# 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?*

**Purpose**

* **Client Overview:** The client, DriverPass, is a new company aimed at enhancing driver training effectiveness. The company was founded in response to the high failure rates observed in driving tests at the DMV, attributed to insufficient practical training and over-reliance on past tests.
* **System Objective:** DriverPass seeks to develop a system that facilitates both theoretical and practical learning through online practice exams and on-the-road training. This system is intended to provide a comprehensive learning platform that better prepares students for their driving tests.

**Primary Functions**:

* To allow students to access and take online practice exams.
* To enable scheduling and management of on-the-road driving lessons.
* To provide a user-friendly interface where students can manage their learning and training schedules.
* To offer administrative capabilities for DriverPass management to monitor and report on student progress and system usage.

### 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?***

* **Problem Identification:** DriverPass identified a what they believe is a gap in the market for comprehensive driver training. Most students fail their driving tests because their preparation is based solely on previous tests without practical experience. DriverPass aims to address this issue by offering a blend of theoretical knowledge through online practice exams and practical experience through on-the-road training.

**System Requirements**:

* **Online Practice Exams**: An online platform where students can take simulated driving tests that mimic the structure and content of actual DMV exams. This component requires a robust database of questions, a user interface for test-taking, and analytics to track progress and performance.
* **On-the-Road Training Scheduling**: A scheduling system that allows students to book, reschedule, or cancel driving lessons. This system should integrate with calendars for students and instructors to manage appointments efficiently.
* **Data Management and Reporting**: To ensure the effectiveness of the training, the system needs to collect and analyze data on student performance, both in online tests and during practical driving lessons. This would involve data input fields, storage, and report generation capabilities.
* **User Authentication and Security**: Different levels of access for various users (students, instructors, administrators) to protect sensitive information and ensure that users can only access data pertinent to their role.

A few things what they want to accomplish:

* Reduce the failure rate at DMV driving tests by providing comprehensive, accessible, and user-friendly training tools.
* Enhance the learning experience by integrating theoretical and practical training methods.

* Offer DriverPass the capability to monitor and improve their service based on detailed analytics and user feedback.
* Address the current inadequacies in driver training while at the same time also setting a new standard for educational tools in this domain.

### 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?***

* **Overall System Objective:** To develop an integrated training platform that significantly reduces the failure rates of driving tests by providing students with both theoretical knowledge and practical driving experience.

**Measurable Objectives:**

1. **Enhance Study Materials Accessibility**:
   * Provide online access to comprehensive practice exams that cover the full scope of DMV requirements.
   * Include at least 500 different practice questions by launch, with plans to expand the question bank over time.
2. **Improve Practical Training**:
   * Enable students to book a minimum of six driving sessions per package, with additional options for more intensive training.
   * Offer flexibility in scheduling, allowing students to select times and dates that best fit their availability.

**Track Progress and Adapt Learning**:

* Implement a tracking system that records student scores and progress on practice exams and during driving lessons.
* Provide feedback mechanisms that allow students to review their performance and identify areas needing improvement.

**Ensure System Accessibility and Usability**:

* Design a user-friendly interface that is accessible on multiple devices, including desktops, tablets, and smartphones.
* Ensure the system is intuitive enough for users with basic computer skills to navigate and use effectively.

 **Maintain High Security Standards**:

* Implement robust security measures to protect personal and payment information.
* Set up different user roles and permissions to ensure that data access is appropriately managed.

 **Integrate with External Systems**:

* Allow for integration with DMV systems to receive updates on driving test requirements and include them in the practice materials.
* Ensure the system can export data in formats compatible with common office software for offline work by administrators.

**Short-term Goals**:

* Successfully launch the system with full functionality for online tests and scheduling within six months.
* Achieve a user base of 1,000 students within the first year.

**Long-term Goals**:

* Expand the question bank and training modules based on user feedback and DMV updates.
* Increase the pass rate of students using the system by at least 40% compared to those who do not use the system.
* Develop partnerships with driving schools to offer integrated services.

