# CS 255 – Project One - Business Requirements Document

Jason Barry

Professor Venckus

CS-255-11329-M01 System Analysis and Design

5 June 2024

## System Components and Design

### Purpose

* Our client, *DriverPass*, seeks out assistance in developing a system that is capable of providing students with access to online practice exams as well as on-the-road training to better prepare them for driving tests. The developed system should be able to be accessed online, preferably over the cloud, from any computer or mobile devices. While online, the user should be able to download reports and other information for offline use.

### System Background

* Because many people fail their driving tests, *DriverPass* was created to provide easy access to online classes, practice tests, and on-the-road training. The company specifically seeks to aid students. To accomplish these goals the client has tasked us with creating a system that includes the following components:
  + The system should be able to be accessed online, from any computer or mobile device.
  + Once online, users should be able to create a new account.
  + When an account is created, customers should be able to schedule, modify, and cancel appointments based on one of the passages offered by *DriverPass.*
  + All appointment modifications should be tracked to avoid overlap.
  + The developed system’s UI should be easy to understand and follow the client’s recommendation.
  + For security purposes, the developed system should limit what a user can accomplish based on the type of user.
  + The developed system should possess the ability to be connected to the DMV.

### Objectives and Goals

* The developed system should be able to be accessed online, preferably over the cloud, from any computer or mobile device.
  + While online, the user should be able to download reports and other information for offline use.
* Customers should be able to create a new account.
  + Whenever a customer creates an account, they should be brought to a registration screen that asks for the user’s first name, last name, address, phone number, state, and their credit card number, expiration date, and security code. Furthermore, at registration, each customer should have the ability to include the pickup location from where the customer wants to be picked up as well as the drop-off location, which should be the same as the pickup location.
* When creating an appointment, there should be three packages offered:
  + (1) Six hours in a car with a trainer.
  + (2) Eight hours in a car with a trainer and an in-person lesson where we explain the DMV rules and policies,
  + (3) Twelve hours in a car with a trainer, an in-person lesson where we explain the DMV rules and policies—plus access to our online class with all the content and material. The online class also includes practice tests.
* All appointment modifications should be tracked to avoid overlap
  + The system should be able to track which driver and car that the customer is scheduled to practice alongside as well as the time of the appointment.
* The UI interface should be easy to understand and follow the client’s recommendation.
  + On the top left, there should be a box containing the user’s online test progress. This should include the test name, time taken, score, and status).
  + Underneath the test progress box, there should be a box containing all driver’s notes.
    - This box should be included in a table that displays when their driving sessions were as well as only notes the driver left them.
  + On the top right, there should be a box containing the customer’s information
    - This showcases the data that the customer provides during registration.
  + Underneath the box containing the customer’s information, there should be a box showcasing the driver’s photo as well as their student photo to the right.
* For security purposes, the developed system should limit what a user can accomplish based on the type of user.
  + The *system administrator* should possess the ability to reset accounts (in case someone forgets their password) and/or block a user’s access to their account.
  + The IT officer user role should be able to maintain and modify the system
  + The secretary user role should be able to answer calls and schedule appointments.
  + The customer user role should allow them to schedule, cancel, and modify appointments.
    - The system administrator and the IT officer user roles should possess the ability to disable specific packages at any time.
* The developed system should possess the ability to be connected to the DMV.
  + This ensures the system may be updated to accommodate any new rules, policies, or sample questions.

## Requirements

### Nonfunctional Requirements

* The developed system should be easy to use and understand
  + The system should adhere to the guidelines provided by the client.
* The system should run smoothly without any noticeable loss in performance. Performing basic tasks like scheduling appointments should be nigh-instantaneous.
* The system should adhere to relevant privacy and security regulations including the ISO 27001. This certification helps “…secure software development by increasing an organization’s ability to protect [the] confidentiality, integrity, and availability of critical business information” ([*10 Best Practices for Software Development Security*](https://www.orientsoftware.com/blog/software-development-security/)).
* The system should be guarded from outside threats by locking access behind a case-sensitive username and password.
  + Further protections to prevent common vulnerabilities should also be addressed.
* The system should be hosted on a web server that is compatible with major operating systems (like *Windows, Linux,* and *MacOS)* as well as provide support for most modern web browsers (*Chrome, Firefox, Safari*, and *Edge).*
* The probability that the system will crash or produce errors should be minimized.

#### Performance Requirements

* The developed system should be accessed online, preferably over the cloud, from any computer or mobile device. This should work regardless of the operating system (including *Windows, Linux, MacOS,* and *Android).*
* The system should suffer little to no performance issues and each request should be returned with minimal latency.
* The system should be updated whenever the DMV changes its rules, policies, and/or sample questions.

#### Platform Constraints

* The developed system should be able to be accessed online, preferably over the cloud, from any computer or mobile device. The experience should be the same for all users regardless of their platform of choice.
* The developed system should have access to a database that stores the date and time of all appointments of each user. Using this database should ensure appointments don’t overlap with one another.
* There should be back-end support that allows the system to be connected to the DMV. This should ensure the system is updated to accommodate any new rules, policies, or sample questions.

#### Accuracy and Precision

* Users should be distinguished based on their predefined user role. These user roles should provide the necessary actions and information that the user can and cannot perform.
* To ensure query parameterization (see the Security section), all inputs should be tested for case sensitivity and whether they meet certain requirements (for example, they must include a number).
* The system should inform the admin whenever the user enters invalid information three times in a row. At this point, the user should be locked out of the system until they reset they’re password or are granted access by the *System Administrator*.
  + This should combat all brute-force attacks.

