# CS 255 Business Requirements Document

## System Components and Design

### Purpose

The consulting company wants to design a system for the new client: DriverPas. DriverPass needs to create a system that gives students access to online practice exams and on-the-road training to well prepare them for driving tests. The client pictures a complete system that permits users to take online classes, practice tests, and schedule driving lessons. They also want the system to enable data access from anywhere and highlights the importance of security, user roles, and permissions. They necessitate the ability to track user activities, like reservations and modifications, and generate reports for accountability. The system should facilitate the reservation process for driving lessons, by assigning drivers and monitoring the scheduled appointments. Different users of the system the owner (Liam), the IT officer (Ian), the secretary, and the customers (students) have specific needs, like access rights, appointment management, and data entry. Moreover, DriverPass wants to be up to date with DMV regulations and requests an interface that runs on the web with a visually attractive design.

### System Background

DriverPass identified the problem of insufficient driver training and a high failure rate in driving tests at the DMV. They think many students fail the driving tests due to insufficient appropriate preparation. To address this problem, DriverPass wants to offer a solution by designing thorough driver training for its customers. They wish to create a system that allows students to access online classes and practice tests and offer on-the-road training with professional instructors. By combining both online and practical training, DriverPass believes they can well prepare students for their driving tests and boost their chances of success. So, the components needed for the system are Online Classes and Practice Tests, Appointment Scheduling, Driver-Customer Matching, On-the-Road Training, Flexible Package Options, Registration and Customer Information, Data Accessibility, Security and User Roles, Activity Monitoring, DMV Agreement, Web-based Interface, and a specific Interface Design.

### Objectives and Goals

|  |  |
| --- | --- |
| Objective | Measurable Tasks |
| Online Classes and Practice Tests | * Interface for retrieving online classes and practice tests. * Instructional material for the classes. * System for tracking progress in practice tests. |
| Appointment Scheduling | * Interface for customers to schedule appointments. * Real-time system showing available times for driving lessons. * Notification mechanisms to remind upcoming appointments. |
| Driver-Customer Matching | * Matching algorithm utilizing driver availability, location, and customer preferences. * Tracking system monitoring driver assignments. * Mechanism for customers to view assigned driver and related details. |
| On-the-Road Training | * Tracking mechanism to monitor the driving lessons. * Feedback system to get comments on driving sessions. |
| Flexible Package Options | * Administration panel managing driving lesson packages. * Functionality for adding, removing, or modifying packages. |
| Registration and Customer Information | * Registration forms acquiring essential customer details. * Storage mechanisms to protect sensitive customer data. |
| Data Accessibility | * Cloud-based structure for hosting the system. * Offline data synchronization capabilities |
| Security and User Roles | * User management system that allows administrators to assign and manage user roles. * Secure authentication and authorization mechanisms to protect user accounts. |
| Activity Monitoring | * Activity logging system that tracks user interactions. * Version control and revision history tracking changes in system data. |
| DMV Agreement | * Regular reviews of DMV regulations and guidelines to check system compliance. * Features and processes to meet DMV requirements for driver training and licensing. |
| Web-based Interface | * Responsive web interface that works smoothly across different devices. |
| Specific Interface Design | * Interface agreeing to the given key components, layout structure, and information display requirements. |

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

The DriverPass system requires a web-based environment, allowing users to access it through various web browsers on different devices like desktops, laptops, tablets, and smartphones. The system's performance should be optimized to ensure a fast and responsive user experience. For instance, pages should load swiftly. Efficient processing of user interactions and prompt responses are crucial to minimize waiting times. The system should be designed to handle concurrent user sessions and scale resources as needed, maintaining optimal performance even under high user traffic. Regular updates and bug fixes should be flawlessly deployed to improve functionality without disrupting user access. Database performance optimization, including indexing and caching techniques, should be employed for efficient data retrieval and storage. Network performance should be optimized to house different network conditions, ensuring smooth operation even with high latency or low bandwidth connections.

#### Platform Constraints

The DriverPass system must be compatible with various platforms, including Windows, Unix, and other generally used operating systems. It should be able to run faultlessly on these different platforms to ensure accessibility for users regardless of their preferred operating system. The system will also rely on a database to support its application. The database will serve as a central storage for user information, driving lesson schedules, test results, and system configurations. In addition, we may require additional tools and technologies such as web servers, application servers, and programming frameworks to facilitate efficient data processing, server-side operations, and smooth integration with the main interface.