## 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?***

**Performance Requirements**

* **Operating Environments:** The DriverPass system is primarily web-based, requiring compatibility across various browsers (Chrome, Firefox, Safari, and Edge) and platforms (Windows, macOS, iOS, Android). This ensures that users can access the system from any computer or mobile device with internet connectivity.
* **Response Time:** The system should have a response time of less than two seconds for all user interactions under normal load conditions. This includes loading pages, submitting forms, and retrieving data from the database.
* **System Updates:** The software should receive updates at least once every three months to ensure security, introduce new features, and refine existing functionalities. Emergency patches should be deployable within 24 hours in response to any critical security vulnerabilities or major bugs that disrupt user experience.
* **Scalability:** The system must be scalable to accommodate increases in user numbers and data volume without degradation of performance. It should support up to 10,000 concurrent users and be capable of expanding as the customer base grows.
* **Reliability:** The system should aim for a 99.9% uptime, excluding scheduled maintenance periods. Backup servers and redundancy measures must be in place to ensure continuous service availability during unforeseen disruptions.
* **Maintenance:** Scheduled maintenance should occur during off-peak hours and be communicated to users in advance to minimize disruption. The system should include tools for monitoring performance and automating routine maintenance tasks to reduce downtime.

#### 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?***

**Supported Platforms**: The DriverPass system should be platform-agnostic on the client side, functioning seamlessly on Windows, macOS, Linux, iOS, and Android. This broad compatibility ensures that users can access the system from any personal or mobile device. This is very important as it lets everyone with any device access, making “portable” to say in laymens terms.

**Backend Infrastructure**:

* **Database**: The system requires a robust relational database management system (RDBMS) such as PostgreSQL or MySQL to store user data, practice test results, appointment schedules, and other operational data. This database should support complex queries and transactions needed for scheduling and tracking. Also need to include some type of masking and tokenization for storing user payments, like credit cards, debit card information for example.
* **Server**: Backend services should be capable of running on both Windows Server and Unix/Linux systems to ensure flexibility in hosting and scalability. The use of containerization technologies like Docker can be employed to maintain consistency across different environments.
* • Cloud Services: Considering the system's need for high availability and scalability, utilizing cloud platforms such as AWS, Google Cloud, or Microsoft Azure for hosting the application and database is recommended. These services provide necessary scalability and reliability features, such as load balancing and auto-scaling. Plus most of the platform have security with them so it would less we have to in that area(we still need security but maybe not as much on certain containers if hosted outside)

• Development Environment:

* **Web Frameworks:** The system should be developed using robust web development frameworks like React or Angular for the frontend, and Node.js or .NET Core for the backend. These frameworks support cross-platform development and can handle the system’s requirements for interactivity and real-time data processing.
* APIs: RESTful API architecture should be used for the backend to facilitate communication between the frontend and the database, ensuring smooth data flow and integration.

#### 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?*

* **User Differentiation:** Each user will have a unique identifier, such as a username or email address, coupled with a secure password for login purposes. The system will also utilize role-based access control (RBAC) to assign specific permissions and functionalities based on the user's role (e.g., admin, instructor, student) within the organization. This ensures that each user accesses only the appropriate data and functions relevant to their responsibilities.
* **Case Sensitivity:** Usernames should be case-insensitive to avoid confusion and login errors; however, passwords will remain case-sensitive to ensure robust security. The input for other user-specific data, like email addresses, should also be case-insensitive.

**Error Reporting and Notifications**:

* The system should automatically notify administrators of any critical errors or security breaches immediately via automated email alerts or SMS.
* For less critical system issues, such as failed login attempts or data entry errors, the system should log the incident and generate a daily report for review by the IT staff.
* Admins should receive notifications for system performance issues if system response times exceed the threshold of five seconds or if there are connectivity problems affecting user access.

