# 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 purpose of the DriverPass project is to create a system that provides comprehensive training services for customers preparing for their driving tests at the Department of Motor Vehicles (DMV). The client, DriverPass, represented by its owner Liam and IT officer Ian, wants the system to help improve the pass rates of driving tests by offering a mix of online and in-person training options. The system is intended to fill a gap in the market for better driving training solutions.

The client wants the system to be able to:

* Offer online classes and practice tests for students to prepare for their driving tests.
* Provide on-the-road training that customers can schedule through the system.
* Allow customers to make, modify, and cancel driving lesson reservations online, as well as through the office (via phone or in-person with the secretary).
* Assign customers to specific drivers and cars based on their reservations.
* Handle flexible training packages that can be enabled or disabled by the owner.
* Track user actions (such as who made or modified reservations) and generate activity reports for accountability.
* Allow role-based access, where different users (owner, IT officer, secretary) have different levels of system access.
* Provide mobile and remote access to system data from any device.
* Ensure that the system stays up to date with DMV regulations and changes.
* Ensure security, including managing user accounts, tracking changes, and resetting passwords.

### 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 wants the system to provide a comprehensive and accessible platform for customers to prepare for their driving tests. The system should offer a combination of online learning, practice tests, and on-the-road training sessions. The goal is to help customers improve their pass rates on DMV driving exams by providing better training resources and easier access to these services. Specifically, DriverPass wants the system to:

* **Provide Online Training and Practice Tests**: Customers should be able to take online driving courses and complete practice tests to prepare for the DMV exam.
* **Offer Scheduling for On-the-Road Lessons**: Customers should be able to schedule, modify, and cancel driving lessons online or through the office, and the system should manage reservations for driving sessions with trainers and cars.
* **Track and Manage User Activities**: The system should log and track who makes or modifies appointments and provide reporting features to identify any issues related to bookings.
* **Support Flexible Training Packages**: DriverPass wants to offer different training packages, each with a different combination of driving hours and additional resources, such as in-person lessons or online materials. The system should be flexible enough to allow the owner to disable or enable packages as needed.
* **Provide Role-Based Access**: The system should allow different types of users (e.g., owner, IT officer, secretary, customer) to access and interact with the system at different permission levels.
* **Ensure Security and Data Integrity**: The system needs to handle customer data securely, allow password management, and maintain role-based access control for various user roles.
* **Ensure Compliance with DMV Regulations**: The system must stay up to date with the latest DMV requirements, ensuring that the provided training and tests match the current standards.
  1. The components needed for this to be done are:
* **Online Learning Module**: To provide the content for driving lessons and practice tests, including tracking customer progress through the tests.
* **Appointment Scheduling Module**: For customers to book, modify, and cancel on-the-road driving lessons with specific trainers and cars.
* **User Management and Authentication**: Role-based access control for different user types (owner, IT officer, secretary, and customer) and a system for managing passwords, account recovery, and user-specific settings.
* **Activity Tracking and Reporting**: A system to log all user activities (e.g., who made, canceled, or modified an appointment) and generate reports for management oversight.
* **Training Packages Management**: A component to handle the configuration of training packages, including enabling and disabling packages.
* **DMV Compliance Integration**: A system to ensure that the training materials and tests are kept up to date with DMV requirements.
* **Cloud-Based Access and Security**: The system needs to be cloud-based to ensure data accessibility from any device and provide secure backup and data protection measures.

### 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?*

When the DriverPass system is completed, it should be able to provide comprehensive services to both customers and the company’s internal users (e.g., the owner, IT officer, and secretary). The system should streamline the process of preparing for a driving test through a combination of online training, on-the-road driving lessons, and practice tests. It should also handle appointment scheduling, user management, and compliance with DMV regulations. Below is a breakdown of the key functionalities and measurable tasks that need to be included in the system design to achieve these goals.

## 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 operate in a web-based environment with cloud support, ensuring that both customers and employees can access the system remotely from various devices, including computers, smartphones, and tablets. The system must be designed to function responsively, adjusting seamlessly across different screen sizes to provide a smooth user experience. Additionally, the system should be hosted on the cloud, allowing for real-time data synchronization and scalability as the business grows. This also ensures that DriverPass does not need to manage data backups or disaster recovery internally. While a dedicated mobile app could be a future addition, the initial system should be optimized for mobile use through responsive web design, allowing customers to book appointments or access lessons on-the-go.