#### Accuracy and Precision

To differentiate between various users within the DriverPass system, distinct identifiers like usernames or email addresses can be assigned during the registration process, enabling user authentication, and defining individual profiles and access privileges. If differentiation based on uppercase and lowercase characters is necessary, the system should consider input as case-sensitive; otherwise, it can be treated as case-insensitive for enhanced user convenience. The system should be further designed to promptly alert the administrator in real time whenever critical issues arise, including system malfunctions, database errors, unauthorized access attempts, and other significant problems.

#### Adaptability

The DriverPass system offers the flexibility to modify user accounts without the need for modifying the system's underlying code via a user-friendly administrative interface that empowers administrators to add, remove, and modify user details as needed. The system is designed to adapt to platform updates by adhering to industry standards, utilizing scalable frameworks, and following modular architectural patterns. This ensures compatibility with the latest operating system versions, database releases, and web server software. By actively monitoring updates and maintaining compatibility, the system can incorporate platform changes while minimizing disruptions. The IT administrator is granted comprehensive access and elevated privileges to effectively manage and maintain the system. This includes configuring system settings, performing database administration tasks, monitoring performance, managing user accounts and permissions, troubleshooting technical issues, and implementing security measures.

#### Security

To access the DriverPass system, users need to provide their unique username and password. To ensure a secure connection and protect data exchange, the system employs HTTPS, which encrypts data during transmission and implements SSL/TLS certificates to verify the server's authenticity. In the case of a brute force hacking attempt, where unauthorized access is repeatedly attempted using various username/password combinations, the system incorporates security mechanisms temporarily locking an account or introducing delays between login attempts after a certain number of failed tries. In the event that a user forgets their password, the system offers a password recovery process such as sending a password reset link to the registered email or answering security questions established during account creation.

### Functional Requirements

The DriverPass system must first perform user credential validation during the login process to ensure secure access. Additionally, the system should offer an intuitive online registration procedure for new users, enabling them to create their accounts easily. Users should be able to schedule and manage their driving tests conveniently through the system, which should generate and present online tests for them to complete. During these tests, the system must accurately record and track users' progress to provide an efficient assessment. Furthermore, users should have the ability to submit driver notes and observations, contributing to a comprehensive evaluation process. An administrative dashboard should be available to authorized personnel, allowing them to efficiently manage user accounts and test-related data. The system should also facilitate timely notifications to users regarding their test schedules and any updates or changes. Differentiating between user roles, such as administrators and regular users, the system should provide varying levels of access privileges to maintain security and data integrity.

### User Interface

The interface of the DriverPass system caters to the diverse needs of its users (administrators, driving instructors, and learner drivers). Administrators utilize an administrative dashboard to manage user accounts, monitor test progress, and track system performance. Driving instructors access a dedicated interface to schedule and oversee driving tests, track learner driver progress, and provide feedback. Learner drivers rely on the interface to register, schedule and complete online tests, view their progress, submit driver notes, and receive notifications. To ensure accessibility, the interface is designed to be responsive and adaptable across various devices such as desktop computers, laptops, tablets, and mobile phones. It accommodates different screen sizes and resolutions to deliver a seamless experience.

### Assumptions

In the previous design, user accessibility, localization, internationalization support, system compatibility with various devices and configurations, specific security measures for data protection, and user training and support were not explicitly addressed. So, assumptions made in the design relate to the users' accessibility needs, language preferences, available technology, and familiarity with the system. To guarantee a comprehensive solution, it is important to further explore and validate these aspects during the development and testing phases, addressing user accessibility requirements, accommodating different languages and regions, ensuring compatibility across devices and platforms, implementing robust security measures, and providing user training and support resources.

### Limitations

One limitation is the potential challenges that may arise in scaling the system as the user base expands. It may require additional resources and infrastructure enhancements to ensure smooth performance. Additionally, the project's timeline and budget constraints may restrict the scope of system features and functionalities that can be developed and implemented. Technological limitations can also arise based on the chosen platform or development tools, potentially impacting the system's adaptability and compatibility across different environments. Furthermore, it's important to recognize that the design may not account for all possible cases and user scenarios, which could result in gaps in functionality and user experience.

### Gantt Chart

A picture containing text, line, plot, diagram

Description automatically generated