William Fishburn  
Project One

## 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 purpose of the project is to develop a comprehensive software solution for our Client DriverPass, a driver training and driving test preparation company.
* Driver Pass aims to use this system for streamlining and enhancing core competencies of the business, including online classes, practice tests, on-the-road training, and appointment scheduling. It should have a user-friendly interface for Administrators and Customers, along with efficient data management, security, and compliance with DMV regulations. Administrators should be able to modify their training packages and offerings at will.

### 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?*

* Driver Pass wants the system to address issues related to poor test preparation driver training.
* Problems DriverPass would like to fix::  
  + High Test Failure Rates: The primary issue DriverPass wants to fix is the high failure rates among those taking driving tests at the Department of Motor Vehicles (DMV). Many struggle to pass these tests, which can cause individuals stress, as well as clog the system with repeat students and hinder the cycle of incoming students.
  + Lack of Effective Driver Training: DriverPass recognizes a need for more effective and comprehensive driver training. Some may have limited access to training materials and resources needed to prepare for driving tests. DriverPass wants to provide easy access to learning materials, online classes and practice tests.
  + Poor Scheduling and Appointment Management: DriverPass wants to streamline the scheduling of driving lessons and appointments. The current system may involve manual scheduling and administrative work, which is much less efficient.
* The different components needed for this system are:
  + User Interface (UI): The UI is an intuitive and user-friendly front-end that allows users to interact with the system. This includes web pages for registration, login, scheduling appointments, accessing training materials, and tracking progress. It should resemble Liam's design sketch.
  + User Management: This component handles user registration, authentication, and account management. It includes functions for users to create accounts, log in securely, reset passwords, and manage their profiles.
  + Appointment Scheduler: The appointment scheduling component allows users to schedule driving lessons. It should provide a calendar-based interface for selecting preferred dates and times, checking instructor availability, and confirming appointments.
  + Lesson Management: This component manages the different driving lesson packages offered by DriverPass. It should support the creation, modification, and disabling of training packages, as well as associating each package with the available instructors and cars.
  + Progress Tracking: Progress tracking is a critical component that records and displays users' progress, including completed and ongoing tests, scores, and other relevant information. It should provide a clear overview of a user's journey through the training program.
  + Database Management: The database component is responsible for storing and managing all data related to users, appointments, lesson packages, progress tracking, and administrative information. It ensures the integrity and accessibility of data.
  + Integration with DMV: This component establishes a secure connection with the Department of Motor Vehicles (DMV) to receive updates on rules, policies, and sample questions. It should handle notifications and updates from the DMV to keep the training materials current and compliant.
  + Security and RBACl: Security is a critical component responsible for safeguarding user data, including personal information and payment details. It enforces role-based access control to restrict access to sensitive data and functionalities.
  + Cloud Infrastructure: The system should be hosted on a cloud-based infrastructure for scalability, reliability, and ease of maintenance. This component ensures that the system is accessible from various devices and web browsers.
  + Reporting and Logging: Storing user activity data for auditing and tracking. It enables administrators to generate activity logs for forensic, chain-of-custody purposes.
  + Business Logic Layer: The business logic layer handles the core functionality of the system,such as role based access management, CRUD operations, and data processing. It ensures that the system operates according to the specified rules and requirements.
  + User Support: User support and communications, such as chat capabilities, email correspondence for customer inquiries or issue resolution, to maintain a consistent positive user experience.

### 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?*

* Objective: The system should provide comprehensive and effective driver training.
  + Measurable Task: Design an intuitive user interface for accessing online classes, practice tests, and educational content. Ensure that training materials align with DMV requirements and are regularly updated.
* Objective: Increase the pass rates of individuals taking driving tests at the DMV.
  + Measurable Task: Implement progress tracking features to help users identify areas for improvement. Measure the pass rates of users who have completed the DriverPass training program compared to those who haven't.
* Objective: Streamline the process of scheduling driving lessons and managing appointments.
  + Measurable Task: Design an appointment scheduling system that allows users to select preferred dates and times and checks instructor availability in real-time. Measure the reduction in the time it takes to schedule appointments compared to the previous process.
* Objective: Improve the overall user experience for both customers and administrators.
  + Measurable Task: Conduct user testing to assess the ease of use and navigation of the user interface. Collect user feedback and track user satisfaction scores.
  + Measurable Task: Develop a system feature that allows administrators to easily modify and disable training packages. Measure the frequency of package modifications and the level of customer satisfaction with package customization.

* Objective: Secure user data and maintain compliance with DMV regulations.
  + Measurable Task: Implement encryption mechanisms for data transmission and storage. Monitor and track security incidents and compliance updates. Measure the frequency and impact of security incidents before and after system implementation.
* Objective: Provide effective user support and communication channels.
  + Measurable Task: Monitor response times for user inquiries and measure user satisfaction with support services. Track the number of support requests and resolutions.
* Objective: Ensure the system is scalable and reliable.
  + Measurable Task: Monitor system performance under varying loads to ensure scalability. Measure system response times and resource utilization to optimize cloud infrastructure.
* Objective: Maintain transparency/accountability through reporting and auditing.
  + Measurable Task: Implement logging for all system activities. Generate and review activity reports to ensure compliance.

