**CS 255 Business Requirements Document J. Hasbrook**

**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?*

* *Client*: Liam, the owner of the company DriverPass
* *Project Purpose*: to solve the high failure rate of driving examinations at the DMV by offering an elaborate program that includes online lessons, practice tests, and on-road instruction. The service promises to be available both online and offline, increasing learning possibilities and convenience for users.

**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?*

* *Primary Problem*: High failure rates at driving exams due to inadequate preparation and practice.
* **System Requirements:**

**-** *Offline Accessibility: Ensure that users may access the system without an internet connection while keeping data synced to avoid redundancy.*

*- Authentication Access: Implement a system that grants different categories of users like students, teachers, and administrators, access based on their responsibilities.*

*- Secure Online Registration: Protect user data by facilitating a safe procedure for account creation and package reservations.  
  
 - Flexible Package Selections: Provide clear alternatives for package selections, with the ability to change or add driving hours as required.  
  
 - Cloud-Based Operations: Use cloud technology for scalability, data backup, and security.*

*- DMV Integration: Ensure that the system complies with DMV regulations and automatically updates any regulatory changes.*

**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?*

* *Enhance Learning Outcomes:* The system will minimize DMV exam failure rates by delivering extensive online and offline learning resources such as courses, practice exams, and on-road training. This can be measured by the system tracking and reporting on user progress and exam results to determine how much improvement has occurred over time.
* *Streamline the Reservation and Package Customization Process:* Allow users to effortlessly book, change, or cancel driving sessions and align their learning packages to their specific needs. This can be measured by monitoring the average time users spend booking or modifying reservations, and strive to reduce it through interface enhancements.
* *Maintain Compliance with DMV Regulations:* Regularly update the system to reflect the most recent DMV policies and regulations, ensuring that all learning materials are up to date. This can be measured by implementing a system for automated updates and notifications of DMV changes, with compliance checks done weekly.
* *User-Friendly Interface:* Create an intuitive and accessible user interface that meets the demands of all users, including students, teachers, and administrators. This can be measured by collecting
* user input on interface usability and utilize it to create improvements, with the goal of achieving a good user experience rating of 90% or above.

**Requirements**

**Nonfunctional Requirements**

*This section 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?*

* *Environment*: The system should be optimized for both online and application contexts to provide accessibility on multiple devices.
* *Speed: Interactions should be quick, aiming for real-time or near-real-time where possible.*
* *Update Frequency: The system should be updated at least quarterly to include new features, issue corrections, and regulatory changes.*

**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?*

* *Platform Support: The application should work with major operating systems including Windows and MacOS for desktop computers and iOS and Android for mobile devices.*
* *Back-end Requirements: Data management necessitates the use of a database system such as, SQL, NoSQL, as well as cloud infrastructure for hosting and scaling.*

**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?*

* *Authentication Differentiation:* The system must properly distinguish between user roles using secure login procedures. To make things easier for the user, inputs should not be case sensitive.
* *Admin Alerts:* The system should alert administrators of serious errors discovered during system operations.

**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?*

* *User Management:* Changes to user roles and permissions should be managed using an administrative interface rather than directly modifying the code.
* *Platform Updates:* The system must be designed for easy adaptation to platform and technology updates.

**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?*

* *Login Requirements:* A non case-sensitive username and case-sensitive password is required to login to a user account. Multi-factor or Two-factor authentication should be used for user logins to improve security.
* *Data Security:* Encryption techniques must protect data flows between the client and the server. In the event of a brute force assault, the account should be temporarily frozen and the situation reported to administrators.
* *Password Recovery:* A safe procedure for password recovery or reset must be provided, with user verification measures in place.

**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 provide data access both offline and online, guaranteeing synchronization without duplication.
* The system shall ensure secure user registration and information management.
* The system shall have a complete user management interface that allows for role-based access control.
* The system shall enable tracking and reporting of user activity such as bookings and revisions.
* The system shall allow for flexible reservation and package customisation.

**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.)?*

* *Needs of Interface:* The interface shall be user-friendly and accessible to all sorts of users, including students, teachers, and administrators.
* *Users:* Different users would require different functions; for example, students should be able to handle reservations and access course material while administrators should have access to extensive management tools.
* *Interaction:* To guarantee accessibility, the interface shall be tailored for use with both mobile and desktop browsers.

**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?*

* *Constant Internet Access:* Users are assumed to have reliable internet connectivity in order to use online functionality.
* *Use Tech Literacy:* The design presume that users have a basic degree of technological skill.

**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?*

* *System Modification Dependency:* Some package changes may need developer participation to maintain system stability.
* *Online Functionality:* While offline, users may be limited to seeing data, with changes necessitating an internet connection to ensure data integrity.

**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.*

