# CS 255 Business Requirements Document – Matthew Berdecia

## System Components and Design

### Purpose

* This project is for one of our clients, DriverPass, a student driver’s information and services company. The purpose is to create a web-based system that offers booking, rescheduling, and cancellation of student driver sessions. They want their system to provide access to a library of materials as well, including videos, available online and for download to view offline.

### System Background

* DriverPass noticed that too many people were failing their driving tests with the DMV. The believe that the failures are a sign of poor education and training.
* DriverPass looks to offer better and more flexible education and training to young drivers as a solution to this problem.
* This education requires a system that allows people to purchase and schedule classes on their own time.
* A cloud provider or a webserver is required for this system to store data online and offsite from their company.

### Objectives and Goals

* The system as a whole should receive updates from the DMV with the latest materials needed for drivers and notify relevant personnel.
* The system as a whole should provide end users with a way to automatically reset their passwords.
* The system should provide management with the following functions
  + Add new administrators
  + Remove administrators
  + Turn off packages no longer offered
  + Reset passwords for administrators
  + Disable packages no longer desired
* The system should provide administrator users with the following functions
  + Reset user passwords
  + Disable packages no longer desired
  + Generate and print activity reports on individual accounts
  + Schedule training (or re-schedule)
  + Cancel training
  + Update online materials
* The system should provide end users with the following functions
  + Purchase a package
  + Schedule training (or re-schedule)
  + Cancel training
  + Access online materials including practice test history

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

* This will be a web-based project that will need to work with all major internet browsers.
* This application needs to run on any operating system (particularly Mac, Windows, and Linux) via the most popular browsers (at least).
* The system should provide response times of less than 30 seconds.
* The system should be updated as needed or when the client receives updated from the DMV regarding new materials.

Rationale: The client requested an online booking system as well as online materials be made available to students who purchase packages. The simplest way to accomplish this is to make this a web-based project like a website or web-app rather than an application requiring installation on the operating system. This web environment should be compatible with all standard operating systems because we do not know what type of OS the end users will all have. A 30-second maximum response time makes sense because some systems are complicated. The system should be faster, but the maximum wait time should not exceed 30 seconds. The updates to the system should be as needed because once it is operational the only necessary updates requested by the client are when the DMV puts out new information.

#### Platform Constraints

* The system should run on a Windows operating system.
* The back end does require at least one database.
  + If possible, the database will hold the course materials, user tables and profiles, relevant DMV data, billing information, and appointment schedules.
  + If this becomes too convoluted to manage, separate databases can be made for each item type like a database for user tables and profiles and another for course materials and test materials.

Rationale: This system will need to be able to serve many users simultaneously. This system also will run as a server-client setup. Furthermore, the website would quickly get bloated with all of the content like online classes, content, and materials as well as all of the user login information for every user and billing information for paying users. This is too much to keep on the front end, and would pose very serious security risks. At least one database should be used for this sort of project.

#### Accuracy and Precision

* The system will distinguish between different types of users by unique user ID numbers, login names, and passwords.
* The ID number should be assigned by the system and should not be changeable, but the login name and password should be set by the user and changeable if needed.
* The input for login name (username) should not be case-sensitive, but the input for the password shall be case-sensitive.
* In the event of a problem with the system, an IT admin should receive an email notification within 5 minutes of the event.

Rationale: Two different major types of users are needed, admin and end users. Therefore, the system needs to be able to distinguish and offer the functionality associated with the account type to the user logging in. The ID number should be assigned by the system to make it easily searchable and to avoid any duplicates. The username should be unique and non-case-sensitive to avoid mix-ups like someone having username JackJones and another having the username jackjones. Passwords should be case-sensitive for added security. A five-minute response time in the event of a problem is reasonable if the problem has not resolved itself. The IT admins don’t need an email every time the internet fluctuates for a minute or two or the server goes down and boots right back up.

#### Adaptability

* The system should provide a way for admin users to add/remove/modify users without changing any of the source code of the system.
* The system should adapt to updates received from Microsoft. The server should have no issues downloading and installing said updates.
* The IT admin should have administrator access to the system. They should be able to see everything about users and other administrator accounts. They should also have access to hide course packages offered by the company, but they should not have read or write access to the source code.

Rationale: The admin users need to be able to remove users, either end users or other admins, in the case of a customer no longer wanting an account, or an admin being terminated or voluntarily resigning. It would be overly cumbersome to have the developers come in every time an account needed to be modified, added, or deleted. The system should have no problems downloading and installing Microsoft updates and applying those updates, generally via system reboot. The client requested that an admin user be able to hide inactive packages from customer view, but this cannot be used to give them access to the source code because it would create an opportunity for unauthorized edits which could crash the system.

#### Security

* The user (admin or end user) must go to the website to log in. They must provide their username and their password.
* Secure connections are established by use of SSL encryption. This protects data by using a TLS handshake, which generates session keys to encrypt all data within the session.
* If there is a “brute force” hacking attempt made on an end user account, the system should temporarily lock the end user out of their account for 10 minutes following the fifth unsuccessful login attempt.
  + If the user is an admin user, the lock out should happen after the third unsuccessful attempt.
* If either user type has 10 or more failed login attempts, their account should be locked until someone at the company unlocks it for them (they must call in and validate who they are).
* If a user forgets their password, they should have access to a “forgot password” link that allows them to reset.
  + This should ask them to provide their username and email address linked to their account to receive a rest link.

