# CS 255 Business Requirements Document – Kyunghoon Lee

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 a company focused on providing driver training.
* Customers should be able to take online classes and practice tests.
* Customers would also be provided on-the-road training if desired.
* The client desires a registration and progress tracking system for the training they provide.

### 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?*

* Liam (the owner of DriverPass) believes that too many people are failing their driver licensing tests.
* DriverPass would like to be able to provide prospective drivers with an opportunity to prepare for the driving test at their local Department of Motor Vehicles (DMV).

### 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 provide a web interface providing an overview of the driver’s profile and their progress in the form of a “dashboard”.
* The system should be web-based or use some form of cloud service to allow data to be accessed remotely.
* There will be a roles/permissions system in place to control user privileges.
* There will be an audit log function to review all activity on the system.
* Customers should be able to make/modify/cancel appointment reservations online, in addition to over the phone or in person (lessons are 2 hours long).
* There will be a method to track which driver is assigned to which customer, time, and car.
* 3 initial driving lesson packages are offered, with the ability to disable a package if necessary.
* Customers, upon registration, should provide contact information for themselves along with a pickup/drop-off location.
* Customers should be able to reset their passwords if necessary.
* There should be a system in place to notify DriverPass if DMV policies change.
* There should be an input form for the student (or a secretary) to fill in the student’s contact information.
* There should be a page to contact DriverPass, as well as a way for DriverPass to contact its students.
* The client prefers “not to deal with backup and security” to minimize technical issues, so this should be handled be an external entity or a specialized individual within the company.

## 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 itself (such as the dashboard and all front-end access) will be web-based. This allows it to be accessed from nearly any device.
* The system should be compliant with modern web programming standards.
* The system should be built with speed and efficiency in mind, as the internet connection used to access it may vary from slow cellular data to broadband internet.
* The system should be updated regularly to keep pace with changing DMV requirements as well as to prevent security vulnerabilities related to a web-based platform. The client wishes to remain hands-off in terms of maintenance, so it may be best to conduct updates at times of low-traffic, potentially once a week.

#### 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?*

* As a web-based system, it should be accessible from any common platform (Windows, MacOS, Linux, mobile).
* For mobile devices, the user interface should resize and adapt to a touch-friendly interface.
* In terms of hosting, a Linux-based operating system has advantages in minimal cost and strong community support (given its open-source nature) as well as hardware efficiency, as running a web server should not require significant resources.
* The back end would require a database (either primarily cloud-based, or local storage that is regularly synced to another cloud database) to store user data.

#### 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?*

* All users should be identifiable by a unique username (e-mail address could be used here, for simplicity) and sufficiently secure passwords (enforcing minimum lengths and complexity).
* E-mail/username and password inputs should be case-sensitive (as a best security practice), and this should be visibly noted at any login screen. While e-mails are generally case-insensitive, less technologically familiar users may have difficulty with this.
* For security, there should be an admin-set upper limit to incorrect login attempts (for example, 5 attempts). An administrator should be notified at this point of several failed login attempts.

#### 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 owner (Liam) wants to have full access over accounts. Depending on his familiarity with programming, he may require a graphical user interface solution to add/remove/modify users. This may also be beneficial to administrators accessing the system remotely. Depending on how long this would take, it may be worth pushing this to later development.
* The system should always be running at the latest stable version possible for all software and dependencies. A regular weekly or monthly maintenance downtime should be scheduled to perform these updates at a time of minimum traffic.
* IT admins should have full access to every aspect of the system (assuming sufficient authentication protocols) so that they can make and implement changes as quickly as possible.

#### 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?*

* Users will log in using a unique username and password combination, to be authenticated server-side. E-mails may be used as usernames for the sake of simplicity.
* All data in transit should be encrypted. It would be best to implement HTTPS for the web server.
* Log in attempts should have an admin-set upper limit (3 to 5) for incorrect log in attempts. At this point, the account should be locked, and administrators notified of a potential brute force attempt.
* If the user forgets their password, they should be given the option to reset their password via a one-time-use link sent to the e-mail associated with their account. This e-mail should be manually input by the user as a security measure.
* If a user does not know their username or password, they should be directed to contact an administrator.

### 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 when logging in.
* The system shall display a dashboard upon log in, with an overview of the current user’s profile and progress.
* The system shall provide users with methods for signing up for or taking online classes, taking practice tests, and scheduling/modifying in-car driver training reservations.
* The system shall provide users with a method of contacting DriverPass (at minimum, by telephone and e-mail).
* The system shall provide users with visual representations of their progress, as well as instructor comments.

### 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 should be simple and direct for the sake of speed and efficiency across various devices.
* The users of the interface may include customers, administrators, instructors, and support employees (such as a secretary).
* The primary method for interacting with the interface will be through a web browser. When a mobile browser is detected, the interface should change to a touch-friendly interface with large buttons and dynamic widths (to account for device sizes).
* Accessing the interface from a mobile device should not compromise functionality.

### 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?*

* We assume the customer (or whoever is accessing the system) is using a reasonably modern, internet-enabled device (smartphone, tablet, desktop, laptop) and knows how to access websites.
* We assume the customer is using a web browser that meets modern standards (compatibility with Chromium-based, Gecko/Firefox, and WebKit/Safari covers all major browsers).
* We assume the customer has a baseline knowledge of operating a vehicle (for example, how to unlock and start the average car).
* We assume the customer has the intention of providing their contact information for account creation and scheduling purposes.
* We assume we have a reasonable time and financial budget allocated for development of the system.

### 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?*

* To avoid inflating costs, for our use case we do not need cutting-edge hardware.
* While reasonable time has been allocated for the development of each task, we cannot afford to fall behind on crucial elements such as user interface development or setting up a database.
* Administrator support is limited to the knowledge and time constraints of the employed IT staff.

### 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.*

