# 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?***

* DriverPass is the client, they want to help their students pass their driving test. To do this they’ve proposed providing a comprehensive online resource for learner drivers which includes instructional content, practice tests, and scheduling features for real driving lessons. The main goal of the system is to provide learning in a controlled environment for our users to access data both online and offline, along with schedule meet ups. This will ultimately prepare them for the driving test and help them succeed.

**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 has identified a problem that many individuals struggle to pass their driving tests at the DMV. DriverPass aims to address this issue by providing an online learning module for clients to improve their chances of passing. Their online system includes the following components.
* Offline Lessons: An online space that can provide data readily available for download. This data includes entries and tutorials for the rules of the road, driving techniques and safety tips.
* Practice Test: An interactive program that provides a practice test. This tool can be downloaded and used to run various tests that prepare clients for the upcoming exam.
* Scheduling System: There are three different packages for an in person meet up and test run for driving. The scheduling systems will have features such as a preferred driving instructor, their schedule, and a meeting for the driving lesson.

**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?***

* **Develop a flexible online platform:** This should be an online platform that is web based, and working on all compatible browsers, (chrome, safari, edge). This will allow it a user-friendly and accessible UI that is scalable on all devices. From not just Desktop, but smartphones, and tablets. The platform will be cloud-based to ensure scalability and flexibility, thus able to handle a growing number of users without performance degradation.
* **Practice Test Simulation:** Platform should provide a test simulation tool, allowing users to download the tool and launch the exe. The tool can pull data from a file, allowing the simulation of a practice test for upcoming exams. Additionally, the simulation will offer feedback in the form of explanations for potentially correct or incorrect answers to help users reflect on their potential mistakes.
* **Online Classes:** Video tutorials will be another learning tool, offering a cinematic learning experience for users to easily digest the information. These tutorials will be structured in a series that unravel over the weeks, to ensure consistency and pacing over learning. Each video will cover the essential topics that the upcoming driving test will go over.
* **On-the-road training scheduler:** Users will be able to book an on-the-road training sessions with a certified instructor. Users can select various lessons from a calendar, view the availability, and choose preferred locations for the meet up. The system will support three different training packages, one with a six, eight, or twelve hours in total learning experience. Each with a combination of in-car and classroom lessons.
* **Security Features:** The platform will use industry standard security measures to protect user data and ensure secure transactions. Accessibility and role-based practices are implemented into the core features of this platform’s framework. Different levels of administrators, instructors, and students are placed throughout. Admins will have the option to reset passwords, block access, monitor system activity and utilize the trackers for all movement of system settings.
* **System runs through the Cloud:** By using services provided by AWS or Azur, the platform will use high availability, and minimal downtime in terms of technical issues. Cloud storage will store data automatically and provide disaster recovery. This will allow DriverPass to save a lot of money by not having to rely on physical server infrastructures, allowing them to focus on business operations. Cloud technology will enable seamless updates and scaling as user demand grows, ensuring the system remains reliable.

## 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?*

* **Environments:** The DriverPass system needs to be designed to be fully integrated into common such as Chrome, or Safari, a web-based interface. This needs to be fully integrated without any extra installations to ensure full compatibility with most internet-based platforms such as Desktop & Mobile.
* **Speed:** We wantto provide a premium user experience; we want the system to experience low loading times. Probably less than 2 seconds each page. This involves optimizing images, javascript, and other assets to reduce long loading times so that students will have quick access to the information.we want the interface to be responsive. Not just from navigating pages but core functionalities such as scheduling appointments, accessing practice tests, printing / downloading notes, and viewing video tutorials. All of this must operate with minimum delays or interruptions. We must ensure an optimized database between different system components to ensure that all features work together effectively.
* **Updates:** System updates should be done in regards to what’s given through the DMV meetings and user feedback. Since there is only one IT Officer, and most of the hardware maintenance is done through a third party such as AWS. Ian(the IT Officer), will be given regular performance metric and user feedback to have optimization projects for DriverPass(software maintenance) and have regular agile sprint goals to complete on a monthly bases. The system architecture should be modular and adaptable to potential future updates allowing for easy modification. This can be done through following polymorphic ideas when designing the framework during the early stages of the front-end development thus allowing for scalability with the system for future developers if any.
  + There will also be **server maintenance time periods** during the slow usage times for enhance security and performance.
  + We could also integrate a **version tracker** to roll back the system incase of emergencies. Previous stable versions could potentially be useful in the event a critical issue arises with the newest version.