#### 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 Flexibility**: The system will include an administrative interface that allows designated admin users to add, remove, or modify user accounts without any need for direct code changes. This interface will provide tools for managing user roles, permissions, and details, enabling dynamic user management as the organization evolves.

**Adapting to Platform Updates**:

* The system's architecture should be designed with modularity in mind, allowing individual components, such as the user interface or database integration, to be updated independently in response to platform updates or new technological advancements.
* Regular updates and compatibility checks will be scheduled to ensure that all components of the system remain functional and secure across all supported platforms. The use of containerization and microservices architecture can help isolate and manage these updates efficiently.

**IT Admin Access Requirements**:

* IT administrators will need comprehensive access to all system components, including backend databases, server management tools, and administrative interfaces.
* Admins should have the ability to oversee and control security settings, manage user roles and permissions, monitor system health, and initiate backups or recovery procedures.
* The system will provide IT admins with tools for real-time monitoring and alerts, enabling them to address potential issues proactively.

#### 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**:

* Users are required to log in using a unique username or email address and a strong password that complies with security standards (minimum length, inclusion of uppercase and lowercase letters, numbers, and special characters).
* For added security, implementing multi-factor authentication (MFA) is recommended, which could include a code sent to the user's registered mobile device or email, or the use of a third-party authenticator app.

**Securing Data Exchange**:

* All communications between the client and the server should be encrypted using TLS (Transport Layer Security) to ensure that data transmitted over the internet is secure and private.
* Sensitive data, such as passwords and personal information, should be stored in the database using strong, industry-standard cryptographic hash functions like SHA-256, with salts to prevent rainbow table attacks.

**Brute Force Attack Mitigation**:

* The system should implement account lockout policies after a predetermined number of unsuccessful login attempts (e.g., 5 attempts). Locked accounts should remain inaccessible for a specified cooling-off period (e.g., 30 minutes) or until an administrator manually unlocks the account.
* The system should also implement CAPTCHA challenges after a few failed login attempts to deter automated login attempts and reduce the risk of brute force attacks.

**Password Reset Procedures**:

* If a user forgets their password, they should be able to initiate a password reset process via a "Forgot Password?" or reset link something along these lines with a link on the login page.
* This process should verify the user's identity by sending a password reset link to the registered email address or a code to the registered mobile number.
* The user should be prompted to answer security questions, if previously set up, as an additional verification step before resetting the password.

### 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.”***

**User Management and Access Control**:

* The system should validate user credentials when logging in.
* The system should allow administrators to add, modify, and delete user accounts.
* The system should provide role-based access control to different features and data based on the user's role (e.g., student, instructor, administrator).

**Online Practice Exams**:

* The system should provide a library of practice exam questions that students can access online.
* The system should allow students to take practice exams and receive scores immediately upon completion.
* The system should save and track the results of each student's practice exams to monitor progress over time.

 **Scheduling System**:

* The system should allow students to schedule, reschedule, and cancel driving lessons through an online interface.
* The system should display available times and dates for driving instructors and vehicles.
* The system should send automatic reminders to students and instructors 24 hours before scheduled lessons.

 **Data and Reporting**:

* The system should generate reports on student performance and progress for both practice exams and driving lessons.
* The system should allow administrators to access and download reports in various formats (e.g., PDF, Excel).

**Security and Data Integrity**:

* The system should encrypt all data transmitted between the client and the server.
* The system should perform regular backups of all system data to prevent data loss.
* The system should log all user activities within the system to provide an audit trail in case of security breaches or disputes.

 **Compliance and Updates**:

* The system should automatically update its question bank and content as per the latest DMV guidelines and requirements.
* The system should notify administrators of any changes in DMV regulations that might affect the content or functionality of the system.

 **Interface and Usability**:

* The system should provide a user-friendly interface that is navigable by users with varying levels of technical skill.
* The system shouldl be accessible on multiple devices including desktops, tablets, and smartphones.

