# 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 (DriverPass) wants to create a web-based application that allows drivers to easily book appointments with the driver training service.

### 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 allow drivers to book appointments with the instructors quickly and easily.
  + The system needs to keep instructors up-to-date on the latest DMV rules and policies
  + The system needs to allow students to manage their appointments and availability.
  + The different components needed for this system include a user interface, a database for storing driver and instructor information, and an API for connecting the user interface to the database. It also will require a web framework to bind it all together such as Django.
  + The user interface should support serving images of the instructors and students

### 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?*

* The system should allow students to easily book appointments with instructors.
* The system should offer the three packages that DriverPass offers.
* The system should be able to handle a large number of users and appointments without slowing down.
* The system should be easy to use and navigate.
* The system should not over-book the lessons, as the company only has a set number of cars and instructors

## 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 should be able to handle many users and appointments without slowing down. It should be able to run in web-based, application, or mobile environments. The system should be updated regularly for bug fixed and feature additions.
* An Apdex score should be agreed upon with the client.

#### 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 front-end should run on any major web-browser as a SaaS app. The backend should run on Linux, likely Ubuntu due to cost. If the client is willing to pay money, RedHat CentOS can be used.
* The backend should be containerized and orchestrated by Kubernetes. The overall architecture of the backend should be microservices-based.
* Backend components should be stateless where possible. The DB will not run as a container.
* A DB will be required and should be maintained in Linux. A relational database such as MySQL should be used to handle this data, though the choices will be bound to which web framework is used. If required and the client is willing to pay, the DB can be hosted and maintained by a cloud service provider such as AWS, GCP, or Azure.

#### 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 system must be able to distinguish between different users and validate their credentials when logging in. User input should be case-sensitive, and the system should inform the admin of any problems.
* Input should be sanitized where possible and web-forms should be restricted to their intended input for both standardization and security reasons.

#### 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 should be able to adapt to platform updates and changes without affecting user experience or functionality. The IT admin may need access to configure settings or update databases.

#### 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 system should be able to disable accounts for terminated employees.
* The system should have a form of RBAC where admin accounts have different privileges than user accounts.
* Multi-factor authentication should be used for user accounts, though this was not explicitly asked for by the client.

### 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 authenticate and authorize students and instructors based on their credentials.
* The system shall allow students and instructors to manage their appointments and availability.
* The system shall allow instructors to make notes on student’s progress and needs.
* The system shall manage appointments and only book as many resources as the company has.

### 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 can be accessed through a mobile device or desktop browser.
* The interface should track the student’s progress and test results.
* The interface should show what test items are in-progress, passed, failed or not taken.

### 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?*

* Assumptions include that drivers and instructors will have access to a mobile or computer device with internet connectivity.
* Assumptions also include that the system will be able to handle a large number of users and appointments without slowing down, and that the database can store and retrieve information efficiently.
* It is also assumed that the service is highly available and, although not stated, it should have a disaster recovery plan.

### 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?*

* Limitations include a potential lack of resources for development and maintenance, a limited budget, and limitations on the functionality and user experience due to technological constraints. We are also concerned about the availability of caffeine for the developers.

### Super-Awesome Gantt Chart