#### 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?*

* While **Windows** is the most popular platform in terms of operating systems for Desktop, adding integration for others such as macOS and Linux/Unix. Mobile platforms are perhaps more important as they have a higher use-base and user preference, making it available for students & instructors on the go It would be highly beneficial to integrate such.
  + **Several advantages of using a browser** include not only the cross desktop, tablet, & smartphone compatibility. It also enables easier maintenance and updates as changes on the back-end development and technical aspects can be updated server-side without requiring extra installs from the clients.
  + While this might be an economic threshold. The **UI scalability** between smartphones and desktops will be a slight investment. The user interface will have to accommodate smaller screens on phones and tablets. Mobiles users should have access to most relevant features that are fit for an on-the-go type of functionality. This includes assigning appointments, viewing appointments, or viewing tutorials / notes. Adding features for a flexible UI to scale between desktop and mobile will enhance the user experience.
* **Back-End Tools:** to enhance scalability, the back end of the system should rely on a cloud-based database. This allows for the system to store information such as user profiles, test results, driving schedules and previous notes/sessions. A cloud-based architecture allows the added benefits of a flexible and scalable system to handle an increase usage of data storage as more lectures & tests are added into the system and more students use the system.
  + **Options of a popular cloud-based** architecture include MySQL or AWS. These would be ideal for this purpose as they provide powerful data management capabilities. They also provide advance searching systems allowing for the pulling of notes, reports, or lectures to be done with quick succession.
  + The DMV requires an **external API integration**. Information updates should be kept accurate and quick. The DMV will want to update these tests automatically through the server, reducing the need for redundant manual updates. It’s also mandatory by law I believe.
* **Security and Backup:** 
  + **Encryption:** user data is very sensitive. In fact, I think it’s mandatory by law to include measures such as a robust encryption system for user information. Such as passwords, credit card information, and billing addresses. Protocols like HTTPS/SSL can provide the encryption needed through the server-based architecture thus ensuring user integrity and trust. However since this is running through a cloud-based architecture like AWS or Azure, they will take care of the security part.
  + **Role-Based Access:** The system must implement role-based annotations to defines what features each user has. For example administrators will have access to all the system versions and changes throughout each system. They’ll even be able to track user activity and role changes. These are tools that students and instructors will not have, as they will have more limited options but still have the full extent of services the system is designed for.
  + **Back Ups:** the driver pass owner wanted a feature to track all significant actions and events from the past. Such as logins, data changes, and any potential unusual activity that may suggest a breach. A back up saved through the cloud infrastructure would be useful in providing data integrity as previous versions could hold and store the progress from previous work. Recovery plans are essential for minimizing downtown and wasted resources.

#### 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 Identity:** 
  + DriverPass will use a standard **authentication process** that is industry standard with login credentials. This includes the username, an email address, and a password. There will also be a role-based system. Each user being assigned a different role such as Admin, IT Office, Student, or instructor. These different roles will dictate the level of access each user has to the system.
  + Each role will have a **different Annotation.** Specific permissions are granted to each different entity. For example, an Admin will have full access to manage user accounts and track potential changes in the system incase of suspicious activity. Instructors will have the ability to manage student lessons, and view driving test scheduling along with providing feedback.
  + Each user will be given a unique **user ID** to accurately represent and record the activity of each individual. This tracker can be useful for potential suspicious behavior. It will monitor their activity and spot any potential breaches or unwanted advances. Administrators will be able to use these IDs to pinpoint accurate suspects.
* **Case Sensitivity:**
  + Both usernames and passwords will be **case-sensitive**. This will effectively add a large line of diversity when creating a password. In fact, a capital letter and some numerical values will also be required to increase password complexity. Effectively reducing the risk of brute force attacks or unauthorized access.
  + on the flip side, I tend to notice that **emails are usually designed to not be case sensitive** when inputting them. This is likely done for a quality-of-life feature as emails can be very long. We would do well to integrate this into our system as well.
* **Admin Notifications:**
  + Admins will be notified of any suspicious activity. There are several indicators such a large number of failed login attempts, detected security breaches, IP addresses associated with each login, and other potential future ideas of suspicious activity. With trackers in place to notify and detect these thresholds. Potential threats could be responded to.
  + Triggers for notifications include:
    - A brute force attack, or effectively a moment where an unusually high number of login attempts are made within a short time.
    - Logins from a previously blacklisted IP address.
    - Unusual activity patterns such as a high number of role changes from users to admins, or systems functionalities being changed.
  + Admins will be given the ability to lock accounts should any of the suspicious activities show up. Preventing further damage, or reverting accounts to previous states with a rollback feature.