 **Support and Maintenance**:

* The system shouldl provide an online help and support section with FAQs and contact information for technical support.
* The system should offer tools for administrators to perform routine maintenance and troubleshooting without needing developer intervention.

### 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**:

* The system should provide a clear and intuitive user interface that minimizes the learning curve and enhances user engagement.
* The interface should be responsive, ensuring usability across different devices and screen sizes, including desktops, tablets, and smartphones.
* The interface should incorporate accessibility features to accommodate users with disabilities, including screen reader support, high contrast modes, and keyboard navigation.
* We may need to access an externa API or database, something for example like DMV records possibly.

 **Different Users of the Interface**:

* **Students**: Primary users who will access online practice exams, schedule driving lessons, and review their progress and scores.
* **Instructors**: Will need to access their schedules, student progress reports, and any changes to their daily lessons.
* **Administrators**: Need to manage user accounts, access comprehensive reports on user activity and system performance, and update course content.
* **IT Support Staff**: Require tools to monitor system health, manage security settings, and perform system updates and maintenance.

 **Functionalities for Each User**:

* **Students**:
  + The system should allow students to log in securely, take practice exams, view past scores, and track progress.
  + Students should be able to schedule, reschedule, or cancel driving lessons and receive automatic reminders.
  + The interface should provide a dashboard summarizing their achievements and areas for improvement.
* **Instructors**:
  + Instructors should be able to view and manage their daily schedules through the interface.
  + The system should enable instructors to input notes and feedback on student performance during lessons.
  + Instructors should have access to necessary student information to tailor their teaching strategies.
* **Administrators**:
  + Administrators should be able to add, remove, or modify the details of users and roles.
  + The system should provide detailed reports on student performance, system usage, and financial metrics.
  + Administrators should be able to update practice exam questions and manage content as per DMV changes.
* **IT Support Staff**:
  + IT staff should have tools to monitor system performance, address security concerns, and manage backups.
  + The system should provide functionalities for routine maintenance and troubleshooting.

**Interaction with the Interface**:

* The system should be accessible via web browsers to ensure it is available on a wide range of devices without requiring any downloads or installations.
* For mobile users, the interface should be optimized for touch interactions, with a layout and navigation suited to smaller screens.

### 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?***

**Technological Proficiency**: We are assuming that all users of the DriverPass system, including students, instructors, and administrators, have a basic level of technological proficiency. This includes the ability to navigate web interfaces, use email, and interact with online systems. This assumption influences the design to not overly simplify the interface, maintaining a balance between functionality and user-friendliness.

**Internet Connectivity**: Our design assumes that users have consistent access to reliable internet connectivity. This is essential as the system is web-based and requires an internet connection for accessing all features, such as taking online exams, scheduling appointments, and accessing training materials.

**Device Availability**: We assume that users have access to devices capable of running a modern web browser, such as smartphones, tablets, or computers. This assumption is critical for ensuring that users can interact with the system as intended, without the need for specialized hardware.

**Security Compliance**: It is assumed that the users will adhere to recommended security practices, such as securing their login credentials and not sharing their accounts. This assumption is important for maintaining the security integrity of the system. We need to make sure that the cloud security from whatever provider is up to par also

**Regulatory Compliance**: We assume that the current regulations and requirements provided by the DMV will remain stable in the near term. This impacts the system’s content updates and compliance features, aiming to minimize frequent overhauls or changes due to regulatory shifts.

**Maintenance and Support**: We are assuming that DriverPass will have the necessary IT support to handle routine maintenance, troubleshooting, and updates to the system. This includes having staff who are trained and capable of managing a web-based platform.

