# CS 255 Business Requirements Document Project 1

Harvey Corey

CS-255-R4784 System Analysis and Design 24EW4

Professor Federico Bermudez

April 13th, 2024

## 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 purpose of this project is to line out the details and requirements of the DriverPass online system so both engineers and the owner are satisfied. This will reduce confusion, increase satisfaction with the final product, reduce unnecessary costs and help both companies reputations. The client is Liam who is the DriverPass owner.

### 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?*

* “Liam is hoping to take advantage of a void in the market when it comes to training students for the driving test at their local department of motor vehicles (DMV).”
* They want to create an online presence for training students how to drive. As is explained, there isn’t much on the market for such an environment.
* The driverpass system will create an online platform for students to study with provided materials, take tests and schedule appointments with an online system. It will allow administrators to modify and maintain DriverPass by managing schedules and updating or changing online materials along with managing customer profiles. It will also provide up to date information about changes to laws and regulations according to the DMV. It also includes a notification system for related information.

### 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?*

* It should be able to manage users and their access to elements of DriverPass in a hierarchal system.
* It should allow admins to make changes to DriverPass and manage schedules.
* It should allow users to schedule driving appointments.
* It should allow course materials to be managed in a user friendly way.
* It should create notifications for DMV changes and notifications for users.
* It should host study materials and user data only for those who have authorization.
* It should allow access to files anywhere with an internet connection and avoid data redundancy.
* Automatic password reset functionality along with the password system in general.
* The ability to track user changes and information, especially in terms of auditing.

## 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.*

* Security. Data encryption and user based security clearance.
* Accessibility. DriverPass should be accessible anywhere with internet access.
* Performance. The system should load requests quickly, especially under load.
* Reliability. Minimal downtime is nice for business. Backups will be needed.
* Scalability. This is for the growing userbase.
* Compliance. Up to date DMV requirements and changes are a must to follow.
* User friendly interface. This also goes for admins who need to change course materials.
* Maintenance friendly. Admins should be able to manage the site without downtime.
* User logging. This is for audits and more. This can be useful for admins as well.

#### 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 environment should run on as many web browsers as possible. Being a web browser based application will ensure that there is no installation or large download for the users. This should improve the user experience.
* In terms of speeds, the system should load fast for each user but the data transfer should be throttled based on the amount of users on DriverPass to ensure that the system resources are used appropriately when under load.
* Updates are not static but instead dynamic based on a number of possibilities. One would be changes in regulations from the DMV. Another could be changes in how up to date the code or tools are. An example would be how sites that used Adobe Flash player needed to make a change away from Flash as it got retired.

#### 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?*

* Another advantage of a web browser based DriverPass is that it has high compatibility on each of the operating systems. The main compatibility comes from each of the web browsers. The operating systems are not necessarily always the main concern due to this. DriverPass needs to just be tested on web browsers and historical versions of those browsers to ensure compatibility for as many users as possible.
* A database will certainly be needed. Without a database you will have no easy way to store user data, or even historical user data in an efficient manner. Plugins were also mentioned in an earlier assignment. Scalability and independence for these plugins will be paramount. The plugins need to be modular and non-invasive always.

#### 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 users should be created and have access based on a hierarchical system. This is important to protect the website and the sensitive information on it. You have Customers who can add and change only their scheduling, payment information, and course progression. You have IT Officer, who can make changes to the website in all cases. There is Secretary, who can modify schedules by adding and removing times. There are Instructors who can use DriverPass to see schedules and get notifications for cancellations.
* Case sensitivity is important for passwords and usernames. This allows for more options with usernames. The database will keep a list of all usernames and prevent username redundancy errors when signing up. Don’t create the user if the username is already in the database, instead offer a random alternative like numbers at the end of the username or make the user change it themselves. Case sensitivity helps with this issue.
* Admins should be notified of customer support tickets. They should also be notified of downtime or system failure immediately and on high priority this could be in the form of a phone call or text message.

#### 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?*

* Yes to the first question without a doubt. This is very important. Adding and removing users should be code that directly communicates to the database’s information. No user should truly be removed from the database, only listed as retired for historical purposes. The username should be regenerated in a way that follows a retired format specifically so that username can be taken by a new user. An example would be “Steve !Retired!0001” instead of “Steve”. A timestamp of the user created date and the retirement date should be added.
* The system should stay up no matter what updates are being made. Realistically, having DriverPass on multiple servers would be ideal so while one is being modified, the others will stay up. When a new update is released, the servers with the new update will become active while the older ones go under maintenance. Juggling uptime this way will be important. A backup plan should be in order in case an update is not working as intended and needs to be reverted quickly.
* The IT administrator and assumed developer will need all access to the DriverPass system.