#### 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 Changes Without Modifying the Code:** The system will implement a role-based access control that allows admins to add, remove, or modify users without changing any underlying code. This means that new users can be created through a standard account creation process, or admins can create their account and give them a different role, such as another Admin, IT Officer, Student, or Driving Instructor. These annotations will dictate the level of access.
  + **Role Modifications:** Admins have the highest authority, able to modify user roles at any time. Adjusting permissions and access as needed. For example, a user can be promoted to an admin role or downgraded to a more limited role like a student or instructor. This will likely never happen, but the flexibility ensures that user management is done without code modifications and done through a user interface instead via the annotative nature.
  + **Remove Accounts:** This is the equivalent of a ban or termination of an account. Suspicious activity or breaches of the terms of services will have to be kept into account and acted upon. Accounts may not permanently be removed, but the data provided through tracking could be useful for future cyber security developers to enhance and patch up the breaches.
* **Platform Updates:** Different versions of the OS will occur over time. Windows tends to receive consistent updates and while that’s the most used platform, the integration of placing this system on a browser-based platform such as Chrome, Safari, and Edge have given it the necessary flexibility needed to run on multiple different platforms / OS’ without the need to worry for future updates.
  + Building the system with widely accepted web standards and frameworks such as HTML5, CSS3 and JavaSCript, the system will be less prone to compatibility issues when browsers or OS’s receive updates.
* **IT Officers Access:** The IT officers will have nearly full access to the system to ensure smooth operations with security management and maintenance. These include the ability to reset user accounts, change permissions, track performance, and address potential issues such as server over loads or slow response times / system errors.
  + It officers will have the access to the system’s **data backup** and recovery features, allowing for them to be responsible for configuring and maintaining regular backups of user data, ensuring that in the event of the system failure or security breach, critical information can be restored thus leading to critical information or critical progress / work can be restored quickly and efficiently
  + IT will be able to perform taskssuch as monitoring, and update. To ensure the system remains secure, IT Admins will have the ability to disable outdated system components or software packages. This allows for the deactivation of plugins or features that are no longer necessary such as packages that the owner does not want.Other features include seeing a large traffic of reports in errors within a system and addressing them, or even using error tracking tools such as sentry to help.
  + **real-time monitoring:** officers will have access to real-time performance dashboard that display key metrics such as server load times, database response times, and system uptimes. These dashboards allow them to track the system’s performance at any given moment and identify potential bottlenecks. Tools like New Relic can provide these features.

