# CS 255 Business Requirements Document

## System Components and Design

### Purpose

This application will be developed for DriverPass, which wants to accomplish the following:

* Provide online DMV test practice
* Provide better driving training for students

By creating a web-hosted application that allows users to schedule appointments and access educational material. It will update with any changes in rules or policies from the DMV. It will offer three levels of packages and will keep track of user information, mentor information, and utilization of available vehicles and mentors.

### System Background

DriverPass has discovered a void in practice and preparation for DMV driving tests. It aims to fill that void by providing customers with purchasable packages that will allow them to get behind-the-wheel driving training, as well as in-person and online lessons and practice tests. They want to focus on the management of this company with as few technical issues or responsibilities as possible, therefore this system will be implemented on a cloud-based platform with automated backup and security features. The company will offer different packages to the user that will allow them to participate in practice sessions and access prepared educational material.

### Objectives and Goals

Once completed, this system should be able to:

* Provide online data access, as well as downloadable reports
* Track modifications with user signatures
* Allow the creation, storage, and modification of driving lesson reservations, including customer and mentor information
* Implement the rule of least privilege (president, IT officer, secretary, users)
* Process payments made by users based on differing packages
* Disable a purchasable package
* Handle authentication and automated user password resets
* Maintain user progress and completion information, as well as notes and photos
* Display multiple functioning pages (user information page, user input form, company contact page, user contact page)
* Provide notifications with updates to DMV rules, policies, or sample questions

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

* The online DMV test practice and driving training system needs to run in a web-based environment as it will be accessed through a web browser. Users will interact with the system through a web application hosted on a cloud-based platform.
* The system should have fast response times to ensure a smooth user experience. Actions such as page loading, form submissions, and accessing educational material should reach completion within two seconds.
* As far as system updates, it is important to maintain a schedule to remain up-to-date with any changes in DMV rules, policies, and sample questions. The frequency of updates relies on the frequency of DMV updates. It is also important to promptly address any security vulnerabilities and bug fixes, as well as performance issues.

#### Platform Constraints

* The system should be designed to run on multiple operating systems to ensure broad compatibility. This includes Windows and Unix systems such as Linux and macOS. The system should be compatible with a range of web browsers to ensure accessibility for users. Popular web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc. should be expected. Compatibility with both desktop and mobile browsers should be implemented.
* Additionally, a database is certainly going to be required to support this application. Leveraging cloud platforms (such as Microsoft Azure) can provide a scalable and reliable infrastructure for hosting the system. This can offer benefits such as automatic scaling, managed database services, and built-in security features, as well as a pay-as-you-go structure that only requires the purchase of what is used.

#### Accuracy and Precision

* To distinguish between different users, unique identifiers or usernames for each user can be used. This ensures that each user has a distinct identity within the system. The system can assign a unique user ID or allow users to choose their usernames during the registration process. This way, users can log in with their credentials to access their accounts and track their progress.
* User IDs and passwords should be case-sensitive, while input fields like name, address, contact information, etc. should not be. To avoid confusion, case sensitivity should be communicated to the user for any given input field.
* As for notifying the admin of a problem, the system should have the appropriate error handling and alert mechanisms in place. Events such as critical errors (server failures, database connection issues, security breaches), user-reported issues, system performance degradation (increased response times, high server load), or data integrity issues (incomplete or inconsistent data) should trigger alerting and notification mechanisms, which ensure that the admin stays informed about issues and can take appropriate actions to address them properly.

#### Adaptability

* The interface should provide an administrative interface that allows authorized personnel (such as an IT admin or system administrator) to manage users without requiring changes to the underlying code (adding new users, removing existing users, modifying user information like contact details or access privileges, etc.). The administrative interface should provide the functionality to handle these user management tasks effectively.
* The system should be built in a completely modular fashion, which allows for adaptation to platform updates to the underlying web server, database, or framework and facilitates future enhancements or modifications without affecting the entire codebase. Regularly applying updates and patches to the platform components helps ensure compatibility and maintain system performance and security.
* The IT admin should have appropriate access privileges and permissions to manage the system effectively, including administrative access to the server infrastructure, database management, configuration settings, and user management. Although the admin should have a greater level of privileges, it is still important to follow the principle of least privilege, and only grant the admin access to the necessary components and functions required for their role.

#### Security

* For the user log-in, the system should enforce strong password policies, including complexity requirements (minimum length, alphanumeric/special character combination) and account lockouts after multiple failed login attempts.
* The system should use encryption protocols such as HTTPS to encrypt and protect exchanged data, which should be regularly renewed. Any sensitive data that is stored should be in the database should be encrypted.
* To mitigate brute force hacking attempts, the system should implement mechanisms such as account lockouts or temporary IP blocking after a certain number of failed login attempts within a certain period. This will help prevent automated password guessing or brute force attacks. Implementing CAPTCHA challenges can further protect against automated login attempts.
* To further prevent unauthorized access, a password reset link or temporary password can be sent to the user’s validated email address, or security questions can be implemented during the account setup. These strategies can be used to recover an account that has been blocked or compromised, or if the user forgets their password.