#### Adaptability

* In future iterations of the system, the client wants to be able to customize the packages that they offer. Without changes to the code, the client should be able to remove, modify, or add new packages.
* Without changes to the code, the system should be able to update rules, policies, or sample questions whenever changes are made by the DMV.
* The client seeks to focus on running their business with minimal technical problems. Therefore, the *IT Officer* and the *System Admin* should only perform minor actions. These actions include disabling specific packages and making slight modifications to the system. However, neither the *IT Officer* nor the *System Admin* canadd or remove entire modules.

#### Security

* Whenever a user logs into the system, their credentials should be validated. Users should be encouraged to make one if they do not have an account. However, if the user inputted the wrong information, they should be asked once more to provide the correct information. If the information is still incorrect, the *System Administrator* should be informed and the user’s password should be reset (an email should be sent to the user providing the steps necessary to create a new password).
  + The email to the user should be sent automatically. Once the user completes the requested query and successfully resets their password, they should automatically be granted access to the system.
* The most common point of vulnerability that the system may encounter is through injection attacks. These injection attacks could allow external threats to access and harm the system. Luckily, injection attacks can be avoided by utilizing query parameterization. Therefore, all queries must be tested to ensure proper protocols are in place.
* Another route in which external threats can harm a system is through hijacking connections or data exchanges between the client and the server. Therefore, all communications should be encrypted.
* For security purposes, the developed system should limit what a user can accomplish based on the type of user.
  + The *System Administrator* should possess the ability to reset accounts (in case someone forgets their password) and/or block a user’s access to their account.
  + The *IT Officer* user role should be able to maintain and modify the system
  + The *Secretary* user role should be able to answer calls and schedule appointments.
  + The *Customer* user role should allow them to schedule, cancel, and modify appointments.
    - The *System Administrator* and the *IT Officer* user roles should possess the ability to disable specific packages at any time.

### Functional Requirements

* The system shall allow users to download reports and other information using external applications like *Excel* for offline use.
* The system shall grant Customers the ability to create a new account.
  + Whenever a customer creates an account, they should be brought to a registration screen that asks for the user’s first name, last name, address, phone number, state, credit card number, expiration date, and security code. Furthermore, at registration, each customer should have the ability to include the pickup location from where the customer wants to be picked up as well as the drop-off location, which should be the same as the pickup location.
* The system shall validate a user’s credentials whenever the user logs in.
* The system shall offer one of three packages whenever an appointment is created:
  + (1) Six hours in a car with a trainer.
  + (2) Eight hours in a car with a trainer and an in-person lesson where we explain the DMV rules and policies,
  + (3) Twelve hours in a car with a trainer, an in-person lesson where we explain the DMV rules and policies—plus access to our online class with all the content and material. The online class also includes practice tests.
* The developed system shall track all appointment modifications (via a database) to avoid overlap. Furthermore, this database shall allow the system to track which driver and car are available whenever the customer is scheduled to practice.
  + This information should be presented clearly and concisely. Therefore, in case something goes wrong, the client can print activity reports and determine who is responsible for any issues.
* The system’s UI interface shall be easy to understand and follow the client’s recommendation.
* For security purposes, the developed system shall limit what a user can accomplish based on the type of user.
* Connections to the external DMV application shall be integrated into the Developed System.
* Whenever an appointment is modified (including when it’s created and/or removed), both the *Customer* and the *System Administrator* shall be notified.

### User Interface

* The different users of the system include the *System Administrator,* the *IT Officer,* the *Secretary,* and the *Customer.*
  + The *System Administrator* role has full access to all accounts. This allows them to reset passwords in case a user forgets it. Access to all accounts should also allow the *System Administrator* role to block a user’s access to their account.
  + The *IT Officer* roleis responsible for maintaining and modifying the system.
  + The *Secretary* role can answer phone calls as well as make, cancel, and modify appointments.
  + The *Customer* role can solely create, cancel, and modify appointments.
* The user should be able to access the system online, from any computer or mobile device.
* The UI interface should be easy to understand and follow the client’s recommendation.
  + On the top left, there should be a box containing the user’s online test progress. This should include the test name, time taken, score, and status).
  + Underneath the test progress box, there should be a box containing all driver’s notes.
    - This box should be included in a table that displays when their driving sessions were as well as only notes the driver left them.
  + On the top right, there should be a box containing the customer’s information
    - This showcases the data that the customer provides during registration.
  + Underneath the box containing the customer’s information, there should be a box showcasing the driver’s photo as well as their student photo to the right.

### Assumptions

* The design assumes that the user has an internet connection whenever they modify and update information.
* The design assumes that the electricity requirement necessary to run or access the system is met.
* The design assumes that the user is utilizing a device that runs on a modern operating system. Theoretically, if the user used an older and/or niche operating system, the system would not function properly.

### Limitations

* The data cannot be modified or updated unless the user possesses a valid internet connection. This avoids producing data on different servers which could cause data redundancy.
* As the system is accessed via an electronic device, electricity is a requirement for the system to work.
* Users cannot add or remove modules from the system. To accomplish these tasks, a developer or system analyst is required.

### Gantt Chart

A diagram with multiple colored boxes

Description automatically generated with medium confidence

**Works Cited**

*10 Best Practices for Software Development Security*. Top Software Outsourcing Company in Vietnam - Orient Software. (n.d.). <https://www.orientsoftware.com/blog/software-development-security/>

*CS 255 DriverPass Interview Transcript*

Manico, J., & Detlefsen, A. (n.d.). *Iron-clad Java*. O’Reilly Online Learning. <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch07.html#ch07lev1sec3>

Valacich, J. S., & George, J. F. . (2023). *Modern Systems Analysis and Design* (10th ed.). Pearson Education Limited.

Wiley. (n.d.). *Systems Analysis and Design with UML, 4th Edition*. O’Reilly Online Learning. <https://learning.oreilly.com/library/view/systems-analysis-and/9781118037423/>