#### 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?*

* **User Login Requirements:** Usually for a login, a unique username and a strong password is required. Usually, the criteria for a strong password is that it’s at least 7 words, and has a mixture of letters, numbers, or symbols. For extra security we can also add a two-factor authenticator including an email login by device, or a traditional authenticator. Such as an Authenticator App through Authy or Google Authenticator.
* **Secure Data Exchange:** The system will use encryption methods such as HTTPS and SSL/TLS to secure the communication between the client and the server. This ensures that all the data exchanged between users and the system is encrypted thus stopping the alteration of the system through malicious parties. However, these services are usually provided through the cloud-architecture.
  + **Personal information** such as credit card details, personal location, and user privacy must be stored and transmitted using secure encryption algorithms like SHA-256 for hashing password. The system should also employ secure sessions, where session tokens are encrypted and stored securely
  + There are also general audits and **monitoring** required for the system. The IT Officer will be tasked with going through and taking the time to monitor for activities, such as looking for vulnerabilities, and writing up reports on various assessments involving potential breaches within the system
* **Brute Force Attack Protection:** A brute force attack is a hacking technique that uses trial and error to gain access to an account or system. This is usually solved by giving an account a limited number of attempts, usually 5 until the IP address or the username in question goes under lockdown, usually 30 minutes. If the user waits 30 minutes and attempts to return to their account and they get it right, a third-party verification such as an email or 2F authenticator would activate and be required. Not to mention, an admin should also be notified of these breaches should they occur. Admins / IT Officers should also be given alerts for suspicious login attempts from unrecognized IP locations so that Officers could block the IP addresses perinatally. This is probably better done manually as interpretations of a security breach are subject to change.
* **Password Recovery:** It’s normally standard for users to simply have a password reset option below, usually recovered with a security question. There are other options such as having the user go through their email and be provided a private link to their email so they can reset the password there, or again we can use a 2FA authenticator to verify their identity and reset the password through those means as well. This is just a manual option that is provided to the user, or the user can call the admin instead and have them reset their account through an authorized figure. However, to add a layer of security. The admin would simply direct the user to verify their identity through their email or other means of proving identity to prevent scams or account theft.

### 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 be a web-based system allowing for the use of it through Phone, Tablet, and Desktop at any location using internet.
* The system shall allow customers to have access to online classes, practice tests, and review their progress and score.
* The practice tests are an on-going service that shall be updated in real-time based on the DMV rules or changes.
* The system shall be updated frequently, and adhere to the DMV’s requirements, as they are ever changing.
* The system shall be run through the server in a cloud-based database, for example AWS.
* The system shall allow students to register and create accounts online.
* The system shall have encryption-based security methods to keep sensitive information safe, such as payment options, user passwords, and user identity.
* The system shall validate user credentials when logging in, thus ensuring accurate individuality among the user base.
* The system shall have robust security features such as case sensitivity and 2Factor Authenticators for added security.
* The system shall provide a password reset feature for students who forget their password.
* The system shall allow students to schedule, cancel, or modify driving lesson appointments through their account or via phone/secretary.
* The system shall allow students to track and display the availability of driving instructors, the cars available, and lesson times.
* The system shall store customer information, including the student’s name, address, phone number, email address, and payment information.
* The system shall allow students to choose from three different packages.
* The system shall notify admins should any trackers for suspicious activity, such as brute force hacking, or data changes go off.
* The system shall track said data changes to give extra information for potential breaches and tracking the culprit.
* The system shall provide an interface for IT admins to manage user accounts, reset passwords, and revoke or modify access permissions.
* The system shall give admins overall control for needed changes without having to adjust the code.
* The system shall have automated updates through its cloud-based architecture to ensure browser compatibility, and tech related issues are kept to a minimum.

### 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.)?*

* **Intro:** The interface needs to be user-friendly, and accessible across different devices (desktop, tablet, & mobile phones). It must provide a premium experience for the different user types mainly students, administrators, IT officers, secretary, and trainers. The interface should be visually appealing, easy to navigate, responsive, and smooth. These qualities involve things like low loading times, and responsive navigation. Meaning low amounts of down time and bugs when clicking through the prompts. There was a mock interface illustrated for a home page for students, including features like.Online Test progress, Driver Note, Information (full name, address, city, state, zip, phone, email), special needs, driver photo, and student photo.
* **Different Users and their Features:**
  + **Customers/students:** View and schedule driving lessons, modify or cancel existing appointments, select driving packages, track progress through lessons, attend practice tests, download lectures & lessons, reset passwords in case they forget. Input information for account personalization.
  + **Secretaries:** Manage student appointments (schedule, modify, or cancel), Input customer information, answers phone when appointments are made.
  + **IT Officer:** Manage system maintenance and updates, reset passwords for users and other admins, modify user permissions, monitor the system for issues or errors, create reports for tracking user activity and system performance.
  + **Administrators:** Manage user accounts & roles (create, modify, or remove), track user activity and generate report on reservations and system changes. Enable or disable driving packages based on business needs.
* **Interaction with the interface:** The user will likely be able to interact with the interface through a desktop-based web browser. There are plans to include a mobile interface for user flexibility purposes, but the main framework will have full compatibility and functionality on a desktop computer with a point and click type of navigation.