#### 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?*

* The user will need a web browser that is supported, they will also need a username and password. Data exchange should be encrypted. Only information relevant to the session and user should be sent.
* A brute force attempt can be circumvented with a captcha and by limiting the amount of attempts by forcing that IP, username and session to wait before another login attempt. If the user forgets a password, a reset link should be sent to the user’s email. Once the secure link is clicked, it will set the user up with an option to reset the 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 should provide an online class for driver training.
* The system should facilitate on the road training for users.
* The system should allow users to schedule appointments.
* The system should track user progress.
* The system should update everyone on DMV regulations and other changes.
* The system should have data privacy for all users.
* The system should provide good performance even under load.
* The system should be reliable and without downtime.
* The system should be easy to maintain and update.
* The system should log all user activity.
* The system should be scalable with a number of systems.

### 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 should accommodate as many browsers as possible. This does include mobile browsers. Users will be able to track progress, change payment information, change profile and personal information, study and view course materials, schedule appointments and take tests.
* The user interface needs to follow the layout drawn in the interview. The mobile layout should absolutely be centered around phone quirks, like touch screen interactions. Mobile will need larger buttons for a smaller screen. Another interview may be needed for designing a mobile UI.

### 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?*

* One large assumption that could possibly cause problems is the IT Officer role. It is assumed that they will be maintaining the code and systems of DriverPass.
* Another assumption is that the hardware has been covered and is a non-issue. Changes to DriverPass may be needed to accommodate the available hardware.
* One assumption is that DriverPass will be maintained for its lifetime instead of creating a new DriverPass. An example would be DriverPass 2. Even if DriverPass is retired, it will keep scalability, adaptability, and modularity in mind. Plug and play between all systems is key.

### 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?*

* DriverPass’s design should be created to be as small as possible. It needs modularity to support more hardware and software. It should easily be scaled up or down based on the availability of the hardware. It is easy to make large requirements for a better system, but this isn’t always the best case. Liam may not want to spend an arm and a leg on hardware. Keeping it as simple as possible is necessary. This also helps performance and load times.

### Gantt Chart notes

*The Gantt Chart will need reasonable due dates and times to create a responsible timeline for the team to create driverpass. The dates provided may not be suitable and are an estimate.*

### Hardware requirements

*What are the hardware requirements?*

* There will need to be a database and hardware to run it and provide fast load times.
* There will need to be a few servers to run DriverPass that will gain information off of the database.
* A drive rack will be needed with plenty of hardware to store all the information that will be accumulated and have excess.
* A backup system and another drive rack will be needed to maintain all this information. This could possibly need another database.
* The second database will be the backup in the even of a system failure.

### 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.*

A screenshot of a grid with a grid and a grid with a grid with a grid with a grid with a grid with a grid with a grid with a grid with a grid with a grid

Description automatically generated

# CS 255 Model Application Short Paper

Corey Harvey

Corey.Harvey@snhu.edu

Southern New Hampshire University

## Process Model Application

[How would you apply a process model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]

Getting a process model will consist of getting the stakeholder’s approval and ideas. An interview is necessary for gathering required information for the DriverPass system. After gathering what is required, we will begin by designing the system around those requirements and creating technical requirements to fill all needs. Hardware requirements and costs can be discussed with the stakeholders. Once all requirements are outlined, the team will start creating the system and implementing all the needs. This includes writing the code and obtaining the needed hardware. Testing will be needed after that. We need to ensure that that everything is working like planned. Once everything is tested, feedback is needed from the stakeholders to ensure everything is covered. Once that is accomplished, the system will need to be deployed in its final version 1.0 stage and should start accepting actual users.

## Object Model Application

[How would you apply an object model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]

Based on an Object Model Application, the first step would be to identify all needed objects. Once that is finished, we would need to identify all needed attributes. Once everything above is identified, we would need to map out all relationships of the model. Other things will be needed as well such as encapsulation, inheritance and polymorphism. These are important relationships and functions of the model.

Polymorphism is important for objects to be used in different contexts. Inherent relationships identify relationships and commonalities between objects. This will allow for hierarchy between all users as an example. Encapsulation allows for organization and for the systems to maintain a sort of body between systems. A car has wheels but the wheels are not a car.

## Process and Object Model Comparison

[What are the advantages of each model for the DriverPass scenario? What are the disadvantages of each model for the DriverPass scenario?]

Object modeling can be complicated and not something you may want to show to stakeholders. For that, you’ll need to use a more simple and clear process model. Just like the names imply, a process model shows the process of the system while the object model shows the actual objects or cogs in the system with all relationships. The object model is geared towards the engineers. Engineers can’t have too much information. Clarity is key for preventing assumptions and mistakes caused by those assumptions. It also reduces costs created by said mistakes. However, on contrast, a complicated object model without a good formula may cause over engineering. Over engineering may cause numerous bugs or load time issues. Simplicity is key, information is good. Object oriented programming is not for everyone and it may confuse stakeholders or developers alike.