# CS 255 Business Requirements Document Template

## 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?*

* Client: DriverPass
* Train students to pass driving tests.
* “ DriverPass wants to provide students with access to online practice exams and on-the-road training to better prepare them for driving tests”

### 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?*

* DriverPass wants the system to utilize ‘packages’ to teach students to drive before they take their drivers test, similar to a driver’s ed course in school, but paid.
* DriverPass wants to improve the pass rate of students for their driver’s test.
* Components:
  + Admin user to update information on students and on packages offered and their prices. Secured with username and password
  + Student user, allowed to purchase packages, and schedule driving sessions. Secured with username and password.
  + Page detailing packages and their differences as well as their prices.
  + Page detailing information on the different driving instructors
  + Contact details for problems with the site.

### 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?*

* Access data from anywhere as long as internet access is available
* Reservations made in the program
  + Customize reservation bundles (to be added later)

## 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 clients stated a preference for web-based rather than application. The system would need to run in standard browsers (Chromium-based and Firefox), as well as on the phone-based web browsers.

#### 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?*

* Database should use MySQL with python, mySQL is best for database construction and management.
* A web host server could be Nginx.
* The website code could be PHP with Python and Javascript. PHP for framework, javascript for customization of webpage design, and python to integrate mySQL.
* Linux Ubuntu could also be used for hosting the web platform.

#### 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?*

* Three different types of users: Teachers, Students, and Admin
  + Teachers will have to enter educator code and be verified by admin before profile is labeled as ‘teacher’ type
  + Student is basic user type, and automatically labeled. So when a new user creates a profile they will be assumed to be a student-type user unless Admin changes the case (either human admin user, or the computer system if the user has a pre-approved educator code).
  + Admin is the strictest type with one user named admin and other admin users being modified by the original admin.
* The input will be case-sensitive in the case of passwords specifically.
  + Usernames will not be case-sensitive
* Users should be able to inform admin of a problem manually through a contact us link on the main page and login page. The system will inform the admin of a problem in the case that there is an error in the programming (a try-catch loop that will inform the admin) or if a new teacher user is created so that the user can be verified.

#### 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?*

* User changes can be made without directly changing the code, but rather interacting with the program (i.e. going to a delete user function available on the users account that will remove information from the database and replace it with null data)
* Unless a platform update directly affects the code the system should run fine.
* The IT admin should have direct access to the code in all programs, as well as the ability to delete and modify user data.

#### 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?*

* User needs username and password to login
* End-to-End decryption and HTTPS can both be utilized to secure data exchange
* To prevent ‘brute force’ hacking a limit of 15 tries can be used before account is locked behind a two-step encryption process
* If a user forgets their password, they can click a forgot password button which generates a temporary password that is sent to the email associated with their account. If they did not request the new password then it should be recommended that they change their password.

### 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 contain a database of users categorized by user type (Admin, Student, Teacher) and containing information based on user type.
  + Admin (username, password, admin capabilities)
  + Student (username, password, packages signed up for, payment information, billing address, personal information)
  + Teacher (username, password, packages offered, personal information)
* The system shall allow student-type users to purchase packages and communicate with the teachers that teach that package type, as well as view teacher-type users and their credentials.
* The system shall allow teacher-type users to change the packages they offer, as well as update their resume to better reflect knowledge and experience.
* The system shall allow admins to edit user information, change package data, and change user type.

### 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.)?*

* Interface needs to display user information (screen one, profile information), package information (screen two, available packages), cart and cart price (screen three, user cart and pricing. Plus payment information and checkout information), teachers and packages offered (screen four, teacher-user list, can be filtered by package or sorted by distance)
* User types are Teacher, Student, and ADMIN
  + Teacher-users should be able to list packages that they can offer, as well as how many spaces they have available for each package as well as dates available to teach. Teachers should be able to update their profile information
  + Student-users should be able to update their profile information, browse teachers and packages, and purchase packages, select teacher for package, and select dates and times for lessons
  + Admin-users should be able to update package types and user information.

### 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?*

* Did not address usage of different payment platforms. Did not address whether to use two-step verification for all users.
* I’m assuming that all users have a reliable email address, and that no one will need more than fifteen tries to remember their own password.

### 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?*

* I’ve chosen mostly open source engines to help with any budget limitations, as well as to allow a wider stream of knowledge. Since more people use open source, it will be easier to reference if there are any issues.
* The system would be limited to browser users only, as no app format was ever specified
* Also the fact that the time limit is a few months, if anything happens to delay the project, it could throw everything out of whack.

### 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.*

