efficiently accommodate numerous concurrent users without experiencing notable performance declines.
* **Timely Report Generation**: The system should generate reports within predetermined timeframes to meet operational needs and user expectations.

#### *Platform Constraints*

• Ensure accessibility via web browsers and mobile devices.  
• Guarantee compatibility with prevalent operating systems, including Windows, macOS, iOS, and Android.

#### *Accuracy and Precision*

• Accurately capture and present user data, encompassing reservations, progress, and test outcomes.  
• Ensure precise and consistent numeric calculations, particularly concerning test scores.

#### *Adaptability*

• Facilitate seamless adaptation to forthcoming alterations in DMV regulations, policies, and sample questions.  
• Support the inclusion or removal of driving lesson packages with minimal technical complexity.

#### *Security*

* Safeguard user data, encompassing personal and financial details, through secure storage and transmission protocols.
* Assign suitable access privileges to various user roles, enabling account management and the capability to prevent unauthorized access.

### Functional Requirements

### *User Interface*

* Ensure a seamless and user-friendly interface to facilitate effortless navigation.
* Display comprehensive user progress, encompassing completed and ongoing tests, featuring pertinent details like test title, duration, score, and status.
* Enable online scheduling, modification, and cancellation of driving lesson reservations.
* Provide users with a channel to communicate with DriverPass through the system, ensuring prompt responses.

### *Assumptions*

* Assume users possess fundamental computer skills and internet accessibility.
* Expect users to furnish precise and authentic information during registration and reservation procedures.
* Employ cloud-based technologies in constructing the system to ensure scalability, dependability, and data redundancy.

### *Limitations*

* Non-developer-friendly modules for adding or removing functionality will not be incorporated into the system.
* Any future feature expansions will be evaluated for separate releases and will not be part of the initial system design.

**[Schedule]**

### Gantt Chart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Gantt Chart** | **January** | **February** | **March** | **April** | **May** |
| **Collect Requirements** | Jan 22 - Feb 4 Sam & Jennifer | |  |  |  |
| **Create Use Case Diagrams** |  | Feb 11 - Feb 18 Sam |  |  |  |
| **Build Activity Diagrams for Each Use Case** |  | Feb 15 - Mar 9 Sam | |  |  |
| **Research User Interface Designs** |  | Feb 27 - Mar 7 Toni & Clark |  |  |  |
| **Build Class Diagram** |  |  | Mar 1 - Mar 9 John |  |  |
| **Get Customer Approval** |  |  | Mar 10 - Mar 11 Sam & Jennifer |  |  |
| **Build Interface** |  |  | Mar 12 - Mar 24 Sam & Jennifer |  |  |
| **Link DB to Interface** |  |  | March 24 - April 3 Sam & Jennifer | |  |
| **Build Business Logic** |  |  |  | Apr 5 - Apr 27  Sam & Jennifer |  |
| **Test System** |  |  |  | Apr 27 - May 7  Sam & Jennifer | |
| **Deliver System** |  |  |  |  | May 8- May 9  Sam & Jennifer |
| **Sign-off Meeting** |  |  |  |  | May 9- May 10  Sam & Jennifer |