# CS 255 Business Requirements

**D.D. Sessions**

## System Components and Design

### Purpose

The purpose of this system design is to understand the client and what they want their system to be able to do.

* The client for this project is DriverPass, and they want us to design a system that allows students to better prepare for driving tests. The system needs to provide students with access to online practice exams and on-the-road training.

### System Background

The background of the proposed system is provided to understand what different components are needed for this system and what problem the client wants to fix.

* DriverPass sees a need for better training for people learning to drive. Many people fail their driving tests at the DMV, and the client wants to be able to provide practice tests online. They will also offer on-the-road training if the student is interested.
* Components needed for this system:
  + Access data online from any computer or mobile device
  + Download reports and data to use offline in Excel
  + Set up different users and roles with certain permissions
  + Access activity reports that track all user entries and modifications
  + Online accounts for students
  + Students can make reservations for a driving lesson online using their account
  + Employees can set up reservations if the student calls or visits the DriverPass office
  + Track which student is matched up with a certain employee driver, time, and car

### Objectives and Goals

The objectives in a system analysis offer measurable tasks that need to be included in the system design to achieve the functioning program that meets the client’s needs. The goals provide an outline of what needs to be implemented to create the final product.

* Disable/enable reservation packages (ability to add and remove packages in the future)
* Add a new student (either in-office or online):
  + First name, last name, address, phone number, state, credit card details (number, expiration date, and security code)
  + Student pick-up and drop-off location (these should be the same)
* Users and roles:

|  |  |
| --- | --- |
| **Users** | **Permissions** |
| Owner | View and download reports as Excel spreadsheets; reset passwords; block user access; maintain and modify the system; add users; make reservations; disable and enable reservation package options; receive notifications when DMV updates rules, policies, or sample questions |
| IT Officer | Maintain and modify the system; reset passwords; block user access; add users; disable and enable reservation package options; receive notifications when DMV updates rules, policies, or sample questions |
| Secretary | Add new students, make reservations |
| Student | Create online account; make, cancel, and modify reservations; view scheduled driving sessions (completed and upcoming); if applicable, view online class material and take practice tests; reset password |
| \*Driver | \*Need more info. Do drivers need to be set up as users to access the reservations they are associated with? |

* Reserve an appointment package:
  + Ten cars and ten driver employees
  + Each driving session is two hours
  + Choose an available day and time for each driving session
  + Match an available driver employee to the driving session

|  |  |  |
| --- | --- | --- |
| **Package** | **Package Details** | **System Functionality** |
| One | 1. Six hours in a car with a trainer | 1. Show available dates and times on a calendar in two-hour driving sessions 2. Available dates are based on driver employee availability 3. Schedule up to three sessions 4. Reserve the selected time for the student and update the driver availability |
| Two | 1. Eight hours in a car with a trainer 2. In-person lesson to explain DMV rules | 1. Show available dates and times on a calendar in two-hour driving sessions 2. Available dates are based on driver employee availability 3. Schedule up to four sessions 4. Reserve the selected time for the student and update the driver availability 5. Schedule the in-person lesson for DMV rules \*need more info |
| Three | 1. Twelve hours in a car with a trainer 2. In-person lesson to explain DMV rules 3. Access to online class and practice tests | 1. Show available dates and times on a calendar in two-hour driving sessions 2. Available dates are based on driver employee availability 3. Schedule up to six sessions 4. Reserve the selected time for the student and update the driver availability 5. Schedule the in-person lesson for DMV rules \*need more info 6. \*Class and test accessibility will be addressed in User Interface |

## Requirements

### Nonfunctional Requirements

This section will detail the different nonfunctional requirements for the DriverPass system by thinking about the different things that the system needs to function properly. The system will need to be reliable, efficient, secure, usable, scalable, and maintainable.

#### Performance Requirements

* The DriverPass system will need to run in a web-based cloud environment. The purpose of this program is to provide students with access to online practice exams, so it should be accessible anytime from anywhere. The owner needs the ability to download reports and data to use offline in Excel.
* The system will need to perform at a high speed to navigate easily through the practice exams as well as the other pages within the application.
* This program will need to be updated frequently. The owner will need to be notified any time the DMV updates their rules, policies, and sample questions. These changes will need to be implemented into the program as they arise.

#### Platform Constraints

* The program should be compatible with multiple operating systems such as Windows, Mac, iOS, and Android. I would recommend a platform such as .NET MAUI which uses the latest technologies for building apps that are compatible with all these operating systems.
* The back end will require tools to support this application. A database such as SQL is necessary to store the customer and employee information, payment details, practice exam scores, etc. The C# programming language would be a great choice to use with the .NET MAUI framework.