### 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?*

* **Technology Access:** It is assumed that everyone (Students, secretaries, administrators, IT officers) have access to a stable internet connection. While DriverPass might not have a campus, there should at the very least be open to the public computers. Mobile phones and desktops at home should also be readily available.
* **Browser Compatibility:** It is also assumed that everyone is using the most up to date version of windows 10. Normally browsers such as Chrome and Safari update automatically, but I’m not sure if if some people may be limited to either Mac or Linux. Updates to windows 10 are free, so it will be assumed that the most stable version will be worked on and tested on windows 10 on Chrome-based web browsers.
* **Technologically fluent:** There wasn’t a lot of specification on a customer / student support team ready to help students if they weren’t able to navigate the website. While it was mentioned that the interface will be easy to navigate and smooth, it is uncertain whether some individuals may still have a rough time. While Administrators and IT Officers are expected to have large amounts of experience with navigating computers, students can be a different story. It is assumed that users will be able to do things like navigate the system, create accounts, schedule lessons, and reset passwords on their own.
* **Coud Hosting:** It is assumed that the cloud provider will provide well established services such as data backups, security, and availability, reducing the need for DriverPass to manage these in house.
* **DMV updates:** The DMV will be providing consistent updates to what the lessons and website will require. Practice tests will be provided and through APIs. These practice tests will constantly be updated automatically as per the DMV’s desires of what they want students to know and they won’t expect us to know their demands through omnipotence.
* **System Scalability & Modularity:** It is assumed that we will hire competent developers who will make the system highly modular to add or remove training modules without having to retrack on previously written code. The system is constantly being updated, and may need to be adapted to potentially future complicated features.

### 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?*

* **Resources Limitations:** A wise man once said that time is money, if you’re wasting time, then you’re wasting money. What that effectively means is that we have limited time and resources. There’s going to be a deadline for developers to make this project, and if problems or delays come into play. We may have to remove several secondary features. While illustrated to be important, things like mobile integration, UI design, or even optimization / user experience if it comes to that.
  + **Development Team Capabilities:** Not every team is going to work like a cog within a machine. Some may run into bugs or complex problems that have requirements beyond anyone’s capabilities. The team’s capacity to deliver all the features on time may be strained or even a brick wall entirely.
  + **Financial Constraints:** Most software developers work on contracted labor. If the project requires an accredited developer for a complicated programming task, the need for these specialized skills could constrain for a highly efficient budget.
  + **Technological Limitations:** Old systems that are unsupported such as Windows XP, or systems that are besides Windows 10, like Linux or Mac will likely experience different outcomes in terms of running the DriverPass system on their browser. Full functionality, compatibility and even performance may be hindered and provide a less than desire experience for some users if they attempt to use it on anything but the most widely used OS (windows 10).
* **Cloud Limitations:** While cloud infrastructures such as AWS or Azure offer significant advantages in scalability, cost, and reduced need for management in technical problems. They provide a slight burden to DriverPass in the limitation of giving the main company reduce overall control over their system. Crtical operations like backup schedule, performance optimizations and recovery protocols are largely managed by the cloud provider. This is a bit redundant because most renowned cloud provides sign legal documents under SLA which guarantees a certain performance metric of 99.9% uptime. There are rare scenarios such as natural disasters that involve AWS’ main region suffering from a power outage or a hurricane disturbing the internet. Of course the same could be said for DriverPass should a they possess a local server and the local server is instead the one suffering from a natural disaster, it will be up to DriverPass to own an in house protection infrastructure and recovery plan.
  + One such limitation is small in score, but **a very specified scenario** where certain encryption techniques are required by law for sensitive data might come into place. I know there are broad requirements for broad PIIs, but to break third wall. I don’t know much about driving school LMS’. In the event that special needs for certain information is required to have stricter security measures. A cloud infrastructure that has full control over the encryption and security protocols might be a problem here, and DriverPass might be forced to implement their own security measures for this specific instance.

### 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.*