In terms of performance, the system should have a fast response time, loading pages and processing actions like scheduling lessons or accessing practice tests within 2-3 seconds. For data-heavy operations, such as retrieving customer records or checking appointment availability, the system should process these requests within 5 seconds to avoid frustrating users. Real-time data synchronization is essential, ensuring that all users—whether customers or staff—see updates immediately when changes are made, such as when a lesson is booked or canceled. The system should also be scalable to handle multiple concurrent users without experiencing performance degradation.

To maintain system functionality and security, updates should occur regularly. Security updates must be applied as soon as vulnerabilities are identified, given the sensitive nature of the customer data being handled. Feature updates, such as enhancements to the user interface or the addition of new functionalities, should be rolled out on a quarterly basis, ensuring the system evolves with customer needs and regulatory changes. Regulatory updates will be particularly important for keeping the system compliant with DMV requirements, and these should be applied immediately when new rules or test materials are provided. Additionally, routine maintenance tasks, such as database optimization and internal code updates, should be conducted on a monthly basis to keep the system running efficiently. By following this schedule, DriverPass can ensure its system remains secure, up-to-date, and responsive to both business and customer demands.

#### 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 DriverPass system should be designed to run on a cross-platform environment to maximize accessibility and flexibility. Since the system is primarily web-based, it needs to be compatible with major operating systems like Windows, macOS, and Linux/Unix, as well as mobile operating systems such as iOS and Android. This ensures that users—whether customers accessing the system via their computers or mobile devices, or employees managing appointments and lessons—can interact with the system regardless of their preferred platform. Additionally, because the system is cloud-based, the operating system on which the system runs locally is less important if the web interface can be accessed through a standard web browser like Google Chrome, Firefox, Safari, or Microsoft Edge.

**Backend Requirements and Tools:**

For the back end, the system requires several key tools to support its operations:

1. **Database**: A robust and scalable database is critical for managing user accounts, appointments, customer progress, training packages, and other data. The system should use a relational database management system (RDBMS) such as MySQL, PostgreSQL, or Microsoft SQL Server. These databases are highly reliable for storing structured data, ensuring data integrity, and handling transactions efficiently. A relational database would allow for easy querying, data management, and security, especially given the need for real-time access and updates across various users.
2. **Cloud Infrastructure**: Since the system will be cloud-based, a platform like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud is required to host both the application and the database. These cloud platforms provide infrastructure scalability, security, and the ability to handle large numbers of concurrent users while offering built-in tools for backups, disaster recovery, and data encryption.
3. **Web Server**: The system will need a web server, such as Apache or Nginx, to handle requests from users, serve web pages, and manage communication between the front-end interface and the back-end database. These servers are essential for managing HTTP requests and ensuring fast, reliable connections between the user and the application.
4. **API Integration**: The system will also need APIs to integrate with external services, especially for DMV compliance updates. This could include connecting to DMV databases to automatically retrieve updates related to driving rules or tests, ensuring that the training materials remain current.
5. **Authentication and Security Tools**: Given the sensitive nature of customer data (e.g., personal information, payment details), the back end should use security tools and libraries that support authentication, encryption, and role-based access control. This can be implemented using tools like OAuth, JWT (JSON Web Tokens), and SSL/TLS encryption to protect data during transmission.

#### 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 DriverPass system will distinguish between different users using role-based access control (RBAC), assigning varying levels of access and permissions based on the user's role within the system. The primary roles include the administrator (the owner), who has full access to all system features, including user management, reports, and package modifications. The IT officer will have access like the administrator but with a focus on system maintenance and troubleshooting, such as managing user accounts, resetting passwords, and monitoring security logs. The secretary will have limited access, mainly to manage customer information and appointments, but without the ability to view sensitive data or change system settings. Finally, customers will have the most restricted access, allowing them to schedule lessons, view their progress, and take practice tests, but not access administrative functions.

In terms of input sensitivity, critical areas like passwords and usernames will be case-sensitive to maintain system security and ensure that user credentials are unique and properly verified. This is particularly important for login authentication. However, inputs such as customer names, addresses, or other general information will not need to be case-sensitive to make data entry more user-friendly.

