# CS 255 DriverPass Business Requirements Document

## System Components and Design

### Purpose

* The purpose of this project is to design and develop a comprehensive driver training system for the client, DriverPass, that will help fill the void in the market for quality driver training.
* DriverPass is a company that aims to provide online classes and practice tests, as well as on-the-road training for customers who wish to take their driving test at the local DMV.
* The client wants their system to allow customers to make reservations for driving lessons online, over the phone, or in-person, and to identify the driver and car assigned to each customer for their lesson.
* The client also wants the system to provide different levels of access and roles for employees, track user activity and changes to records in the system, and collect customer information, including their credit card information.
* The system should integrate with DMV updates to ensure that the tests and practice provided are current with the DMV requirements. The client wants the system to be easy to use, secure, and flexible enough to accommodate future modifications.

### System Background

What does DriverPass want the system to do?

* DriverPass wants the system to provide a comprehensive driver training solution for customers who wish to take their driving test at the local DMV.
* The system should allow customers to take online classes and practice tests, as well as on-the-road training if they wish.
* Customers should be able to make reservations for driving lessons online, over the phone, or in-person, and the system should identify the driver and car assigned to each customer for their lesson.
* The system should collect customer information, including their credit card information, and integrate with DMV updates to ensure that the tests and practice provided are current with the DMV requirements.

What is the problem they want to fix?

* DriverPass has identified a void in the market for quality driver training and aims to fill that gap by providing a comprehensive driver training solution.

What are the different components needed for this system?

* Online classes and practice tests
* On-the-road training
* Reservations system (online, over the phone, in-person)
* Driver and car assignment tracking
* Customer information collection and storage
* Integration with DMV updates
* Different levels of access and roles for employees
* User activity and changes tracking
* Security features
* Interface design
* Database tables
* Business logic layer
* Testing and sign-off process.

### Objectives and Goals

What should this system be able to do when it is completed?

* The system should allow customers to take online driver classes and practice tests.
* Customers should be able to reserve driving lessons online, over the phone, or in-person.
* The system should identify the driver and car assigned to each customer for their lesson.
* The system should collect and store customer information, including credit card details.
* The system should integrate with DMV updates to ensure that the tests and practice provided are current with the DMV requirements.
* The system should allow different levels of access and roles for employees.
* The system should track user activity and changes.
* The system should have security features to ensure that sensitive information is protected.
* The system should have an interface design that is user-friendly and intuitive.
* The system should have a business logic layer to manage the security, roles, and rights of employees.

What measurable tasks need to be included in the system design to achieve this?

* Develop online classes and practice tests
* Build reservation system (online, over the phone, in-person)
* Create driver and car assignment tracking system
* Develop customer information collection and storage system
* Integrate with DMV updates to ensure compliance
* Implement different levels of access and roles for employees
* Implement user activity and changes tracking
* Add security features to ensure the protection of sensitive information
* Design an intuitive and user-friendly interface
* Develop business logic layer to manage employee security, roles, and rights.

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

* The system needs to run on the web, preferably over the cloud, according to Ian, the IT officer from DriverPass.
* The system needs to be designed to handle multiple users making reservations and accessing the online classes and practice tests simultaneously.
* The system needs to be designed to handle a moderate to heavy load and provide acceptable response times for users.
* It is important to keep the system up-to-date with the latest DMV rules and policies, as well as security updates.
* The system should be updated regularly to ensure that it remains compliant and secure.
* The frequency of updates will depend on the rate of changes in DMV policies and security threats.

#### Platform Constraints

* The system should be designed to run on the web using a cloud-based platform such as Amazon Web Services (AWS) and will be Linux based
* The back-end of the system will require a database to support the application.
* The database system should be able to handle large amounts of data and support the required functionality of the system, such as tracking reservations and test results.
* AWS provides a variety of database options such as Amazon RDS, Amazon DynamoDB, Amazon Aurora, and Amazon Redshift, among others.

#### Accuracy and Precision

* To distinguish between different users, use a unique identifier such as a username or email address.
* If case-sensitivity is required, the system should validate the input to ensure that it matches the correct case.
* The system should inform the admin of a problem as soon as it occurs. This can be achieved by setting up notifications that alert the admin when a problem is detected.
* The admin should also be able to access logs and reports that provide information about any errors or issues that have occurred within the system.

