# CS 255 Business Requirements Document by Kirk Mashburn

Complete this template by replacing the bracketed text with the relevant information.

## 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 this project is to build an online driver training service for Driver Pass.
* This program should be available on the web so they can train their customers from anywhere in the world.
* Students can also schedule training in certain locations for road testing during a certain time frame.
* Students should be able to practice up to date DMV tests with the Driver Pass 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?*

* The system should allow students to take online tests that mirror that of their local DMV tests.
* students should be able to pick packages and time slots for personal on the road driving with instructors, while also being able to modify, change and cancel any packages.
* There also needs to be certain levels of access for the owner, IT officer, instructors and students.
* These services would need to be made available to both mobile and desktop users.
* A website and corresponding database would need to be set up.
* All the proper security and interconnecting parts would need to be included.

### 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?*

* Tasks seem to be based on dependency. The system design should be set up for the database, security and web functionality of the product.
* Tasks should be set up for the creation of the UI to allow the driving teachers to update their profiles and student driving scores in the form of a Learning management system.
* Students should have access to the LMS but with lower-level access, that allows them to contact their instructors, review grades and take online tests.

## 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 should be a web-based application hooked up to the cloud for easier security access. Besides regular security updates the site also needs to update rules and be tested based on when the DMV updates things on their end.
* Mobile compatible programing languages should be used to allow easier access to the website via phone.
* A LMS could help consolidate user requirements in a way that allows users to have better access to learning tools across desktop and mobile devices.

#### 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 backend could be run on either Windows or Unix. Unix would be preferable provided you have someone who knows the OS well. The database could be either a relational DB or NoSQL databases and their programming languages can vary.

#### 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 admin will receive notifications and reports any time a change is made in the system, different users will have different scopes of access with the Admin/Owner being at the top and student at the bottom. Case sensitive inputs would be best for passwords and usernames/email should be unique.

#### 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 design should be modular as to where the admin should have the ability to add, change or delete users.
* The design should have a process to create and modify customer plans and perform regular DMV rule updates
* These services can be rolled out in the form of class object creation with standard OOP encapsulation considered.

#### 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 need case sensitive usernames and passwords, with no duplicate usernames passwords should be hash encrypted with only their references stored. Passwords would need a certain length and required symbols to prevent brute forcing.
* Emails would need to be unique and allow for a password recovery process.
* If there is a brute force attempt login attempts should be paused and an email notification to the user should be sent out.
* Standard website and data processes should be used, there are multiple options available.
* Different levels of access should be given within the Drive pass system with Admin being the highest and student being the lowest.

### 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 allow administrators to create accounts and modify access.
* The System shall allow communication between users via email.
* The system shall allow students to take tests and request driving packages.
* The system shall allow instructors to grade and review students.
* The system shall allow students to track their progress.

### 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 user interface should allow a person to navigate from the main page to other services such as, viewing personal information, a place to take online tests, a way to track their test progress.
* The UI should have a section that displays the driver and instructor photos.
* The UI should have a section for Driver notes and include any special needs or requests for the students.
* The UI should allow an easy way for users to access the LMS on mobile and desktop.
* On mobile the LMS can simplify the view in a way that makes mobile navigation easier.

### 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 that the user will have the means to create an account, I.e. Internet access, email, etc....
* We assume the user will have the ability to respond via phone.
* We assume that the user will have some form of web browser to use our 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?*

* A limitation would be Drive pass’s requirement for a cloud-based database, we would need to set realistic expectations of how their product would work within the cloud.
* The budget would need to include the scope of creating the LMS, Website, Database/Server setup, programming hours and testing.
* The Driver Pass product need to Programming languages that are compatible with their desired product specializations.

### 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.*

*A screenshot of a computer

Description automatically generated*