**CS 255 Business Requirements Document Template**

Complete this template by replacing the bracketed text with the relevant information.

This template lays out all the different sections that you need to complete for Project One. Each section has guiding questions to prompt your thinking. These questions are meant to guide your initial responses to each area. You are encouraged to go beyond these questions using what you have learned in your readings. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead, the goal is to complete each section based on your client’s needs.

**Tip:** You should respond in a bulleted list for each section. This will make your thoughts easier to reference when you move into the design phase for Project Two. One starter bullet has been provided for you in each section, but you will need to add more.

**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 objective is to develop an enhanced system aimed at facilitating comprehensive training for students preparing for driving tests. Liam, the client, is keen on incorporating both online and in-person training modules. The online component will offer training materials and practice tests, while the in-person aspect will provide hands-on training assistance from the Driver Pass team.

**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?*

DriverPass aims for the system to be accessible both online and offline. The development team suggests leveraging cloud-based architecture for the interface to ensure seamless access. However, they express concerns about potential conflicts arising from saved changes when the system is used offline.

To address this, a conflict resolution algorithm could be implemented to reconcile offline changes with the cloud data when the system goes back online. This ensures data integrity across the platform.

Security is another critical consideration. The system should have role-based access control to ensure that only authorized employees can access sensitive information. For instance, training staff may have access to training modules and student progress, but only administrative staff should have access to private student information.

**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?*

DriverPass aims to offer a platform where consumers can select from a range of training packages. To manage demand effectively, the system should automatically disable the option to book packages that are already fully booked. This could be implemented through a real-time inventory management system integrated into the cloud-based interface.

Additionally, the platform should feature a dynamic dashboard for tracking student progress in online tests. The dashboard would display key metrics such as test name, time taken, score, and status, which could range from "Not Taken" to "Failed," "Passed," or "In Progress." This real-time tracking can be achieved by fetching data from the backend database and updating the user interface accordingly.

Lastly, the client wishes to have a feature where instructors can input notes and time metrics for each driving lesson. These should be viewable in an administrative dashboard, accessible only to authorized personnel. Security is a paramount concern, and role-based access control could be employed to restrict access to sensitive information.

By incorporating these functionalities, the system will not only fulfill the client's specific requirements but also align with best practices in system design and analysis.

**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?*

DriverPass requires a system that is updated frequently to address any bugs or security vulnerabilities. This aligns with the principles of DevOps and Continuous Integration/Continuous Deployment (CI/CD), which can be used to automate the testing and deployment of updates.

Additionally, the system must be agile enough to adapt to changes in DMV guidelines. Immediate updates are crucial to ensure that students are always working with the most current and accurate information. This could be managed through an admin panel that allows for the quick modification of training content and guidelines.

The system also needs to be web-based, which is in line with modern system architecture, offering both scalability and ease of access. Given that it will be cloud-hosted, considerations around data latency and speed are vital, especially for functionalities like real-time exam monitoring. Optimizing server response times and employing efficient algorithms can help in ensuring that the system runs at the desired speed.

**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?*

Firstly, the system should be browser-agnostic, meaning it should run seamlessly on various web browsers like Chrome, Microsoft Edge, and others. This ensures a wider reach and better user experience, aligning with the principles of progressive enhancement and graceful degradation in web development.

Secondly, the system should feature a responsive design to adapt to different screen sizes, particularly for mobile devices. This is crucial in today's mobile-first world and can be achieved using frameworks like Bootstrap or through custom CSS media queries.

Lastly, the backend will require a robust database to store various types of information, ranging from user profiles and test scores to package availability and instructor notes. Given the need for real-time updates and high-speed data retrieval, a relational database like PostgreSQL or a NoSQL database like MongoDB could be considered, depending on the specific data requirements.

Incorporating these technical specifications will not only meet the client's operational needs but also adhere to industry best practices in system design and analysis.

**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?*

The system will distinguish between different users primarily through their unique email and password combination. This is a standard practice in identity verification and ensures that each user has a unique identifier.

Input case sensitivity is crucial for enhancing security. It makes it more difficult for unauthorized users to guess passwords, thereby adding an extra layer of protection. This aligns with best practices in secure authentication.

The system should be configured to alert the admin if a user inputs incorrect information a certain number of times. This could be indicative of a security threat like a brute-force attack and should be immediately addressed. An alert mechanism can be implemented to notify the admin through email or other means.

**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?*

The system should allow for the addition, removal, and modification of users without requiring changes to the codebase. This can be achieved through a well-designed admin panel that interacts with the backend via POST requests and controllers. CRUD (Create, Read, Update, Delete) operations can be implemented to manage user data effectively.