#### Adaptability

* The system should allow the IT admin to add, remove or modify users without changing the code.
* The system should also provide appropriate user access control levels to ensure data security and privacy.
* The system should be built to ensure compatibility with platform updates, with minimal or no code changes required.
* The system should be able to adapt to new versions of the platform or technologies used, without causing any disruptions to the user experience.
* The IT admin should have full access to the system, including the ability to modify, add, or remove functionalities as required.
* The IT admin should also be able to access data analytics and generate reports to monitor system performance, usage, and security.

#### Security

* Login Requirements:
  + The user needs to provide a valid username and password.
  + The system should also support multi-factor authentication, such as using a security token or a one-time password.
* Connection and Data Security:
  + The system should use encryption to secure the connection between the client and server, such as SSL/TLS.
  + Data exchanged between the client and server should also be encrypted.
  + Access to sensitive data, such as passwords and credit card information, should be restricted to authorized users only.
* Brute Force Attack Prevention:
  + If there is a brute force hacking attempt, the system should lock out the account after a certain number of failed login attempts.
  + The account should remain locked out for a specified period of time before the user can attempt to log in again.
* Password Recovery:
  + If the user forgets their password, the system should provide a password recovery mechanism, such as sending a password reset link to the user's email.
  + The password reset link should expire after a certain period of time to ensure security.

### Functional Requirements

* The system shall allow users to create a new account.
* The system shall allow users to log in with their username and password.
* The system shall validate user credentials when logging in.
* The system shall allow drivers to enter their license and insurance information.
* The system shall verify the validity of driver's licenses and insurance policies.
* The system shall allow the admin to review and approve driver information.
* The system shall allow the admin to reject driver information and provide a reason for rejection.
* The system shall allow the admin to add or remove users.
* The system shall allow drivers to schedule and manage their driving tests.
* The system shall allow the admin to view upcoming and completed driving tests.
* The system shall notify drivers of their test results.
* The system shall keep a record of all driver information, including test results.
* The system shall allow the admin to generate reports on driver information and test results.

### User Interface

* Needs of the interface: The interface needs to be user-friendly and easy to navigate, with clear visual cues and instructions. It should also be responsive and accessible across different devices and platforms. Security is also an important consideration for the interface.
* Different users: There are several different types of users who will interact with the interface, including:
  + Drivers who need to access and update their personal information, view their driving record, and complete training modules.
  + Fleet managers who need to manage their drivers' information and access reports on their driving records and compliance status.
  + IT administrators who need to manage user accounts and access system logs and analytics.
* User tasks: Each user will need to be able to perform different tasks through the interface, such as:
  + Drivers: view and update personal information, complete training modules, view driving records and compliance status.
  + Fleet managers: view and manage driver information, access reports on driving records and compliance status.
  + IT administrators: manage user accounts, access system logs and analytics.
* User interaction: The interface should be accessible through a web browser, as well as through mobile devices. The interface should be designed to be easy to use and navigate on both desktop and mobile devices. It should also be optimized for different screen sizes and resolutions.

### Assumptions

* Users have basic computer skills and are able to navigate the web-based user interface.
* Users have access to a computer or mobile device with internet connectivity.
* The system will be built using modern web technologies.
* The system will be hosted on AWS cloud platform.
* The system will be integrated with a secure payment gateway for processing transactions.
* The system will be able to handle a high volume of users and transactions.
* The system will comply with relevant data protection laws and regulations.
* The system will have user roles with different levels of access and permissions.
* The system will be designed with a responsive user interface to support different screen sizes and devices.

### Limitations

* The accuracy and precision of the system may be affected by factors outside of the system's control, such as the quality of the images provided by the user.
* The system may be limited in terms of the number of users it can support simultaneously, depending on the chosen infrastructure and the resources available.
* The adaptability of the system may be limited by the platform chosen, as some platforms may require more effort to modify or update than others.
* The security of the system may be limited by the level of encryption or other security measures available, and by the potential for human error or social engineering attacks.
* There may be limitations in terms of the resources, time, and budget available to develop and maintain the system, which could impact the complexity and scope of the project.

### Gantt Chart

Table, timeline

Description automatically generated