# 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 project is to build a cloud-based driver training web application for the client, DriverPass, that allows users to purchase and schedule online and in-person driver training. Administrator users should to download activity reports and data sheets.

### 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 sees that there’s a lack of hands-on driving training and aims to fill the void by offering a system to provide better driver training to prepare students for driving tests. The system requires a cloud-based back-end API, a reactive front-end UI cloud-based client, and a cloud-based relational database. The front-end UI will display the application and trigger the back-end API containing the business logic and passing user and registration data to the database.

### 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?*

* DriverPass shall have a system that provides students with three different driving training package options that include online classes, practice tests, and/or on-the-road training with varying training program lengths. The client will have the ability to disable these packages but also the ability to modify them in the future. The client will also be able to download data reports to Excel documents and activity reports. They also will have different role permissions for the different types of user roles: Admin (All access), IT Officer (Maintaining/Modifying the system), Secretary (Modify, delete, and create reservations), and Users (Modify, delete, and create reservations).

They will be able to keep track of reservations for driving lessons(2 hrs) and who took action on the reservation: create, cancel, or modify. Reservations will be able to be made by users on their account or by the secretary from their account if the user calls or comes in person. Reservations will include the date, time, pick-up location, drop-off location, driver, and car. User registration form will include first name, last name, username, password, address, phone number, state, credit card (cc) number, cc expiration date, and cc security code. The user will be able to reset their password from the login page.

The user dashboard will show test progress and reservations with the ability to create, modify, or delete reservations. The admin and secretary dashboard will both also display user contact information. There will also be a public page for contacting DriverPass. The application will connect the DMV through an API to display updated policies, rules, and sample questions for the driving test. The client will also have role-based security and backend controls built into the administrative settings for IT and admin dashboards for ease of use. Finally, the application’s front-end, back-end, and database will to be hosted on the cloud.

## 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?*

* This is a web-based system and needs to run web browsers including Safari, Chrome, Firefox, Microsoft Edge, Android, and iOS. The system should run very quickly; ideally, interactions should take no more than 300 ms since this will be an interactive learning system, and delays could deter users from using the system. The system should be updated often for security updates and bug fixes to protect the system and sensitive data and prevent website failures.

#### 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 able to run on Windows, Chrome, and Mac to support various systems students may use to complete assignments and check their progress and assignments. To support the system, the backend requires a user database to store user and role data and a reservation database to store reservation data. We also need to use a cloud provider such as Google Cloud or Amazon Web Service to deploy our API server to the cloud using a load balancer to mitigate traffic overload or denial of service attacks.

#### 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?*

* We will distinguish between different users based on their respective roles, which will have access to the least privilege necessary for each role. Input such as during registration and login will be case-sensitive, while creating appointments will not be case-sensitive. The system should inform the admin of a problem immediately by setting up log alerts and a ticket system for users to submit problems that may occur.

#### 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?*

* Changes can be made to the users, such as modifying, adding, or removing, without changing the code with an administrator dashboard that allows faculty with the correct permission to view the dashboard and make changes. The system will adapt to platform updates by scheduling tests with new browsers and operating system versions and updating the system accordingly while using continuous integration/continuous deployment to easily test and integrate the updates into the production system. The IT admin needs access to the frontend, backend, databases, system logs, and the support ticket requests to adequately maintain the system and make any necessary changes.

#### 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?*

* The user is required to enter a username and a password to log in. I can secure the connection between the client and the server by using Transport Layer Security (TLS) protocol with a certificate authority to authenticate the connection between the client and server. If there’s a brute force hacking attempt, the account in question should be locked and require a set waiting period to expire before being able to attempt signing in. A suspicious login attempt message should also be sent to the user of the account using the account information on file for the user as well as to the IT team in the form of a log alert. If a user forgets their password, they should be able to reset their password by using the reset the password link on the login page.

### 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 before logging in.
* The system shall allow users to log out.
* The system shall allow users with the Student role to register as users.
* The system shall allow users with the Student role to make a driving test reservation.
* The system shall allow users with the Secretary role to modify user reservations.
* The system shall respond to user interactions within 300 ms.
* The system shall validate all user input before saving.
* The system shall allow users to report a problem with the interface or system.
* The system shall log all errors and failed login attempts.
* The system shall send log alerts for potential problems to the admin.
* The system shall allow faculty to download activity and data reports as Excel documents.

### 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 different users for this interface include the students, secretary, IT officers, and admin. The students need to be able to register, log in, log out, make their reservations, modify their reservations, view course material, submit coursework, view course progress, and submit problems. The secretary and admin should be able to register, log in, log out, create anyone’s reservation, modify anyone’s reservation, delete anyone’s reservations, and view student contact information. The admin should also be able to download activity and data reports as Excel documents. IT officers should be able to log in to the system in the development environment with full access to the system to test, develop, and maintain the system.