### Functional Requirements

* The system shall allow new account creation with user-entered data such as name, contact details, and username/password.
* The system shall provide secure logins for users using the validation of their credentials.
* The system shall provide users with profile management, including updating information and preferences.
* The system shall offer different packages to users, allowing them to select and purchase the desired package for access to driving training, practice sessions, and educational material.
* The system shall allow the scheduling of driving lessons and practice sessions based on availability, selected preferred dates, times, and mentors.
* The system shall provide access to a comprehensive repository of educational materials, including study guides, sample questions, and instructional videos.
* The system shall allow the creation, modification, and cancellation of reservations for driving lessons, specifying the mentor, vehicle, and preferred location.
* The system shall track and display user progress and completion information, including completed lessons, practice sessions, and overall training progress.
* The system shall facilitate secure payment processing for users, allowing them to purchase packages and make payments online.
* The system shall provide timely notifications and updates to users regarding changes in DMV rules, policies, and sample questions.
* The system shall Generate downloadable reports, allowing users or administrators to access data such as user progress, lesson reservations, and package utilization.
* The system shall provide communication channels for users to contact mentors, customer support, or other relevant personnel for inquiries, assistance, or feedback.
* The system shall provide an administrative dashboard for authorized personnel to manage user accounts, view system analytics, and perform administrative tasks such as user management and content updates.
* The system shall log relevant user activities, errors, and security-related events for auditing, monitoring, and troubleshooting purposes.
* The system shall provide a secure mechanism for users to reset their passwords if they forget them, using methods such as email verification or security questions.

### User Interface

* The interface should be intuitive, user-friendly, and visually appealing to provide a positive user experience. It should facilitate easy navigation, efficient access to features, and clear presentation of information. It should also be responsive and accessible across different devices and screen sizes.
* One type of user is the student, who needs to access educational materials, schedule appointments, track progress, access practice tests, and interact with mentors and support staff. The tasks they will need to perform through the interface are to register for new accounts, log into their accounts, select and purchase packages, schedule lessons, access educational material, track progress, communicate with mentors or support staff, and receive notifications and updates.
* Another type of user is the mentors/instructors. These users are responsible for providing driving lessons and guidance to students, and they need access to student information, lesson schedules, progress tracking, and the ability to communicate with students. They will need to access their assigned student’s information, lesson schedules, student progress, and communication channels with students, as well as update lesson statuses or provide feedback.
* Administrators need to have higher-level access and control over the system. They manage user accounts, monitor system performance, generate reports, configure system settings, and ensure smooth operation. They will need access to user management functionalities, system configuration settings, analytics and reporting tools, content management, and the ability to monitor and manage system activities.
* The interface should be designed to accommodate different devices and user preferences. The system should be accessible through both web browsers on a desktop or laptop computer and mobile devices. This includes popular browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, as well as dedicated mobile applications or mobile-optimized web interfaces for both Android and iOS platforms, providing a seamless and optimized experience on smartphones and tablets.

### Assumptions

* The design assumes that users have a basic level of technical proficiency to navigate and interact with the system.
* It assumes that users are familiar with common tasks such as registration, login, and using web-based interfaces.
* It assumes that users have access to a stable internet connection to interact with the system. The online features of the system rely on internet connectivity.
* It assumes that users also have access to compatible devices to interact with the system, such as desktop or laptop computers, smartphones, or tablets capable of running web browsers or dedicated mobile applications.
* It also assumes that users are comfortable using one specific language, or whichever languages are included in the design, for registration, interface navigation, educational materials, and communication.
* It assumes there will be timely updates regarding DMV rules, policies, and sample questions, and that the updates will come from reliable sources and be integrated into the system quickly.
* Finally, it assumes that the system will comply with relevant data privacy and security regulations, such as handling user data securely, protecting sensitive information, and obtaining necessary consent.

### Limitations

* This system requires various resources, including skilled developers, designers, content creators, and system administrators. Limited availability of resources, both in terms of personnel and infrastructure, may impact the implementation, scalability, and ongoing support of the system.
* Developing a robust and feature-rich system can be time-consuming. Meeting project deadlines and time constraints may require prioritization of certain features, and balancing development efforts with testing and quality assurance to ensure a reliable and user-friendly system.
* Limited financial resources may impact the scope of development, the level of customization, the integration of advanced features, or the selection of third-party tools and services.
* The system may rely on specific technologies, frameworks, or platforms. Limitations may arise if there are compatibility issues with certain devices, operating systems, or web browsers. Upgrading or adapting the system to newer technologies or addressing compatibility challenges can incur additional development efforts and costs.
* As the user base and demand for the system grow, scalability becomes a potential limitation. The system should be designed to handle increased user traffic, database size, and concurrent user interactions. Ensuring optimal performance, responsiveness, and scalability requires careful architectural planning and consideration of infrastructure resources.
* The system design involves integration with external components such as payment gateways, DMV data sources, and third-party services. Depending on the availability and compatibility of these integrations, there may be complexities in implementation and ongoing maintenance.
* Ensuring robust security measures and protecting user data is crucial. However, there may be limitations, in terms of budget or timeframe, to implement advanced security measures. Balancing the need for security with usability and convenience is vital to maintaining user trust and complying with privacy regulations.
* The system needs to adhere to relevant regulations and legal requirements, such as data privacy laws or DMV guidelines. Compliance with these regulations may introduce limitations or constraints on certain system functionalities or data-handling processes.

### Gantt Chart

*A screenshot of a gantt chart

Description automatically generated with low confidence*