The system will also be equipped to notify the admin (owner or IT officer) whenever a problem arises that could affect functionality or security. For example, after three failed login attempts, the system will alert the admin to possible unauthorized access attempts. Any signs of data breaches, unauthorized access, or security threats, such as repeated attempts to access restricted areas or unusual login behavior, will trigger immediate notifications to the admin. Additionally, the system will inform the admin of performance issues, such as system downtime or slow response times, and any database errors that could disrupt operations. Payment failures will also generate alerts, ensuring that issues with financial transactions can be addressed promptly. Finally, the admin will be informed when system updates or maintenance are needed, ensuring that the platform stays secure and up to date. By providing these safeguards and notifications, the DriverPass system ensures smooth and secure operations across all user levels.

#### 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 DriverPass system should be designed with flexibility in mind, allowing for user management (add/remove/modify) without needing to change the underlying code. This can be achieved through an administrative interface that provides authorized users (such as the IT admin or system owner) the ability to manage users directly through the system. By implementing a role-based access control (RBAC) mechanism, the admin can easily assign roles, update user details, or deactivate accounts without the need for code changes. This approach ensures that user management is handled dynamically through the interface, making the system more efficient and user-friendly.

As for platform updates, the system must be adaptable to updates in its operating environment, such as changes in the underlying cloud platform, web servers, or database management systems. Since the system will be hosted in a cloud-based infrastructure, it can leverage built-in platform services such as automatic updates and maintenance from cloud providers like AWS, Google Cloud, or Microsoft Azure. These platforms handle much of the lower-level updates (e.g., security patches, server optimizations) without requiring manual intervention. Additionally, the system itself should be developed with modular architecture and version control in mind, ensuring that updates to one part of the system (such as adding new features or updating security protocols) do not disrupt other components. Regular updates should be part of the system maintenance plan to ensure compatibility with platform changes and keep the system secure and functional.

The IT admin will require comprehensive access to manage and maintain the system efficiently. They will need full administrative privileges to manage users, such as creating new user accounts, resetting passwords, modifying user roles, and deactivating accounts. The IT admin will also need access to system logs for monitoring user activity, tracking changes to the system, and diagnosing any errors or issues. This includes monitoring security logs for unauthorized access attempts and reviewing activity logs for user actions like creating or modifying appointments. Additionally, the IT admin will need access to system settings to manage backups, data recovery, and security protocols (such as configuring firewalls, encryption, and user authentication). Finally, the IT admin will have to ensure that updates and patches are applied promptly to maintain the system’s security and performance, including updates to the cloud environment, web servers, or database configurations.

#### 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?*

For a user to log into the DriverPass system, they will need to provide their username (or email address) and password. These credentials will be matched against the records stored in the system’s database to authenticate the user. The login process should include additional security measures, such as password encryption during storage (using hashing techniques like bcrypt or SHA-256) and transmission (using SSL/TLS encryption). This ensures that user credentials are protected both while stored in the database and while transmitted between the client and the server.

**Securing the Connection and Data Exchange:**

To secure the connection and protect data exchanged between the client (user) and the server, the system should implement SSL/TLS encryption. This creates a secure communication channel that prevents unauthorized parties from intercepting or tampering with data during transmission. The use of HTTPS (enabled by SSL/TLS certificates) ensures that all communication, including login details and other sensitive data, is encrypted. Additionally, multi-factor authentication (MFA) could be introduced as an extra layer of security, requiring users to provide a secondary form of verification, such as a code sent to their mobile device or email, before they can successfully log in.

**Handling Brute Force Hacking Attempts:**

In the event of a brute force hacking attempt—where an attacker tries to guess a user’s password by trying numerous combinations—the system should have mechanisms in place to detect and prevent these attacks. After a set number of failed login attempts (commonly 3-5 attempts), the system should:

1. **Lock the account temporarily** (e.g., for 15 minutes) to prevent further login attempts.
2. **Notify the user** via email that multiple failed login attempts were detected, suggesting they take steps to secure their account.
3. **Alert the administrator** of the suspicious activity so they can monitor the situation and take action if necessary.
4. Optionally, the system could introduce CAPTCHA after a certain number of failed attempts to further prevent automated brute force attacks.

