# CS 255 Business Requirements Document

## System Components and Design

### Purpose

* The client is DriverPass.
* DriverPass seeks to fill a void in the availability of driving training materials for students seeking to take the drivers’ licensing test at their local departments of motor vehicles.
* DriverPass wishes to provide training materials and resources for young drivers as well as in-person lessons. Their system should be able to handle all of these components.

### System Background

* The DriverPass CEO believes many young drivers fail their first exams and that his company can help rectify the knowledge or information gap underlying this problem.
* System components:
  + Study materials: online classes, practice tests
  + Driving lesson reservation system
  + User and admin account system
  + Logging, tracking

### Objectives and Goals

* Online classes
  + Connect to and update via DMV
* Practice tests
  + Connect to and update via DMV
* Elective on-the-road training availability
  + driver matching system
    - 10 cars, each with its own driver
  + tracks reservations, cancellations, and modifications
    - printable activity reports
  + tracks the matching of users to their instructor, the car, and the time of the lesson
    - printable activity reports
  + Three discrete driving packages: **This component of the system should be customizable later, but, for now, each package should be vulnerable to being disabled at will by the admins**. Currently, all lessons are given in two-hour increments.
    - Package One: Six (6) hours total driving-training time (3 lessons)
    - Package Two: Eight (8) hours total driving-training time (4 lessons) plus in-person instruction on DMV policy/rules
    - Package Three: Twelve (12) hours total driving-training time (6 lessons), in-person instruction on DMV policy/rules, access to online class with all content, including practice tests
  + Online and telephone user registration
  + User registration parameters:
    - First name
    - Last name
    - Address
    - Phone number
    - State
    - Credit card number
    - Expiration date
    - Security code
    - Pickup location
    - Drop-off location
* Cloud-based web server hosting
* Location agnostic data access
  + Online data access and update
  + Offline data access
* User types:
  + Boss/CEO
  + IT admins (rights over all accounts, including password resets)
  + Secretary (can make, cancel, reschedule appointments on behalf of users; can make or update user profiles)
  + Users (can make, cancel, reschedule appointments; can access lessons and materials; can change their own passwords; can make or update their own profiles)
* User profiles:
  + Online test progress
    - Test names
    - Time each was taken
    - Scores
    - Statuses (not taken, in progress, failed, or passed)
  + User information
    - First name
    - Last name
    - Address
    - City
    - State
    - ZIP
    - Phone
    - Email
  + Driver notes
    - Lesson time
    - Start hour
    - End hour
    - Driver comments
  + Special needs
  + Driver photo
  + Student photo

## 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 web-based system needs to run in the most popular browsers (Edge, Firefox, Safari).
* A mobile version of the site needs to be developed for mobile users of these same browsers.
* There should be a low latency between user inputs and responses served; less than one second delay unless processing payments (in which case: sleep 5000 ms).
* Updates made to the site should roll out immediately. e.g. if the owner wishes to disable certain learning content, those courses or materials should be locked immediately, refreshing the clientside browsers, so that no users can linger on unavailable materials in their current browser session.

#### 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 servers hosting the site should run Unix, optimized for functionality over user experience.
* The browsers navigating to the site may be run on Windows, Linux, Mac OS, iOS, Android…; the site should function properly for the user on each platform.
* The backend should connect to and update site materials via the DMV; this will require server-to-server communications and possible DMV database access.
* Our system will require databases for the user list, driver/car list, content library, and appointments/scheduling system.

#### 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 site users will be assigned roles (admin, IT, secretary, student).
* Passwords will be case-sensitive. Usernames will not.
* The principle of least privilege will be utilized.
* Admins should be notified when the site connectivity goes down.
* There should be a reporting feature for users to report site issues in case automated surveillance fails to catch performance issues.

#### 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?*

* Users should not be hard-coded, such that queries to the user database are sufficient to add, remove, or update users.
  + These queries should be generated when users make changes to their own accounts or they may be made manually by IT, admin, or secretary on behalf of any user.
* The IT admin has typical administrator permissions as well as access to the back-end systems and source code.
* The system should backup its own data every 15 minutes and save backups on the timescale of a month, so as to prevent an inability to revert to an older version in case of a faulty update.
* Admins should be notified when a system update breaks the site.

#### 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 authenticate via password.
* Passwords stored will be encrypted and transmitted securely (https, no URL leaking, etc.)
* Accounts will require email at registration for account recovery and security alert purposes.
* Third login attempts should lock accounts and inform users via email.
* A password reset will require navigation via an email sent to the registered address.
* Login attempt logs should be generated and kept by the system.

### 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 register and store new users and their information.
* The system shall handle login validation via user credentials.
* The system shall host, handle, and administer multimedia learning materials, classes, and practice tests.
* The system shall track (and display) user academic/testing progress.
* The system shall handle appointments, scheduling.
* The system shall handle secure checkout and payments.
* The system shall exhibit connectivity to the DMV (so as to procure the latest, most accurate driver testing standards).
* The system shall utilize full database connectivity and functionality. Database(s) will handle users, drivers, materials, and the schedule.
* The system shall report and log its own behavior for IT/admin.

### 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 shall be consistent. Color palette, symbolism, fonts, design, etc. should be graphically unified.
* The UI should contain a navigation bar present (and consistent) on all pages. Options will include at least: home, user profile, cart, learning materials, help, contact us.
* The UI should provide feedback and dialogues to the user, especially at the termination of a successful action (“test complete”, “order complete”, “account deleted”, etc.)
* The UI should disable interaction with fields or options irrelevant to a given user.
* The UI should allow for incomplete or faulty forms to be refilled at only the points of error, saving the rest of the query for the user’s subsequent attempts to correct.

### 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?*

* Users have internet access.
* Users are running common operating systems and browsers.
* Users may be using mobile devices or PCs.
* Users have personal email addresses.

### 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?*

* The site has only one IT admin.
* The site’s storage and memory usage will be confined to the capabilities of the server utilized, with cloud-based options imposing budgetary constraints on the company over hardware constraints.
* The site is to be built on the timescale of months.
* The performance of the site at each endpoint may be limited by the connectivity strength of the user accessing it.

### Gantt Chart

