# CS 255 Business Requirements Document Template

## System Components and Design

### Purpose

* The purpose of this project is to take advantage of the lack of online driving education tools and courses. Liam, the owner of DrivePass wants to create a scalable web application that clients can use from any device with web connectivity to teach courses to users that are learning to drive. Applications similar to this one are Blackboard, and D2L.

### System Background

* The system shall be accessible on any device with web connectivity.
* The system’s material must be available offline – i.e., the ability to download reports and other data.
* Tiered account access for security: Admin role, IT role, teacher roles, and student roles.
* Individual, unique logins are required to protect data, and the system.
  + System should allow users to reset their password
* Record tracking: the system needs to notify the admin when reservations are made, cancelled, and modified.
  + The system must allow users to create reservations for driving lessons. The data related to the reservation is time, date, users’ name, address, phone number, state, pickup location, and credit card information.
* A database consisting of at least two tables to tie user’s IDs to their data.
* The system needs to incorporate three different driving packages when the user is creating an appointment.
  + The admin role must be able to disable and enable these packages.
* The system needs to connect to a DMV API that will update the system with any new DMV rules.
* The system needs to be cloud based.

### Objectives and Goals

* Once completed, the system will be able to allow users to register/sign in to DrivePass and manage their in-person appointments, study modules, take tests, and view their progress in their course.
* The system will utilize a database to store the user’s data alongside their ID – this will allow the user’s data to remain persistent.
* Upon the creation or modification of an appointment, the owner will receive a notification detailing the changes.

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

* Must be a web application
* System must be built to allow future updates
* System must be updated during peak downtime
  + Users must be notified of updates in advance
* System must load in a web browser within 2 to 5 seconds

#### Platform Constraints

* The system must run on all major operating systems (Windows, Unix), and browsers (Safari, Chrome, and other chromium browsers).
* The system will require a database to log user persistent data and class data.
  + The database will require several tables (ex: user data table, class data table)
* The system will require different access levels for user accounts.
  + Student, teacher, IT, and admin access levels.
  + Students have the most restrictive access – only allowed to view and edit certain content.
  + Admin account has access to show/hide classes.

#### Accuracy and Precision

* The system must require users to have unique login IDs.
  + IDs will be checked against the database to ensure the ID has not been used before.
  + IDs are case sensitive and must include uppercase characters and numbers.
* The system will only allow 5 consecutive invalid login attempts before it flags the attempt as a security risk.
  + After 5 attempts the system will lock the users account and notify the IT account user.
* The system must require users to register two factor authentication.

#### Adaptability

* Since the system utilizes a database, the IT admin will not need to use code to edit user data or modify courses – there will be a UI for this.
  + Users will have text fields on their profile page that will allow them to edit personal information without requiring code modifications.
* The system must be built to allow future updates, these updates may include bug fixes, additional content, etc.

#### Security

* All users must use two factor authentication to login
* The system will use HTTPS to authenticate and secure the connection between the web application and the browser.
* For password resets, the system will send a code to the user’s email. The user must then enter their code into the prompt provided by the application.
  + Once authenticated, the user may reset their password.
* The system will only allow 5 invalid login attempts before it disables the users account and notifies the IT admin. This system is in place to combat brute force attacks.
  + Two factor authentication further negates brute force attacks.

### Functional Requirements

* The system shall display the UI in an easily digestible view for all users.
* The system shall validate connections from client to server.
* The system shall allow the implementation of future updates.
* The system shall validate user credentials upon login.
* The system shall have different authorization levels (student, teacher, admin, IT).

### User Interface

* The interface needs to be accessible to all users.
  + The system must have accessibility settings for users to enable.
* The UI must be clutter free and easy to navigate.
* Teacher and admin accounts must have the option to add/edit/hide content.
* Student accounts must be able to download and view content.
  + Student accounts cannot modify content but can edit content that requires their input (ex: quizzes, forms, and tests).
* The UI must present content in ‘module’ form.

### Assumptions

* It is assumed all users understand how to use web browsers.
* It is assumed all users have internet access.
* It is assumed that users understand how to use systems like DrivePass (ex: blackboard, D2L).
* It is assumed the system will handle all simultaneous users without breaking in some capacity.

### Limitations

* Building the system to work well on every major OS is a limitation.
  + The mobile version of the system will require a mobile friendly UI.
  + This limitation is burdened further by the time limitation.
* Time:
  + The system must be completed in 15 weeks, this is a short period of time for a web application of this capacity.
  + To meet this requirement, some features may have to be introduced in future updates.
* Budget:
  + This project will require a team of developers, but the budget may not allow for the optimal number of personnel.

### Gantt Chart

Timeline

Description automatically generated