# CS 255 Business Requirements Document

## 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?*

Purpose: The purpose of the DriverPass system is to provide comprehensive driver training to students preparing for their driving tests. The system will offer online practice exams and on-the-road training to better prepare students for their tests.

### 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 Background: DriverPass aims to address the lack of effective tools for training students to pass their driving tests. The system will provide online classes, practice tests, and driving lessons to enhance the preparation process.

### 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?*

Objectives and Goals: The objectives of the DriverPass system are to improve the pass rates of driving license exams by providing comprehensive training, offer convenient access to online resources for students, and enable efficient management of driving lessons and appointments.

## 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 DriverPass system should be web-based and accessible through modern web browsers.

• The system should have fast response times to ensure a smooth user experience. Page loads and form submissions should occur within a few seconds.

• The system should be updated regularly to incorporate new features, bug fixes, and security enhancements. Updates should be released at least once every three months.

#### 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 DriverPass system should be platform-independent and able to run on various operating systems such as Windows, macOS, and Linux.

• The back end of the system may require a database to store user information, driving lesson schedules, and payment details. The database should support high performance, data integrity, and scalability.

#### 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?*

• Each user should have a unique identifier, such as a username or email address, to distinguish between different users. The system should enforce case sensitivity where applicable.

• The system should inform the admin immediately when critical problems occur, such as database connection failures or server errors.

#### 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?*

• The system should allow IT administrators to make changes to user accounts, such as adding, removing, or modifying user information, without requiring changes to the underlying code.

• The system should be designed to adapt to platform updates, ensuring compatibility with new operating system versions, web browsers, and other related technologies.

• IT administrators should have privileged access to manage system settings, user accounts, and perform administrative tasks.

#### 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 should be required to log in with a unique username/email and password combination to access the DriverPass system.

• The system should use secure encryption protocols (e.g., HTTPS) to secure the connection and data exchange between the client and the server.

• The system should implement measures to detect and respond to brute-force hacking attempts, such as temporarily locking user accounts after a specified number of failed login attempts.

• If a user forgets their password, the system should provide a secure password recovery mechanism, such as sending a password reset link to the user's registered email address.

### 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 authenticate users by validating their credentials (username/email and password) when logging in.

• The system shall allow users to search for available driving lesson slots based on their location, preferred instructor, and date/time preferences.

• The system shall enable users to book driving lessons by selecting a preferred time slot and instructor, and provide confirmation of the booking.

• The system shall allow users to view and manage their upcoming and past driving lesson appointments, including the ability to reschedule or cancel appointments.

• The system shall provide users with access to educational resources, sample questions, and practice tests to help prepare for the driving license exam.

• The system shall send email notifications to users to confirm bookings, provide reminders for upcoming lessons, and notify about any changes or cancellations.

• The system shall generate reports for administrators, including statistics on lesson bookings, revenue, and instructor performance.

### 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.)?*

• The interface should cater to different user roles, such as students, instructors, and administrators, each with their specific needs and permissions.

• Students should be able to view their upcoming lessons, book new lessons, access educational resources, and manage their account details.

• Instructors should have access to their schedule, student profiles, and the ability to update lesson status (completed, canceled, etc.).

• Administrators should have access to system management features, including user management, lesson scheduling, and generating reports.

• The user interface should be accessible through web browsers on desktop and mobile devices, ensuring a responsive design for optimal user experience.

### 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?*

* The DriverPass system assumes that users have access to a reliable internet connection to access the online resources and make reservations.
* It is assumed that users will provide accurate and up-to-date personal information during the registration process.
* The system assumes that the driving lesson instructors and trainers are qualified and certified to provide training.
* It is assumed that the system will receive regular updates from the DMV regarding rule changes, policies, and sample questions.
* The system assumes that customers will adhere to the terms and conditions, including cancellation policies and payment requirements.

### 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?*

**Limitations in the system design:**

• The DriverPass system may have limitations in terms of scalability. If the number of users and driving instructors grows significantly, the system might experience performance issues or require infrastructure upgrades to handle the increased load.

• The system design assumes that users have reliable internet connectivity. If users face connectivity issues or have limited access to the internet, they may have difficulty accessing the system or booking driving lessons.

• The system design may not account for specific legal and regulatory requirements related to driving licenses and scheduling systems in different regions or countries. Adapting the system to comply with various local regulations may require additional development and customization.

**Limitations in resources, time, budget, or technology:**

• Resource limitations: The development team may have limited access to hardware resources, such as servers or testing devices, which can affect the testing and performance evaluation of the system.

• Time limitations: The project may have a strict timeline, which can limit the development team's ability to implement all desired features and conduct thorough testing. This can potentially lead to a trade-off between speed and the depth of system functionality.

• Budget limitations: The available budget for the project may impose constraints on the resources, technologies, or services that can be utilized. This could impact the scalability, security, or overall quality of the system.

• Technology limitations: The choice of technology stack and development tools may have inherent limitations in terms of compatibility with different platforms, integration with third-party services, or support for specific features. These limitations may affect the system's performance, flexibility, or extensibility.

### 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.*

[Insert chart]