# **CS 255 Business Requirements Document Template**

## **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?*

* The client is DriverPass, represented by the owner Liam, and the IT officer Ian.
* Provide online driver training services including classes, practice tests, and appointment scheduling/management for driving lessons
* Allow customers to schedule, modify, and cancel driving lesson appointments
* Offer 3 different packaged deals for driving lessons
* Track student progress and link students to specific instructors, cars, and lesson times
* Collect customer information like name, contact, address, payment
* Allow admins to reset passwords, block users, and generate reports
* Get notifications from the DMV about rule or test changes
* Be cloud-based and accessible online from any device

### **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?*

* Provide online driver training services (classes, practice tests, appointment-based on-the-road training, scheduling/management)
* Allow customers to schedule, modify, and cancel driving lessons online
* Offer different packaged driving lesson deals
* Track student progress and link to instructors, cars, times
* Collect customer information and payment
* Allow admins to reset passwords, block users, generate reports for offline use
* Get notifications from the DMV about rule/test changes
* Be accessible online from any device

The main problems DriverPass wants to fix are:

* Making driver training services more accessible and convenient through online scheduling and classes
* Integrating with DMV to stay up-to-date on rule changes

The key components needed for the system are:

* User accounts and roles (users, admins, instructors/employees)
* Appointment scheduling and management
* Student progress tracking
* Course content and online classes
* Practice tests
* Payment processing
* Admin features like generating reports, password reset, etc.
* Notifications from the DMV system
* Responsive online interface accessible from all devices

### **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?*

* Allow customers to create accounts, log in, take practice exams/classes, and schedule driving lessons online.
* Offer different packaged deals for driving lessons. The system needs to be flexible enough to allow customization of these packages in the future.
* Link students to specific instructors, cars, and lesson times for each appointed driving lesson of class.
* Allow students to track their progress and view lesson details like instructor feedback and grades. Progress tracking and driving logs need to be built in.
* Collect customer information (name, address, etc.) and payment during the booking process. Payment processing and data collection forms are required.
* Give admins the ability to reset passwords and block a user’s access.
* Get notifications when the DMV provides updates to rules or tests.
* Provide a responsive interface accessible from all devices. Mobile-friendly and responsive design is needed.

## **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 client’s system needs to run over the cloud and be accessible online from any device meaning a web-based application would be best suited.
* The system needs to have fast performance when customers are booking appointments or taking practice tests. Slow load times or delays in booking will negatively impact user experience.
* The system should be updated regularly based on DMV notification of rule/policy changes
* Appointment scheduling, changes, and cancellations by customers should be reflected in real-time across the system

#### **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?*

* As a web-based cloud system, the front end should run on common web browsers like Chrome, Firefox, Safari, etc. on platforms like Windows, MacOS, iOS, and Android.
* A relational database is required to store user accounts, appointments, courses, test content, grades, instructors, students, etc.
* An integration with a payment processor is required for collecting customer payments.
* Services for sending emails and texts would be needed for notifications to instructors and students.
* A cloud hosting provider should be used to deploy and host the web application.

#### **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?*

* The system should have different user accounts and roles with different access rights
* Users would log in with unique credentials to identify themselves. These should be case-sensitive to improve security.
* Any errors in the normal flow of operation like payment processing, server error, or decrease in performance should send alerts to admin or admins.
* This could be done through email/SMS or through a dashboard only accessible to the admin role.

#### **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?*

* An admin dashboard could allow for adding, editing, and deleting user accounts without code changes.
* The web interfaces should follow responsive design principles to adapt across platforms and updates to browsers or operating systems.
* The backend should be platform-independent by using cloud hosting
* IT admin would need access to servers and databases for maintenance and if the IT admin is in charge of the user’s technical issues, they would need access to the user account to be able to assist.

#### **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?*

* Users need registered accounts with a unique username and password.
* Allow users to reset passwords by emailing a time-limited token or link
* Use HTTPS for client-server communication.
* Validate all user input on the server to prevent attacks.
* Use password hashing to avoid storing passwords in plain text
* Lock accounts after a maximum number of incorrect password attempts
* Could implement captcha challenges after a certain number of failed attempts.

### **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 customers to create new user accounts with unique usernames and passwords.
* The system shall validate user credentials when customers are logging in.
* The system shall allow customers to select and purchase driving lesson packages.
* The system shall allow customers to schedule new driving lessons by selecting a date, time, and instructor.
* The system shall allow customers to modify or cancel scheduled driving lessons.
* The system shall update driving lesson availability in real-time as appointments are made/changed.
* The system shall send appointment reminders and notifications to customers.
* The system shall track customer driving lesson history, grades, and progress.
* The system shall provide driving instructors tools to log feedback and details after lessons.
* The system shall allow administrators to add, modify, and deactivate user accounts.
* The system shall generate reports for administrators.
* The system shall integrate with DMV systems to receive rule and test updates.
* The system shall be built with a responsive design to optimize the experience across devices and maximize compatibility.

### **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.)?*

* Customers - Schedule, manage, and pay for driving lessons, take and track test progress, and view instructors' notes and grades
* Instructors - Log feedback and grades after giving lessons
* Admins - Manage system, users, and review reports/alerts
* The interface should be accessible via web browsers on desktop and 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?*

* Localization - Support for other languages, etc.
* Accessibility - No specific accessibility features for users with disabilities.
* User Interface - Did not specify how the admin dashboard or alerts would change the provided UI layout.
* This assumes that potential users would need access to a modern web browser either on a desktop or mobile device.

### **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 integration with the DMV system relies on the DMV providing accessible APIs
* Dependent on access to the internet and no current plans for a mobile app.
* Time and budget could be a constraint for a lot of different aspects. For instance, Cloud hosting costs may restrict the volume of users the system can support.

### **Gantt Chart**

