# 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 client for this project is DriverPass, a company aiming to improve driver education through a system built to better prepare students for their driving tests. The client wants to build a system that would be able to support both online learning and in-person driving lessons. System activities will include user registry, accessing course materials (uploading course materials by instructors for viewing as well), scheduling driving sessions, and taking practice tests. Also, the system must support secure data access, user role management, activity tracking, and appointment scheduling with instructors and vehicles. The purpose of our design is to provide a flexible and scalable web-based platform that meets these needs, improving the experience for both staff and end users (students). This system should perform the same regardless of the users’ platform (mobile, desktop, tablet).

### 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 is attempting to fix the societal problem of individuals failing their driving tests due to lack of proper preparation and training. A system with a combination of online course materials, practice tests, and live-driving road lessons with instructors will be used to accomplish this goal. The system should be a central hub that integrates all aspects of student registrations, learning, class tracking, and appointment scheduling into one interface (multiple pages are allowed within the system). Multiple user types such as admin, student, IT, and secretary should also be supported with varying permission sets. Security measures should also be implemented to protect the personal and financial information of students along with encryption and account recovery options. The system should also allow for flexible service package management to allow the business to fine-tune the services they offer.

### 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 completed, the system should allow for multiple user roles with permissions as follows:
  + Students:
    - Account registration and creation
    - Schedule, cancel, modify driving lessons online
    - Access online course materials
    - View test history (in person driving tests, online courses)
    - Securely update payment information
    - Reset password, recover account if password lost
    - View billing history and payment statuses/due dates
    - Leave feedback on lessons (online or in-person)
    - Receive upcoming appointment reminders
  + Instructors:
    - View driving lesson schedule
    - Set availability for lessons
    - Record student attendance (for in person lessons) and lesson completion
    - Enter lesson notes and give feedback to students
    - Access student profile for relevant information (test history)
    - Upload new online course materials
    - Access a dashboard to view current day activities and student progress at once
  + Secretaries:
    - Manually schedule appointments for students over the phone or in person
    - Input and update student information
    - View instructor availability, assign to sessions
    - Modify existing appointments (cancel, change time, change instructor)
    - View instructor and vehicle availability in a calendar format
  + Admin & IT:
    - Manage user accounts and permissions
    - View system activity (log who modified reservations)
    - Disable/enable or change training packages offered
    - Generate reports on various metrics like usage, scores, test completions, student count, instructor count
    - Run system health checks and performance analytics
    - Access logs of database changes
* In addition to these roles and permissions, the following should also be accomplished by the software:
  + MFA (multi factor authentication) for account creation and login
  + Encryption to protect payment transactions and login credentials
  + Run as a secure, cloud hosted web app
  + Export reports (likely as CSV) for offline use
  + Accept input from Admins in form of CSV reports to update database
  + Scrape from or integrate from DMV website to update rules or testing material
  + Notify users of change in DMV policies
  + Support scalability to handle growing user base of any/all roles
  + Minimize IT maintenance by utilizing scalable, cloud-based architecture
  + Compliance with ADA standards for accessibility (text size, color contrast, keyboard navigation)
  + Facilitate business growth by allowing DriverPass to offer new training modules, locations, or packages with minimal system changes.
  + Test environment for quality assurance before deploying updates to production

## 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 is expected to run in a web-based environment and should be optimized for industry standard responsiveness of maximum about 2 seconds of load time per page, providing users with a seamless experience (Nielsen, 2010). The system should be able to maintain this level of responsiveness while handling multiple concurrent user sessions, especially during peak traffic times. System updates and maintenance is recommended routinely during lowest traffic hours, along with daily system log generation and monitoring.

#### 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 system should be compatible with all of the major modern web browser platforms like Chrome, Firefox, Safari, and Edge while functioning consistently on devices like desktop, tablet, and mobile. The backend will be hosted on a cloud-based hosting service like Amazon AWS or Microsoft Azure to allow for scalability and flexibility as the DriverPass service grows. The program will run on Linux due to its ease of integration with cloud environments and modern backend technologies like Apache, Node.js, Docker, and Kubernetes. (RedHat, 2025). Linux is the foundation for scalable, cloud-native applications due to its stability, open-source flexibility, and enterprise-level features like automated security patching and container orchestration (RedHat, 2024). A design like this would ensure that the system could handle modern web traffic demands and would be cheaper than self-hosting the server.

#### 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?*

* Different users will be distinguished to the system from login. RBAC means instructors, students, secretaries, and admins will all be shown different interfaces entirely. For example students will only be able to view their own information, test results, and schedules while secretaries can manage scheduling for everyone. Proper access control helps maintain data integrity and reduces the chance of unauthorized modifications.