Rationale: Users must provide an identifier (username) and password to unlock an account because it is the cleanest way to validate a user. Two-factor is an option, but, I believe, unnecessary for this level of project. SSL is widespread and easily-understandable and should be used for establishing secure connections because it encrypts data in the session. This gives users confidence and prevents the company data from being accessed without authorization. Brute force attacks can be mitigated with slow-downs (the 10-minute lock out) and stopped entirely with locking the account and requiring a person to unlock it. Using this method makes brute force wildly impractical.

### Functional Requirements

* The system shall determine the user ID and information from the login credentials.
* The system shall provide end users the following options: review packages, purchase package, schedule/cancel/modify in-person training, schedule/cancel/modify behind-the-wheel training, review progress/profile, and logout.
  + The system shall provide special options based on the package that the student is subscribed to including: access online classes, take practice tests, review online content, and download content for offline work.
* The system shall determine the role of a user that has logged in.
  + For admin users, different options must be displayed including: add/modify student profile, add/modify admin profile, delete user, reset user password, validate user using security questions, schedule/cancel/modify in-person appointment, schedule/cancel/modify behind-the-wheel training, print activity report, download content for offline viewing, and disable purchasable packages.
* The system shall email customers receipts when they purchase packages.
* The system shall notify instructors via email when they have a session booked for training or behind-the-wheel instruction.
* The system shall validate users attempting to enter restricted areas of the site (like online classes which are restricted based on the level of user).
* The system shall provide technical reports to admin users when they request via their on-screen menu.

Rationale: The first step is validating a login by returning the correct user. This determines what will be displayed throughout the rest of the system. Once a user is logged in and the correct profile is accessed, the system determines end users vs admin users. For end users, the system needs to display the mentioned features which are further determined by the package the end user has purchased. These are simply based on the specifications provided by DriverPass. The admin users need the functionality outlined above for the same reason. They need to be able to perform tasks like resetting passwords without calling the developers every time as well as disable packages that are not currently offered. The system should provide some sort of confirmation to instructors and drivers when sessions are booked. This will keep everyone on the same page. The validation of users when they attempt to enter certain areas of the site will prevent unauthorized access for security reasons. The final requirement about reports will enable the admins to determine what was done on an account and by whom in case of a dispute. It will also provide them with detailed information about their software and its effectiveness.

### User Interface

* Many possible physical user interfaces exist for this system. All share at least some features in common.
  + The user interface shall have a display screen.
  + The user interface shall have a keyboard or simulated keyboard.
  + The user interface shall have a mouse or touchpad.
* The users will interact with the interface using a browser, whether a mobile browser or computer (desktop/laptop) browser.
* The different users for this interface are customers (or end users) and administrators (or admins).
* The user interface for customers consists of basic messages including:
  + Enter Username
  + Enter Password
  + Select Package
  + Enter Payment Information
  + View/Edit Profile
  + Change Password
  + Schedule/Cancel/Modify Appointment
  + Access Online Training Materials
  + Take Practice Tests
  + Download Content
* The user interface for admins consists of basic messages including:
  + Add/Modify/Delete User
  + Change Password
  + Print Reports
  + Hide Packages
  + Schedule/Cancel/Modify Appointment
  + Download Content

Rationale: Since this is going to be a web-based system, the interface is a computer or mobile device using a desktop or mobile web browser. The user will need access to an input device like a touch screen or a physical keyboard and mouse. There are two defined types of users, customers and administrators. That is the determining factor as to what content is displayed in the interface. The rest of the points are just enumerations of the core functionality the software should provide to each type of users.

### Assumptions

* The users (all types) have reliable, high-speed internet access.
* The users have computers or phones that have proper input devices (keyboard and mouse or touch screen).
* The server providers can provide the bandwidth to support any number of users on the service at any given time.
* The users (all types) have electricity or battery power sufficient for their own needs on the system.
* The users (all types) are using relatively up-to-date web browsers to access the content.
* The reports requested are the only reports deemed essential in the design at this time.

Rationale: These assumptions have been made for individual reasons. The internet access and electricity are because no online system would work without those two things. The server provider bandwidth is an assumption that was made because no mention was made of expected numbers of users and server load in the discussions. The web browsers assumption was made because clients should not expect developers to develop for every possible browser like the old Netscape Navigator or AOL browsers that have statistically zero use today. The final assumption about reports was made because several other potential reports come to mind, but it is up to the client to tell us what they need and since no mention was made, I would not have my team generate all sorts of possible reports when they were not asked for directly.

### Limitations

* The system cannot function without power.
* The system cannot function usefully without internet access.
* The system relies on Microsoft Windows. Therefore, if a major problem affecting all Windows servers were to come up, the system could be compromised until a patch becomes available.
* The timeline is relatively short for producing the system which could impact desired functionality.
* The system would have limited real-time notification ability due to the fact that it is browser-based. If a user is not logged in and active, they may miss a cancelled appointment or something similar.
* The system has no way to prevent unauthorized republication of course materials since the client requested download access for said materials.

Rationale: The rationale for each of these is different, with the exception of the power and internet bullet points. The power and internet limitations are for any web-based system. A web-based system needs power and internet access at the server side and client side in order to function as designed. The Windows limitation is because Microsoft is a third party whose products we are relying on. If their product fails, so does ours. The timeline limitation is due to the fact that the timeline is less than five months. That seems like a quick turnaround for a system with so much going on. The real-time limitation is common to many web applications as well since you have to be in an application to get the notifications from it. This could be mitigated with email, but that assumes reliable internet coverage as well. The final limitation in regards to republication is a serious one because people could have one person pay for the classes, download all the materials, and then share it with potentially thousands of other people. This should be addressed in a future design document to mitigate that threat.

### Gantt Chart

