# 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

• A front-end web interface that can be accessed on PC and mobile devices  
  
• Backend logic that manages data changes, reporting, registration, and scheduling  
  
• A centralized cloud database that houses test, driver, car, appointment, and user data  
  
• A safe mechanism for user authentication and login  
  
• Role-based access controls for administrators, IT personnel, secretaries, and students  
  
• Integration point for data sync and DMV update alerts

### Purpose

*What is the purpose of this project? Who is the client and what do they want their system to be able to do?*

* Client: DriverPass
* Purpose: To provide a reliable system that allows student drivers to arrange on-road instruction, register for packages, take online tests, and monitor their progress toward passing DMV examinations.
* Support multiple user types and system access roles

### 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 offers thorough instruction (both online and in-person) in an effort to reduce the high failure rate of DMV test takers.  
  
• The system will address problems with ineffective communication, lesson tracking, and manual appointment scheduling.  
  
• Components: admin control, package management, test tracking, user registration, appointment scheduling, and security enforcement

### 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?*

• Ensure that training materials are easily accessible online.  
  
• Arrange driving lessons with teachers and cars.  
  
Organize lessons and packages using a single interface.  
  
• Monitor student test results and progress.  
  
• Give administrators and IT personnel the ability to log user activities, reset access, and manage roles.  
  
• Make sure modular design makes future changes manageable.

## 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 system has to be accessible online via web browsers around-the-clock.  
  
• For routine user tasks, response times for scheduling and updates must be less than two seconds.  
  
• Updates on routine maintenance every 30 days

#### 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?*

• Web-based platform accessible through a browser on Windows, macOS, and mobile OS  
  
• A relational database (such as MySQL/PostgreSQL) and cloud-based hosting are necessary for the backend.

#### 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?*

* Individual login credentials are necessary for every user.
* Passwords entered by users will be case-sensitive.
* Admins are notified when there are schedule issues or questionable activities.

#### Adaptability

*Can you make changes to the user (add/remove/modify) without changing code? How will the system adapt to platform updates? What type of access does the IT admin need?*

• The system enables lesson packages to be enabled or disabled without requiring code modifications.  
  
• The IT administrator can change accounts and reset passwords with complete authority.  
  
• The system may grow to accommodate additional teachers, students, and cars.

#### 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?*

• Admins can log in securely using multi-factor authentication and encrypted credentials.  
  
• SSL to ensure client-server communication is safe  
  
• After five unsuccessful tries to log in, lock the account and automatically notify the IT administrator.  
  
• Users can reset their passwords by sending a confirmed email.

### 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.”*

• When a user logs in, the system will verify their credentials.  
  
• Students will be able to register and enter their personal and financial information using the system.  
  
• Users will be able to select and sign up for training packages through the system.  
  
• The technology will enable staff and students to schedule, change, or cancel class appointments.  
  
• The system will schedule available drivers and cars for appointments.  
  
• Lesson hours, teacher comments, and student progress will all be tracked by the system.  
  
• Reports for management will be generated and exported by the system.  
  
• The system will alert users to updates and modifications to DMV regulations.

### 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.)?*

• Student, secretary, teacher, and administrative interfaces  
  
• Student dashboard: lesson plan, test results, and feedback  
  
• Instructor dashboard: student feedback, scheduled appointments  
  
• Admin dashboard: logs, user role management, and system use  
  
• Every interface needs to be responsive and mobile-friendly.

### 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?*

• Every user will have access to a contemporary gadget with an internet connection.  
  
• Training sessions usually last two hours and start and conclude on schedule.  
  
• A specific car is allotted to each instructor.

### 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?*

• New features or modules cannot be added by users without developer assistance.  
  
• Internet connectivity is required for full performance.  
  
• Phase I third-party integrations may be limited by financial restrictions.  
  
• Limited non-technical staff customization

### 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.*