Input fields would be configured with formatting requirements like “MM/DD/YYYY” for dates, and proper validations for emails and payment cards. This information would be validated at the front and back ends, ensuring data integrity. Case sensitivity would be enforced in fields like usernames and passwords. The system will default to name case sensitivity for profile updates like names, addresses, etc. These precautions will keep information in the system accurate.

Logs would also be used to store actions like scheduling changes, failed validations, and payments with timestamps and account names. The system would notify an admin of potential issues if repeated unauthorized access attempts occur, if validation fails a certain number of times, or if payments aren’t working system wide.

#### 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 admin users of the system will definitely be able to make user changes without modifying the source code. A dedicated interface will be shown to admin users that will facilitate any actions they may need to take across the whole system. This flexibility will be made possible by implementing a dynamic user management system that stores user data and permissions in a centralized database, accessible through a secure backend. To remain compatible with browser or operating system updates the application will be built using modular technologies such as HTML5, CSS3, and JavaScript. This ensures that visual and functional components can automatically adjust to updates across different platforms without requiring major redesigns. Hosting on cloud platforms like AWS or Azure also has the benefit of support for patch management and version control. IT administrators will require elevated access privileges like the ability to manage system settings, monitor logs, reset user accounts, and deploy updates. They will not have access to sensitive user content like test results or personal payment details unless explicitly granted through role-based permissions. This separation of administrative powers supports both flexibility and data protection as the system evolves.

#### 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?*

* When logging in the user will be required to enter the correct username and corresponding password that is linked to the account in the database. This communication will be encrypted and secure using HTTPS with TLS to protect data. Password and payment data will also be encrypted for storage within the database using standard encryption algorithms.

The system logging system mentioned earlier would catch a brute force attack by keeping a log of all failed login attempts. A threshold can be set and when login attempts pass that threshold, the user can be locked out preventing them from more than a handful of login attempts. After a lockout users may be required to verify their identity through email or another secure method. If a legitimate user forgets their password, the system will provide a secure password recovery process involving identity verification, such as a password reset link sent to the users registered email address. For additional protection, multi-factor authentication will be required for admin and IT accounts. All administrative actions and security related events will be logged with timestamps and user identifiers. Hosting the system on platforms like AWS or Azure also allows access to advanced built-in security tools such as firewall management, intrusion detection, and compliance with recognized industry standards.

### 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 let users create an account and authenticate access using secure login system.
* The system shall utilize MFA for admin and IT roles for extra security.
* The system shall treat user roles distinctly and display custom interfaces/have appropriate permissions.
* The system shall let students schedule, cancel, and modify lesson times online.
* The system shall send alerts to students for appointment reminders and payment confirmations.
* The system shall let instructors view their schedules and mark student attendance.
* The system shall let instructors submit material feedback to students.
* The system shall let students view and complete online course material.
* The system shall let instructors upload and modify course material.
* The system shall let students view their course history, attendance, and grades.
* The system shall enable students to securely update payment information.
* The system shall let students securely view billing history and make payments.
* The system shall allow secretaries to manually schedule lessons, modify appointments, manage logistics (assigning instructors and vehicles).
* The system shall let secretaries view instructor and vehicle schedules in calendar format.
* The system shall enable admin and IT accounts to assign permissions and manage accounts.
* The system shall allow admins to export and import of CSV reports of student records.
* The system shall keep a log of activities and changes like account updates, failed logins, appointment changes, payment confirmations, and database modifications.
* The system shall temporarily lockout accounts with multiple failed logins and verify user identity.
* The system shall have a password recovery option involving verifying identity and secure email links.
* The system shall be a web interface compatible with desktop, tablet, and mobile devices and work across all major web browsers.
* The system shall support automated system health monitoring and notify IT staff of outages or abnormal behavior.
* The system shall connect to a centralized database that securely stores all student records, personal details, course progress, and test history.

### 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 interface must be capable of displaying dynamic information based on which role the user in the current session has. Students, instructors, secretaries, admin, and IT will all see a different version of the application, displaying only the relevant information to them.
  + Students will need to access course materials, scheduling, and profile details.
  + Instructors will have to access their schedules, classrooms, material within each class, student submissions, and provide feedback.
  + Secretaries will need to be able to manually add driving time slots, as well as see a calendar view of the teaching and vehicle schedules/availability.
  + Admin and IT roles will need to view or download system reports, manage user accounts, and perform system maintenance.

The user will interact with the web-based interface through a desktop, tablet, or mobile device.

### 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?*

* It is assumed that users accessing the system will have internet access and a device capable of running modern web browsers.
* It is assumed that all users will have a valid email address to register with and use for password recovery.
* It is assumed that the DriverPass staff will have received appropriate system training.
* It is assumed that the DMV provides some method for public access or integration to rules and testing materials.
* It is assumed that DriverPass will make necessary business arrangements for databases, encryption libraries, and cloud agreements.
* It is assumed that desktop, tablet, and mobile devices are responsive and will all support one application.
* It is assumed that all accessibility standards will be in interface design from the start.
* It is assumed that cloud services could automate backups, version control, and avoid data loss and downtime during updates.

