# Victor Jarvis

# CS 255 Business Requirements Document

## 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?*

* The client is DriverPass, who would like to provide more training via online and on-road practice so that more people pass their driving tests, and they need a system that can handle these objectives. The system should facilitate purchases of lessons and reservations of lesson times, as well as host the respective interfaces for customers and employees, with various tasks handled so that they can focus on the business side of things versus the technical side

### 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?*

* The problem is that too many people are failing their driving tests.
* The customers need to be able to study and practice, and the employees need to be able to provide assistance in the processes required to set up lessons and profiles.
* Frontend: Internal interface for employees to handle reservations and customer info, and customer facing interface that can be used to make reservations and take practice lessons.
* Backend: Databases for customer information, including lesson progress, reservation and session information, availability of drivers, and package availability.

### 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 on and offline, edit data while online, manage roles and permissions for the internal processes of employees, tracking of changes
* Employees save reservations for customers, and customers make reservations for themselves online, depending on what is available
* The system has to handle multiple types of packages, which can be disabled if they aren’t currently available.
* Customers and employees should be able to reset their passwords on the respective interface
* System running on the web, preferably on the cloud
* Receive a notification from the DMV if policies or requirements change so that they can stay up to date

## 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 system needs to be able to run in a web and application based environment. The system should be able to handle varying workloads, including and especially increases in workload to accommodate new users and additional features.

#### 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 system should be able to run on all platforms to account for different students with different access to different resources. Training may be required for instructors and admins as systems add or remove features. The backend will require a database to hold student, instructor, and admin information, as well as student records and resources.

#### 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?*

* All users will be distinguished with unique ID, as well as a specific variable identifying them as a Student, Instructor, or Administrator. The administrator should be informed of major system errors and reports of any bugs. All error handling should be established and given proper tags in order to sort problems by type and severity.

#### 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?*

* All users should be able to have their information updated within the database as a means of keeping all information current. Follow basic CRUD processes to keep information updated without unnecessary code changes. IT should have access to most major system components, with proper security measures in place to distinguish admin as such.

#### 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 should have a valid student, instructor, or administrator email address, as well as a unique password subject to requirements to prevent unauthorized access. If a user forgets their password, or if too many attempts are made incorrectly, then the user will have to reset it. Major brute force attempts should trigger a report to IT administrators immediately.

### 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 allow students to register with student emails and set their account up.
* The system shall display student grades and progress in coursework when requested by the student.
* The system shall maintain information in cloud storage to be accessed on and offline.
* The system shall allow access based on permission and account type.
* The system shall offer multiple packages for students to choose from.

### 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 should be simple and relatively easy for users to navigate on a regular basis, with any changes made being fully explained visually within a notice or the interface itself. Students, instructors, and administrators should all be able to navigate the interface and access their respective resources easily. The Students should be able to access coursework and additional resources, as well as check their grades and reach out to IT support or academic support. Instructors can assign, grade, and change their coursework provided to students. They should also be able to add resources and view the students they are grading or the appointments they may have scheduled that day.

### 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?*

* The assumptions are mainly focused on the management of resources by the administrators and the instructors. The instructors are going to require a way to manage coursework and informational resources supplied to students, and admins will require a method of viewing database information and support resources.

### 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?*

* Some limitations are largely due to the variety of users who may either learn in different methods at different paces, curriculums that may change over time, the need for appointment management, and the variations in hardware that may be used to access the application.

### 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.*