This strategy prevents attackers from continuously trying passwords without triggering a response and protects user accounts from being compromised.

**Handling Forgotten Passwords:**

If a user forgets their password, the system should provide a password recovery process that is both secure and user-friendly. The user will be able to click a "Forgot Password" link on the login page. This will initiate a process where:

1. The system will ask the user to provide their registered email address.
2. A password reset link, or a temporary code will be sent to the user’s email address. This reset link will typically expire after a short period (e.g., 24 hours) to ensure security.
3. Upon clicking the link or entering the temporary code, the user will be prompted to create a new password. The system should enforce strong password requirements (e.g., minimum length, inclusion of letters, numbers, and symbols) to ensure security.
4. Once the new password is set, the user can log in using the updated credentials.

This process ensures that password resets are secure and that unauthorized users cannot easily gain access to accounts by abusing the password recovery process.

### 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, ensuring the entered username/email and password match the stored records.
* The system shall allow users to register for a new account, including the ability to input necessary information such as name, address, phone number, and email.
* The system shall allow customers to book, modify, and cancel driving lesson appointments through an online interface or via the office through the secretary.
* The system shall manage driving lessons by matching customers with available drivers, cars, and time slots based on customer preferences and availability.
* The system shall enable administrators to manage user roles and permissions, allowing the addition, modification, and deactivation of accounts without changing code.
* The system shall allow users to reset their password by sending a reset link or code to their registered email address.
* The system shall log user actions (e.g., booking, modifying, or canceling lessons) and make these logs available to administrators for review.
* The system shall provide real-time data synchronization, ensuring that changes made by one user are immediately visible to other users (e.g., appointment availability).
* The system shall allow administrators to enable or disable training packages, based on business needs, through a management interface.
* The system shall send email notifications to users for significant account activities, such as successful bookings, canceled lessons, or failed login attempts.
* The system shall generate activity reports for administrators, detailing actions taken by users, including who made reservations, cancellations, or modifications.
* The system shall support secure data transmission using SSL/TLS encryption to protect user data during exchanges between the client and the server.
* The system shall enforce password security policies, requiring users to create strong passwords that include a combination of letters, numbers, and symbols.
* The system shall notify administrators of suspicious activity, such as multiple failed login attempts, and lock user accounts after a defined number of failed attempts.
* The system shall update training materials and practice tests based on changes from the DMV, ensuring that the content remains current with regulatory standards.
* The system shall allow customers to track their progress in online lessons and practice tests, displaying completed lessons, scores, and practice test statuses.
* The system shall provide mobile-friendly access, allowing users to access their accounts and make reservations from smartphones or tablets.
* The system shall notify administrators when system updates or maintenance are required, ensuring the platform remains up-to-date and secure.

These functional requirements outline the critical capabilities that the DriverPass system must deliver to meet both user and business needs.

### 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.)?*

The DriverPass system interface must be intuitive, user-friendly, and responsive to cater to different types of users, including customers, administrators, IT officers, and secretaries. The interface needs to ensure seamless navigation, quick access to core functionalities, and the ability to handle multiple roles and tasks. It should be designed with a focus on simplicity for customers and more detailed management tools for administrators and staff. Additionally, the interface must be mobile-responsive, allowing users to access the system via smartphones and tablets, and should work efficiently across different web browsers.

**Different Users of the Interface:**

1. **Customers**:
   * These are individuals using the system to register for driving lessons, access practice tests, and track their progress in courses.
2. **Administrators (Owner)**:
   * The system owner who has full access to all system features, including managing users, packages, lessons, and reviewing system activity logs.
3. **IT Officer**:
   * Responsible for system maintenance, managing accounts, resetting passwords, and overseeing security and system updates.
4. **Secretary**:
   * Handles appointment bookings, modifications, and cancellations on behalf of customers who call or visit the office, as well as managing driver and car assignments.