## 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 needs to run in a web-based environment to ensure accessibility from various devices and web browsers.
* The system should be responsive to user interactions and aim to load pages and perform tasks within 2 seconds for an optimal user experience. Database queries for customer information should be completed within 1 second to ensure quick access to data.
* The system should be updated regularly to maintain compliance with DMV regulations, security, and functionality. Update frequency will depend on DMV policy changes and system improvements.

#### 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 system should be platform-agnostic in terms of operating systems, making it accessible from platforms such as Windows, Unix, macOS, and mobile platforms. The focus is on web-based accessibility, which is inherently cross-platform.
* The back end of the system requires tools such as a database management system (DBMS) to support data storage, retrieval, and management efficiently. Tools such as Java SpringBoot, JPA and hibernate can be used to provide a back-end framework for web-based CRUD operations.

#### 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?*

* Users should be distinguished by unique usernames, email addresses and passwords, which will be treated as case-sensitive. Admin accounts may only be added to the system by Global Administrators such as Liam.
* The system should promptly notify the Admin of any problem via email and/or text communication.

#### 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?*

* Adding, removing and/or modifying users should be available without changing code.
* The system should be capable of handling browser-based, as well as OS-level system updates.
* The IT admin should have full access and administrative privileges into the system, including the ability to reset passwords, manage user accounts, as well as system updates and monitoring.

#### 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?*

* User authentication should require valid login credentials, typically a username (or email) and a password. Multi-factor authentication (MFA) should be implemented for an additional layer of security.
* Data exchange between the client and the server should be secured using encryption protocols such as DNS Filtering, HTTPS (SSL/TLS) to encrypt data in transit.
* The system should have protection in place to detect and respond to brute force hacking attempts, such as temporarily locking out an account after a certain number of failed login attempts, and the use of CAPTCHA prompts during authentication.
* If a user forgets their password, they should have the option to reset it by sending a password reset link to their registered email address or by way of security questions.

### 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 Users to register for an account by providing their first name, last name, address, phone number, state, and credit card information (including card number, expiration date, and security code), and enable the user to log in securely with their credentials..
* The system shall provide a scheduling system for Customers to book two-hour long driving lessons, allowing the Customer to specify their preferred date/time for the lesson.
* The system shall support multiple lesson packages:
  + Package One: 6 hours with a trainer
  + Package Two: 8 hours with a trainer and in-person lesson
  + Package Three: 12 hours with a trainer, in-person lesson, and access to online content and practice tests
  + The system shall allow Liam to customize and disable these packages at will.
* The system shall allow users to have separate roles and permissions:
  + Liam should have full system access with no restrictions.
  + Ian should have full access to all accounts for password resets and account management.
  + Users should be able to make appointments, cancel appointments, and modify appointments online.
* The system shall record and maintain a log of activities, including who made reservations, cancellations, and modifications.
* The system shall be designed as per Liam's sketch, displaying progress for completed and ongoing tests, including test name, time taken, score, and status.
  + Users should find the interface intuitive with clear navigation menus.
* The system shall be connected to the DMV to receive updates on rules, policies, and sample questions. It should notify administrators whenever there are updates from the DMV.
* The system shall be hosted in the cloud for scalability and reliability. It should be accessible from various devices and web browsers.
* The system shall provide password reset functionality for Users who forget their credentials.

### 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.)?*

* Customer:

Needs: Customers are individuals seeking driver training. They need access to the following features:

* + Register for an account.
  + Log in securely.
  + Schedule driving lessons and appointments.
  + Access online classes and practice tests.
  + Track their progress, including completed and ongoing tests.
  + View appointment details and instructor assignments.
    - Interaction: Customers will primarily interact with the interface through web browsers on desktop computers or mobile devices.
* Administrator:

Needs: The Administrator is responsible for system maintenance and user management. Admins need access to administrative functionalities, including:

* + User account management: User creation, deletion, modification and password resets.
  + Managing user roles and permissions.
  + Activity Log extraction
  + Modification of Lesson Packages displayed and available for selection on the website.
  + Monitoring system health and security.
    - Interaction: Admins may use a web-based administrative portal accessible via web browser.
* Instructor:

Needs: Instructors are responsible for providing driving lessons.They may need limited access to:

* + View lesson schedules and Students(Customers)they are assigned to.
  + Communicate with Admins to report critical or urgent matters.
  + Provide feedback regarding Student performance.
    - Interaction: Instructors may interact with the interface through a mobile application, which allows them to access their schedules and submit feedback on the go.

### 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 system assumes a reliable internet connection for users accessing online materials and making appointments.
* Users are responsible for providing accurate registration information, personal details and payment information.
* DMV compliance updates will be made available through a secure data connection between DriverPass and the DMV.
* Customers are physically capable of operating a motor vehicle.
* Customers are not intoxicated nor impaired during each phase of the sales cycle.

### 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?*

* While the system will offer flexibility for modifying training packages in the future, major structural changes may require Software Developer involvement.
* The system may not support every web browser; compatibility with major browsers is priority.
* The system's performance may vary based on the user's internet connection and device capabilities, which are beyond the system's control.
* The system employs embedded modern security methods, however the system will require regular monitoring and updates to maintain security.

### Gantt Chart

