# CS 255 Business Requirements Document Template

## System Components and Design

### Purpose

The goal is to help student drivers to learn and pass driving test, and fill the gap in the online/digital market and benefit from that as a company, which will be a mutual win for both customer and DriverPass owners.   
It should be accessible from anywhere and people can sign up, book classes, driving lessons, and take test online or in person.

### System Background

As mentioned above, the DriverPass application idea came from recognizing a gap in digital market of driving test and teaching market, and is aiming to fill that gap and help people in getting/passing their driver license and driving skills journey.   
Implementing the application as a web-based software enables customer, trainers and employees to have access from anywhere, however requires more security attention. Each use should have limited and only necessary access to the information that they really need. The application should [provide option for booking, modifying reservations, payment, and 3 preset packages should be offered through it to the customers.

### Objectives and Goals

The final and fully functional application should allow customers to sign up, reserve class, test, training, etc., modify or cancel them, and also provide report and access to the employees and admin to se reservations, modifications, cancellations, etc. and each employee should have different and appropriate access to the data and privilege for modification, based on the role and need.  
Selection of the server side OS/platform, programming language and technical factors of designing application methods such as UML diagram, process mode and object model are very important.

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

The application needs to be web-based/online and therefore a could-based solution seems to be very reasonable. It should be easy and fast to access, while safe to browse and keep customers privacy and personal data safely. The latest DMV changes should be updated directly from DMV website, and client should be able to add features, edit packages and update capabilities at anytime that they need it.

#### Platform Constraints

Depending on the budget and programmed g language that is chosen, the answer may be different. However, in general, Linux is a good choice for many cloud-based and online applications and is very cost-effective as well.

#### Accuracy and Precision

First of all, system need to make sure each user is choosing/getting unique ID and the password has minimum number of characters (6-8) with certain characters that make it harder to break with guessing; the password should be case sensitive as well. The system should auto-block de-activate account after certain number of unsuccessful attempts (3-5) and inform admin in such situations immediately.

#### Adaptability

User modifications and similar activities should be implemented as a feature without any need for coding. After testing and working with the release candidate version, client may feel that they need new features or may want some functions to be edited, upgraded or removed. After gathering all the info, updates may be performed. IT department may have the ability to enable/disable modules, users, employees and change the level of access for each employee.

#### Security

Username and password and option to 2 step verification are suggested for user login.   
Cloud service providers with good history and reputation are suggested for the hosting the application and data, so that they communication between the server and clients stays safe and secure.  
As mentioned above, the system should disable user’s account after certain number of un-successful attempts (3-5 for instance) to prevent brute force hacking, and unwanted access to the customer’s data.  
In case user forgets his/her password, their email or phone number can be used to send reset link to a temp password/code, so that they can regain control of their account. It is also suggested to verify user’s identity by asking the questions that they answered during sign up, before sending the reset info.

### Functional Requirements

The application needs to validate user identity/credential before letting them to log in  
The application needs to offer different packages and make them available for users to reserve/book  
The application needs to be/run fast, safe and reliable  
The application needs to provide reports to both users and employees about their performance, history, passed tests, etc.  
The application needs to provide educations material such as training classes, videos and tests

### 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 must give access to customers to book for available packages, classes, etc. and be able to edit those reservations  
The interface must give access to run report from reservations, modify, update or remove any reservations  
The interface must run smoothly and efficiently on different devices and platforms such as cell phone, tablets, computers, etc.

### Assumptions

We do not know about the budget, target audience age, preference, accessibility to the technology (such as internet speed, tec.)

### Limitations

Not knowing the exact budget and having only 5 months to complete the project are the main opportunities. Shorter time means more people need to work on the project and the bigger the team gets, the harder it would be to manage it, be efficient and communicate effectively.

### Gantt Chart

I used an Excel template instead of Lucid and did not get a change the day to week, as it needed to change some formula that made everything out of order for the whole sheet. I hope that it shows the concept and is good enough for the project.