### 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?*

* **Resource Constraints**: Given the scope of the DriverPass system, the design and implementation may be limited by the available budget and human resources. The ability to recruit skilled developers, designers, and system analysts within the budget constraints could affect the quality and timeliness of the system development.
* **Time Constraints**: The project timeline is aggressive, aiming to launch the system within a few months. This limitation may result in prioritizing certain features over others, potentially leading to a phased roll-out rather than a full feature release at launch. Especially, if there are any changes, errors or scheduling conflict so planning should be of the utmost importance here.
* **Technology Adoption**: The system’s reliance on modern web technologies and cloud-based services might limit accessibility for users in regions with poor internet connectivity or outdated technology. This could restrict the system’s reach and effectiveness in certain demographic or geographic areas. That being said, regions with no internet access would probably not need to utilize this type of tech so it could be put on the bottom of the to do list.
* **Scalability Challenges**: While the system is designed to be scalable, rapid growth in the user base or data volume could strain the current infrastructure. Scaling up infrastructure to meet increasing demands may require additional investments and could lead to performance bottlenecks if not managed proactively.
* **Integration with External Systems**: The system’s functionality depends partially on integration with external systems, such as DMV databases for regulatory updates. Any changes or disruptions in these external services could impact the functionality and reliability of the DriverPass system. So there may need to be some research done to see if a DMV database has an API or something like this and if there will be any additional costs for this type of stuff.
* **Security Risks**: Despite implementing robust security measures, the system is susceptible to evolving cybersecurity threats. The ongoing need to update and reinforce security measures could impose additional costs and require continuous monitoring and adjustments.
* **User Diversity**: The system design assumes a certain level of uniformity in user behavior and technological proficiency. Variations in user needs and capabilities, especially among users with disabilities or those unfamiliar with digital interfaces, could affect the usability and accessibility of the system.

### Gantt Chart

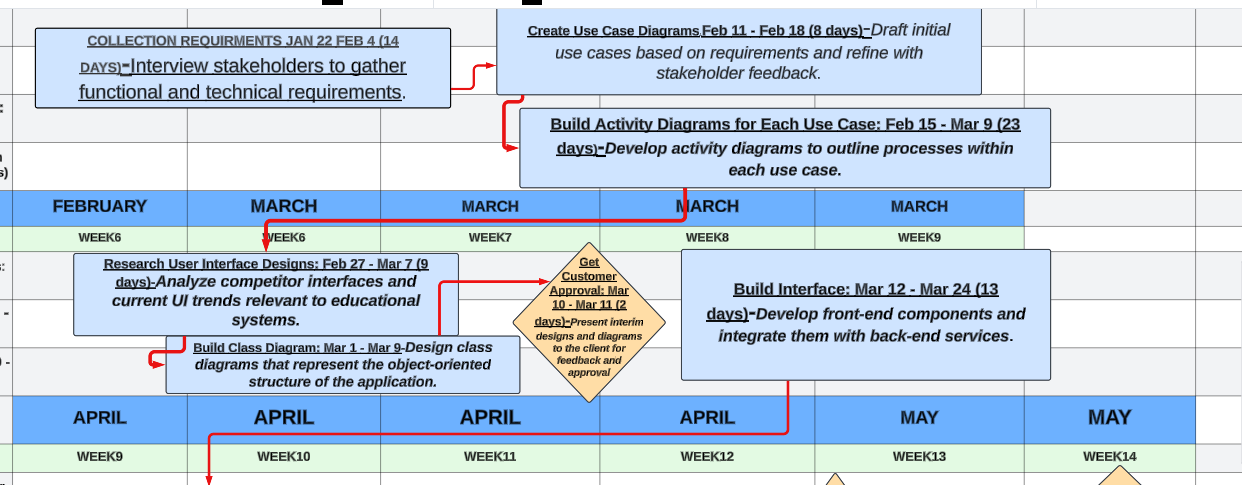
*Please include a screenshot of the GANTT chart that you created with Lucidchart. Be sure to check that it meets the plan described by the characters in the interview.*

**Below is entire screeshot of Gannt Chart….below that I have zoomed in on the top, bottom and side.**

A screenshot of a calendar

Description automatically generated

**TOP**



**BOTTOM**

A screenshot of a calendar

Description automatically generated

**SIDE**