#### Accuracy and Precision

* To distinguish between different users, everyone will need a unique username. The user will be able to enter any variation of upper or lowercase characters, but the input will first be converted to all lowercase before performing the check to see if it already exists. This will keep usernames consistent and prevent confusing similar inputs such as “Alpha” and “alpha”. This information will be stored in the database.
* All errors should be reported to the admin team and stored. This could include security breaches, system failures, data corruption, or anything else that impacts the functionality of the application. They want to focus on running the business with minimal technical problems, but they should still be notified if any issues come up.

#### Adaptability

* It will be possible to make changes to DriverPass users (such as add, remove, or modify) without changing the code since we are building a custom software application with an object-oriented approach.
* The system will be adaptable to platform updates, but the ease of making changes will depend on the updates being released. Some changes may include updating the database schema, adjusting the code to work with new libraries, or even making changes to the system architecture.
* IT admin will need access to the codebase and database to maintain and modify the system, reset passwords, block user access, add users, and disable and enable reservation package options. He will need to access information such as login credentials, system logs, and error reports.

#### Security

* For the user to log in, he or she will need to create a unique username and a secure password. It would be good practice to choose a password with at least ten characters that uses a mixture of uppercase and lowercase letters, numbers, and special characters.
* To secure the connection or the data exchange between the client and the server, you should use HTTPS to ensure data confidentiality. Username and password authentication should be utilized as well as SSL certificates to verify the server identity. I would recommend using SSH protocol to establish a secure connection and pair that with SFTP to secure file transfers.
* The administrators will need to set a maximum number of log in attempts for the user. This will help prevent brute force hacking attempts. The account should be locked after a certain number of failed attempts.
* If a user forgets his or her password, it can be reset by answering security questions.

### Functional Requirements

This section will detail the different functions the system needs to provide. The system will need role assignments and adjustable user profiles, an intuitive user interface, smooth data integration, effective content management, responsive mobile learning support, and flexible online testing options.

* The system shall provide user registration and login functionality.
* The system shall provide online classes and practice exam questions and answers.
* The system shall provide exam progress functionality for not taken, in progress, failed, or passed.
* The system shall provide scheduling services for on-the-road training.
* The system shall provide user roles and permissions.
* The system shall provide a tracking functionality for activity reports.
* The system shall provide contact functionality to the office and to the student.

### User Interface

This section will detail the needs of the interface, the different users for the interface, what each user will need to do through the interface, and how the user will interact with the interface.

* The interface will need to be user-friendly, clear and concise, interactive, and offer easy navigation.
* The different users for the interface will be students, owner, IT officer, secretary, and drivers.
* The owner will need to view and download reports as Excel spreadsheets; reset passwords; block user access; maintain and modify the system; add users; make reservations; disable and enable reservation package options; and receive notifications when the DMV updates rules, policies, or sample questions.
* The IT officer will need to maintain and modify the system; reset passwords; block user access; add users; disable and enable reservation package options; and receive notifications when the DMV updates rules, policies, or sample questions.
* The secretary will need to add new students and make reservations.
* The student will need to create an online account; make, cancel, and modify reservations; view scheduled driving sessions (completed and upcoming); if applicable, view online class material and take practice tests; and reset his or her password.
* The driver will need to access his or her reservation schedule, monitor the students’ progress, and provide feedback to the students.
* All users will need to interact with the interface by mobile, web browser, or any other device that has an internet connection.

### Assumptions

This section will detail things that were not specifically addressed in the design above and address assumptions that I am making in my design about the users or the technology they have.

* I am assuming that the program will be developed with C# in a .NET MAUI framework and a SQL database.
* I am assuming that the users will have access to a device with up-to-date web browsers.
* The theme of the application is to be determined. No logo or color scheme has been provided yet.
* I am assuming that administrators can add students, make reservations, and monitor student progress. Other permissions depend on the level of administrator.
* I am assuming that students can submit work, view content, and measure progress.

### Limitations

This section will detail any limitations in the system design or in resources, time, budget, or technology.

* Our timeline to complete this project is sixteen weeks. Time could be a limitation when developing this system due to having a small team of five.
* Costs will need to be kept as minimal as possible.
* The system will need to be updated in the future to implement adding or removing modules.

### Gantt Chart

Below is the Gantt chart I created to offer a visualization of the project schedule. Tasks are on the vertical axis and time intervals are on the horizontal axis. The tasks are color coded depending on the phase of the development lifecycle.

A screenshot of a graph

Description automatically generated