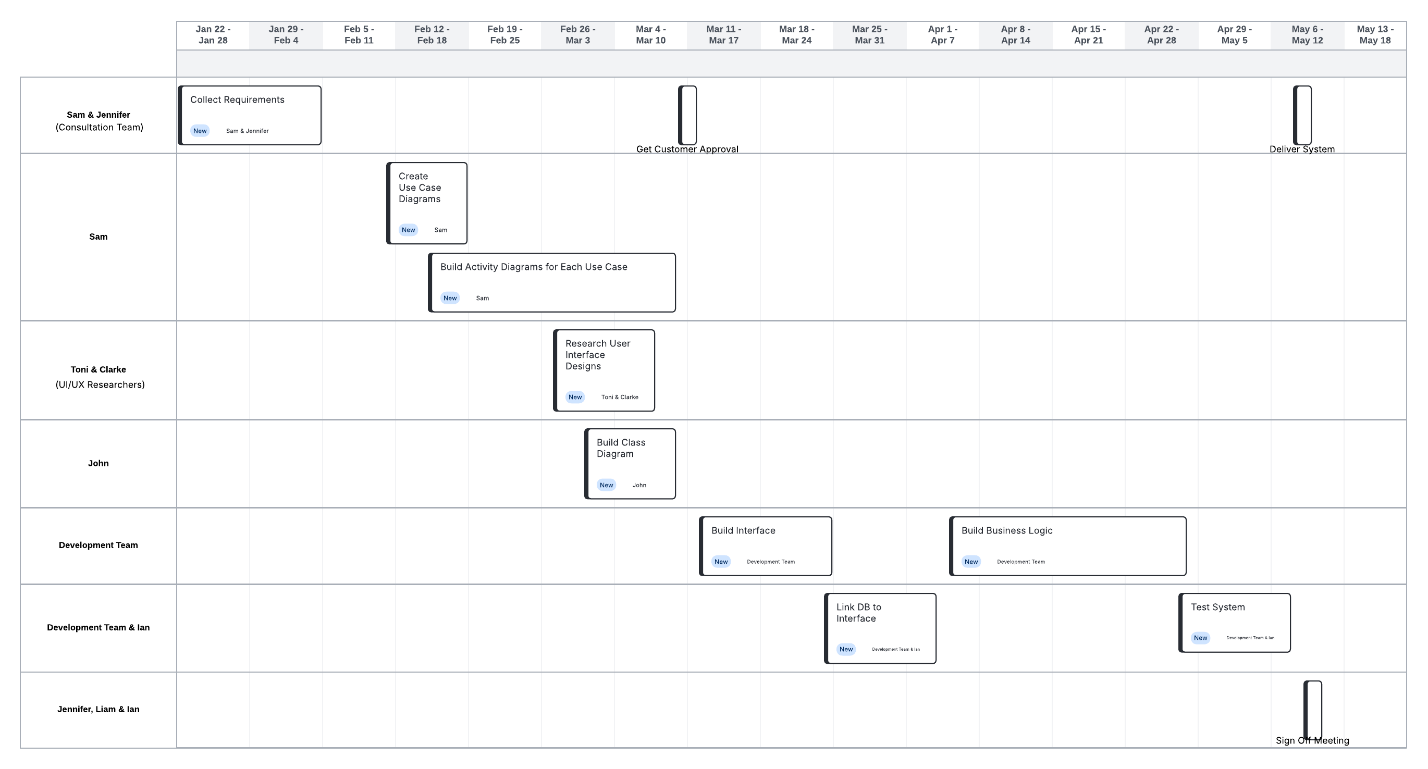
### Limitations

*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?*

* Initial deployment may be limited by budget constraints and possibly restrict capability for a mass amount of concurrent users.
* Real time integration with DMV materials not possible without public APIs, requires manual update instead.
* The system will be web-based only, not a native application.
* Offline functionality will not be supported.
* Reliance on third party cloud hosting and payment processing creates dependency on their uptime, price, and support.
* Including new access roles in the system in the future may require reworking the design.

### 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.*



**Resources**

Nielsen, J. (2010, August 9). *Response times: The 3 important limits*. Nielsen Norman Group.

<https://www.nngroup.com/articles/response-times-3-important-limits>

Red Hat. (2024). *Why choose Red Hat Enterprise Linux*. Red Hat. Retrieved July 30, 2025, from <https://www.redhat.com/en/topics/linux/why-choose-red-hat-enterprise-linux>

Red Hat. (2025, January 21). *How RHEL 10 supports modern cloud and DevOps workflows*. *webasha.com*. Retrieved July 30, 2025, from <https://www.webasha.com/blog/how-rhel-10-supports-modern-cloud-and-devops-workflows>