**What Each User Needs to Do Through the Interface:**

1. **Customers**:
   * **Register for an account**: Create and manage their user account, providing personal details like name, email, phone number, and address.
   * **Login**: Access their account using secure credentials and view their dashboard.
   * **Book, modify, or cancel driving lessons**: Choose from available dates, times, drivers, and cars to schedule or adjust lessons.
   * **Access online lessons and practice tests**: View, start, and track progress in their driving lessons, and complete DMV practice tests.
   * **Track their progress**: See which lessons they have completed and their scores on practice tests.
   * **View lesson history**: Access records of previous driving lessons and payments.
   * **Reset password**: Request a password reset link if they forget their credentials.
2. **Administrators (Owner)**:
   * **Manage users and roles**: Add, modify, or deactivate user accounts (customers, secretaries, IT staff) and assign permissions.
   * **Manage training packages**: Enable or disable specific driving lesson packages based on business needs.
   * **View system activity logs**: Track who has made appointments, modifications, or other changes in the system.
   * **Generate reports**: Review and export reports on user activities, lesson bookings, and overall system performance.
   * **View security alerts**: Receive notifications about suspicious activities, such as brute force login attempts or system errors.
3. **IT Officer**:
   * **Manage system security**: Monitor user activities, manage security settings, and address potential breaches or hacking attempts.
   * **Handle user account issues**: Reset passwords, assist users who are locked out, and ensure accounts are securely managed.
   * **Monitor system updates**: Manage system upgrades and ensure the platform remains up to date with security patches.
   * **Backup and recovery**: Ensure the system data is regularly backed up and can be restored in case of failures.
4. **Secretary**:
   * **Book, modify, or cancel appointments on behalf of customers**: Input customer information and make or adjust lesson bookings.
   * **Assign drivers and cars**: Manage the schedule of driving lessons, ensuring the correct drivers and vehicles are matched with the customers.
   * **View customer schedules**: Check appointments, make adjustments, and confirm details for customers.

**How the User Will Interact with the Interface:**

1. **Web Browser**:
   * All users, whether customers, administrators, IT officers, or secretaries, will interact with the system primarily through a web-based interface. This allows users to access the system from any device with an internet connection, including desktops, laptops, and tablets. The interface should support major browsers like Google Chrome, Firefox, Safari, and Microsoft Edge to ensure broad accessibility.
2. **Mobile-Friendly (Responsive Design)**:
   * The system must be mobile-responsive, meaning that it should automatically adjust its layout and interface based on the screen size of the device being used. This allows customers to easily book lessons, track progress, or reset passwords from their smartphones or tablets. While the system may not need a dedicated mobile app initially, the mobile version of the web interface should ensure smooth navigation and access to all key functionalities.
3. **Administrator and Staff Interfaces**:
   * For administrators, IT officers, and secretaries, the web-based interface must offer a more detailed view of the system, with dashboards that provide quick access to account management, scheduling tools, and system logs. These users will access the system from desktops or laptops, so the interface should be optimized for both larger screens and responsive views for tablet use.

### 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?*

* **Accessibility Features**: The design does not specifically address features that would make the system accessible for users with disabilities, such as screen reader compatibility, keyboard navigation, or color contrast settings. Ensuring compliance with Web Content Accessibility Guidelines (WCAG) could be an important addition to make the system usable for a wider range of customers.
* **Scalability**: While the design mentions real-time data synchronization and cloud-based infrastructure, it does not explicitly cover how the system will scale as the business grows. For example, how will the system handle an increasing number of users or appointments without performance degradation?
* **Payment Integration**: Although customer registration and booking were mentioned, specific details regarding the integration of payment gateways (e.g., credit card processing for lesson fees or subscriptions) were not addressed. There should be a plan for securely handling customer payments, managing refunds, and tracking payment histories.
* **Data Privacy Compliance**: The design touches on security but does not explicitly address compliance with data privacy regulations such as GDPR or CCPA. This could be critical since the system will collect personal information from customers, including potentially sensitive payment details.
* **User Training**: The design does not specify how users, particularly administrators, IT officers, and secretaries, will be trained on how to use the system effectively. This could involve documentation, tutorials, or in-person training sessions.