### 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?*

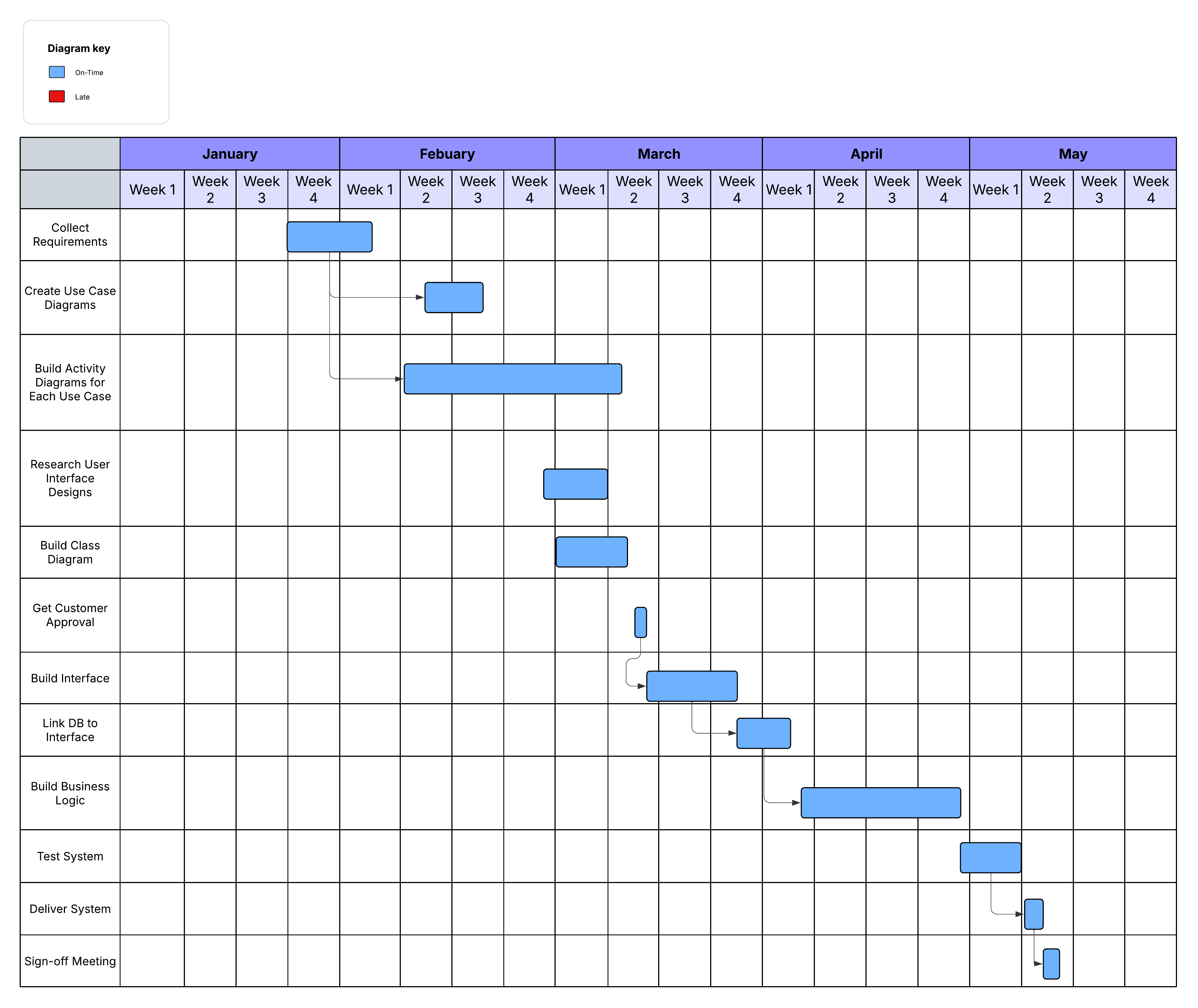
* Things that were not specifically addressed in my design above include how tests will be graded. I’m assuming that the tests will be auto-graded by the system once submitted by the student. The platform compatibility requirements were also never specifically addressed, so in my design I assumed that users will need the system to be compatible with the most used web browsers on Windows, Chrome, Mac, iOS, and Android to increase compatibility with system users worldwide.

### 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?*

* I see resource and budget limitations in my system design. Since the goal is for the system to be accessible to users worldwide, if system users grow faster than expected, then the website could crash if the system doesn’t scale well. On the other hand, if we use AWS as a cloud provider, we can take advantage of autoscaling, but there’s also a cost tradeoff. As the system users grow and the system scales, the cost of the resources will also increase, which could become too expensive to maintain.

### 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.*

**References**

Albey, J. (n.d.). Nonfunctional requirements. California Polytechnic State University. Retrieved October 4, 2025, from <http://users.csc.calpoly.edu/~jdalbey/SWE/QA/nonfunctional.html>

Amazon Web Services. (n.d.). What is a data center? Retrieved October 4, 2025, from <https://aws.amazon.com/what-is/data-center/>

Coara. (2020, August 27). Business requirements vs. functional requirements. Retrieved October 14, 2025, from <https://coara.co/blog/business-requirements-vs-functional-requirements>

Coggno. (n.d.). Common LMS course assumptions. Retrieved October 4, 2025, from <https://coggno.com/blog/common-lms-course-assumptions/>

D2L Brightspace Community. (n.d.). Add files from Google Drive. Retrieved October 4, 2025, from <https://community.d2l.com/brightspace/kb/articles/18123-add-files-from-google-drive>

Instructure, Inc. (n.d.). JWTs API. Retrieved October 4, 2025, from <https://canvas.instructure.com/doc/api/jw_ts.html>

Jones, D. (2009, August 23). LMS characteristics and limitations. Retrieved October 4, 2025, from [https://djon.es/blog/2009/08/23/lms-characteristics-and-limitations/index.html#lms-characteristics-and-limitations](https://djon.es/blog/2009/08/23/lms-characteristics-and-limitations/index.html%23lms-characteristics-and-limitations)

PM Study Circle. (n.d.). Assumptions and constraints in project management. Retrieved October 4, 2025, from <https://pmstudycircle.com/assumptions-and-constraints-in-project-management/>