The system should be designed to easily adapt to platform updates. This can be facilitated by setting up endpoints that can receive requests from programmers to update or modify functionalities. An API-first approach can be beneficial here, allowing for easy integration and updates.

IT admins should have comprehensive access to the system, including the ability to manage user accounts, passwords, and employee availability. This aligns with the principle of role-based access control, where different roles have different levels of access to the system's functionalities.

**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?*

Both students and administrators will be required to log in using their username and password. This is a standard practice for secure authentication.

HTTPS (not HTTP) should be used to ensure that the data exchanged between the client and the server is encrypted and secure. This is crucial for protecting sensitive user information and aligns with industry best practices.

In the event of a brute force hacking attempt, the system should alert the administrator after a pre-defined number of consecutive failed login attempts, which can be set between 1-10. After four failed attempts, the system should lock the user out from making further attempts and alert the admin. This dual-action approach enhances security.

If a user forgets their password, they can initiate a password reset request. The system should then send a password reset link to the user's registered email address, allowing them to create a new password securely

**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 validate user credentials when logging in.

The system shall confirm consumer selection from the three packages the client wishes to include.

The system shall capture and confirm consumer details, such as customer first and last name, address, credit card information, phone number, and state.

The system shall operate primarily online but make certain materials, like study material, available for offline use.

The system shall identify the type of user logging in, whether it's a student or an administrator.

The system shall display the three types of packages that the client has suggested.

The system shall allow the client to disable packages if they become unavailable.

The system shall indicate disabled packages to consumers.

The system shall provide a password reset function for users.

The system shall display real-time exam progress and scores for students.

The system shall verify login information to grant users access to their accounts.

The system shall update promptly to reflect any changes in DMV guidelines.

**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.)?*

Interface Needs:

* Responsiveness: The interface should be responsive to adapt to various screen sizes, including mobile devices and desktop browsers.
* Intuitiveness: The UI should be intuitive, requiring minimal learning curve for users.
* Security: Secure login panels for both regular users and administrators.
* Real-time Updates: The interface should update in real-time to reflect changes in exam status, grades, and other dynamic content.
* Accessibility: The interface should be accessible, following best practices like ARIA (Accessible Rich Internet Applications) standards.

Regular Users (Students):

* Home Page: Overview of available packages, study materials, and other resources.
* Exam: Access to online exams and practice tests.
* Access to Grades: View scores and feedback on completed exams.
* User Information: Update passwords, address, and other personal details.
* Registration: Sign-up functionality for new users.
* Exam Status: Real-time updates on the status of ongoing or upcoming exams.
* Instructor Notes: Access to any feedback or notes left by instructors.
* Contact Information: Details for customer support or instructor contact.

Administrators:

* User Management: Ability to add, delete, or modify user accounts.
* Package Management: Enable or disable training packages.
* Exam Monitoring: Real-time view of ongoing exams and the ability to intervene if necessary.
* Content Updates: Update study materials and exam questions to align with DMV changes.

User Interaction:

* Regular Users: Primarily through mobile devices for on-the-go access but also through desktop browsers for more in-depth tasks like taking exams.
* Administrators: Primarily through desktop browsers for better management capabilities.

**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?*

* Data Backup and Recovery: The design doesn't specify how data will be backed up and recovered in case of system failures.
* Internationalization: The design doesn't account for users who might not be English speakers or who are accessing the system from different time zones.
* Scalability: While the system is web-based, the design doesn't explicitly mention how it will handle increased user loads.
* User Training and Onboarding: The design does not include how new users, especially administrators, will be trained to use the system.
* Audit Trails: The design doesn't specify if there will be audit trails for tracking changes, especially by administrators.

Assumptions:

* 24/7 Availability: It's assumed that the system will be available around the clock, requiring robust server infrastructure.
* User Base: The assumption is that the primary users will be DMV students, which influences the type of content and exams offered.
* Up-to-Date Information: It's assumed that the system will always be updated to reflect the latest DMV guidelines.
* Future App Version: The design assumes that there will be a mobile app version of DriverPass in the future, which would require additional development efforts.

**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?*

Limitations:

Internet Dependency:

* The system's functionality is heavily reliant on a stable internet connection. Without it, users won't be able to update data or access the database. This could be a significant limitation for users in areas with poor internet connectivity.

Time and Budget Constraints:

* The project is bound by the time frame and budget set by the client. This could limit the scope of features that can be implemented and may necessitate prioritizing certain functionalities over others.

DMV Guidelines:

* The content and exams are based strictly on DMV guidelines. While this ensures accuracy, it also limits the system's flexibility to offer additional or alternative training materials.

Limited Physical Resources:

* The client has only 10 cars available for training, which necessitates precise scheduling to ensure availability and avoid conflicts.

**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.*