### 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?*

1. **Limited Offline Access**: The system is designed as a web-based application, which means it requires a consistent internet connection for most functions. Users without reliable internet access may face difficulties, especially in regions with poor connectivity. There is no provision for offline access in the current design, which could be problematic for both customers and staff who need to manage appointments or training materials in areas with limited internet access.
2. **Scalability Concerns**: While the system is built on cloud infrastructure, specific details around scalability were not addressed. If the number of users or transactions grows significantly, the system could face performance bottlenecks, such as slow response times, database overload, or server downtime. Additionally, the ability to scale without affecting user experience—such as how many concurrent users the system can handle before it needs additional resources—remains a potential limitation.
3. **Limited Customization for Non-Developers**: The ability to add or remove training packages is limited to enabling or disabling predefined packages through the administrator interface. Customizing or adding entirely new packages or modules might require developer intervention. This limits the flexibility for business owners who may want to frequently adjust or personalize service offerings without relying on technical staff.
4. **Dependency on External Systems**: The system's **compliance with DMV updates** and other external factors (e.g., payment gateways, email services) relies on integrations with third-party APIs. If this external systems experience downtime or change their APIs, it could disrupt core functionalities such as practice test updates or payment processing. There is no fallback plan for handling these disruptions in the current design.
5. **Data Privacy and Compliance**: While the design mentions security and encryption measures, it lacks explicit detail on how the system will handle data privacy regulations like GDPR or CCPA. Meeting these regulatory requirements can be complex, especially when dealing with customer personal data, and failing to comply could lead to significant legal and financial risks.
6. **User Support and Error Handling**: The design does not include any provision for user support within the system, such as live chat or automated help for troubleshooting. In the case of technical issues, users (customers or staff) may not have immediate resources available to resolve problems, which could lead to frustration and lost business. Similarly, error handling for unexpected issues, such as failed payments or server errors, has not been detailed, which could leave users stuck without clear recovery options.

**Resource, Time, Budget, and Technology Limitations:**

In addition to design limitations, practical constraints such as resources, time, budget, and technology will impose limitations on the development of the DriverPass system:

1. **Resource Constraints**:

* **Human resources** may be limited, particularly in terms of development, testing, and ongoing system maintenance. For example, if there is only a small team of developers or limited access to IT personnel, it may be difficult to meet feature demands, fix bugs, or implement updates in a timely manner. Additionally, managing security (e.g., responding to threats or monitoring logs) may be challenging with limited staff.

1. **Time Constraints**:

* **Time to market** is always a critical factor, especially for businesses looking to fill a gap in the market quickly. The system design as outlined could take several months to fully develop, test, and deploy. Meeting aggressive deadlines could lead to certain features being rushed or cut, which may affect overall functionality or user experience. Delays in the development schedule might also reduce the company’s competitive advantage if they miss an opportune launch window.

1. **Budgetary Limitations**:

* Building a comprehensive system like DriverPass requires a substantial budget, not only for the initial development but also for ongoing maintenance, security, hosting, and updates. If the budget is limited, there may be constraints on the features that can be implemented. For instance, advanced functionalities like multi-factor authentication, payment integrations, or scalability solutions might need to be delayed or simplified to fit within the budget. Additionally, a lack of funds for marketing, user training, or customer support infrastructure could limit the system's adoption and effectiveness.

1. **Technology Limitations**:

* Depending on the chosen technology stack (e.g., cloud platform, programming languages, database), there may be technical limitations in terms of performance, flexibility, or compatibility. For example, certain cloud providers or databases may have limitations on data throughput or storage, which could impact the system’s performance during peak usage times. Moreover, integrating third-party services, like APIs for DMV updates or payment gateways, could pose challenges if these systems do not fully align with the chosen tech stack.

1. **Legacy System Integration**:

* If DriverPass has existing systems or tools that need to be integrated into the new platform, this could pose challenges. Legacy systems are often harder to integrate with modern cloud-based platforms, potentially requiring additional development time and budget, or leading to functional compromises in the integration.

### 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.*

*A diagram with multiple colored squares

Description automatically generated with medium confidence*

Resources:

* Kendall, K. E., & Kendall, J. E. (2019). *Systems analysis and design* (10th ed.). Pearson.
* Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
* Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
* Cohn, M. (2010). *Succeeding with Agile: Software development using Scrum*. Addison-Wesley.
* Bass, L., Clements, P., & Kazman, R. (2012). *Software architecture in practice* (3rd ed.). Addison-Wesley.