**

***Jonathon Smith***

***Computer Science, Southern New Hampshire University***

***CS 255: System Analysis and Design***

***Professor Bryant***

***June 2, 2023***

## System Components and Design

### Purpose

Liam, the owner of DriverPass, has noticed a gap in the training of soon-to-be drivers that has caused many people to fail their driving tests at the DMV. DriverPass will offer online classes, practice tests as well as in-person driving instruction to their clients. Their clients are likely the parents of prospective drivers, but the users of the application will be the student drivers themselves.

### System Background

The System needs to allow for users to register for driving training, include multiple packages with various benefits, have multiple roles that have access to certain information, facilitate login/registration, include online classes and practice tests and more.

### Objectives and Goals

* User Login/Registration
* Driving lesson scheduling including creating, cancelling, and modifying lessons
* Offer multiple packages to clients
* Report generation varying by user role
* Test completion/progress monitored and displayed to user
* Updates with DMV requirements

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

DriverPass is looking for a web-based application. This will allow access from any device that has an internet connection. The system should be extremely fast as little to no processing is being done with data. The system needs to be updated whenever DMV has changes at minimum.

Since the userbase is going to be primarily teenagers, an App should also be suggested.

#### Platform Constraints

As a web-based cloud system, it should run on all platforms that have an updated web browser. A database is needed to store user information like name, address, payment info, reservations, test progress etc.

#### Accuracy and Precision

Various users will have a role and an ID number to distinguish them from each other and allow/disallow privileges. Inputs outside of passwords do not need to be case sensitive. Admins should be informed of problems during major events like malicious attacks, failure to communicate with the cloud, internet outages, etc.

#### Adaptability

The system should absolutely allow for changes to a user depending on the role. For example, a user role should only be able to make changes to themselves. An admin, however, should be able to make changes to all users. The IT admin needs access to all data. Assuming financial data is encrypted as well as passwords, the admin will need access to all other data to maintain the system and troubleshoot issues.

#### Security

After registration, a user will login with a username and password. The data between the client and server will be encrypted immediately after validation for security purposes. A “Brute Force” attack should have an account locked. Accounts can unlock after some time has passed and allow for login attempts again or a reset of a password. If a user forgets their password, a reset password email should be sent. Two factor authentication is also an option we should offer to all users.

### Functional Requirements

* The system shall validate user credentials when logging in.
* The system shall allow users to register and manage their information.
* The system shall enable scheduling, modifying, and cancelling driving lessons.
* The system shall track users' progress on driving tests.
* The system shall manage user roles and permissions.
* The system shall automatically alert admin in cases of catastrophic failure.
* The system shall allow a user to reset their password.
* The system shall lock out a user after 3 failed login attempts.
* The system shall automatically update to the newest DMV requirements.
* The system shall offer multiple packages/options for users to purchase.
* The system shall encrypt sensitive information before sending it to the server.

### User Interface

The interface needs to have a login/registration page. The home pages for various roles should differ. Users should see their upcoming appointments, test progress, information, special notes, and their instructor photo. Drivers will need to see their student photos, their schedule, where they are picking students up from, and be able to add notes to student’s home pages. Admins will need access to all information and have it organized in an easily digestible way. All users will interact with the system via a responsive web browser for now. An app will be discussed in a future meeting.

### Assumptions

Whether there will be an app or not has yet to be addressed. We are assuming that the user is online and has access to the internet when using the app. We are also assuming users have basic knowledge of web browsing and device peripherals.

### Limitations

The biggest limitation is the lack of an application. This type of service would benefit from having an app. It’s also crucial that the DMV has some sort of API for automatic updates or at least some way to be notified when requirements change. Depending on how that information is relayed, it could require some time for development of an update leading to issues with the accuracy of the system’s requirements in the meantime.

### Gantt ChartA screenshot of a graph Description automatically generated with low confidence

**Models**

**Process Model**

According to Modern Systems Analysis and Design, “Process modeling involves graphically representing the functions, or processes, that capture, manipulate, store, and distribute data between a system and its environment and between components within a system.” (Tegarden et al., 2013). This system would likely start with a user that has an option to either login or register. That information would then be sent to the database after verification and either create a new user or login the user.

If the user is a new user, we need to offer them various packages at various price points according to DriverPass specifications. Once a package is selected, the user will then be asked to pay for the product and be taken to the dashboard. Their information will be encrypted and sent to the database as well. Whether the user is a new user or not, both will see the dashboard at this point.

The dashboard will check the user role and permissions and display the corresponding UI. From here, Users will be able to take practice tests, schedule driving training or modify a scheduled training, as well as view their profile and change necessary information. Admins will have access to all accounts to make changes and handle issues as well as view reports. Instructors will be able to view their upcoming appointments.

Regarding process models for this system, we are looking at the functions that are going to take place as a user interacts with the website. Each path needs to be mapped out in detail, but we are not concerned with the objects in this model. We just know that a user will exist, but we aren’t concerned with how they are made just yet; simply what they can do and when they can do it.

**Object Model**

According to Margaret Rousse of Techopedia (2013), “An object model is a logical interface, software or system that is modeled through the use of object-oriented techniques. It enables the creation of an architectural software or system model prior to development or programming.” The object model graphically represents any interfaces in the system as well as how objects interact with each other. This could be via inheritance, encapsulation, etc.

For this system, we will have a user, but not all users have the same permissions. We can have a class of BaseUser that has attributes that all other users will have in common. An ID number, personal info, etc. Now we can have user, admin and instructor classes that inherit from the BaseUser class and have their own unique attributes. A regular user may have a subscription type or payment information, an instructor will have access to schedules and advanced access to changing schedules without needing authorization, and admins will need access to mostly everything on the site so they can handle issues that may arise and receive data that other roles should not see. There will be other objects in the system as well, like a driving lesson object with attributes to describe the student, instructor, when and where and so on.

Our object model is less concerned with what we are doing and more concerned with the structure of the objects and how these objects are related to each other. This is important because we need to know what attributes each object has so we can build out the functionality of the system and manipulate data in the way needed for an efficient system that fits the needs of DriverPass.

**Comparison**

Comparing these two models is difficult because they essentially have no redundancy between them. An object model breaks down our objects and how they are comprised. Our process models break down what those objects can do. They do not represent the same information. In fact, they need to be used simultaneously to understand the larger picture of how a system is supposed to work and how it is to be constructed.

The advantage of object models is they provide the structure needed for a system. A good object model will make for efficient development as developers should not write redundant code. For example, if attributes are shared between multiple classes, they can be inherited. It’s also important to note that if the structure of the system is created, then construction on the database and back end could begin without knowing what the objects do.

Conversely, a process model describes what the objects can do, and their functionality. While seeing this is very useful when trying to track the flow of data and how a user can interact with the system, we are missing the starting point of the data itself. To summarize, while both models have the advantage of describing their respective aspects of a system, they also share the same disadvantage that they do not represent a complete picture without the presence of other models.

**Resources**

Rouse, M. (2013, December 20). Object model. Techopedia. https://www.techopedia.com/definition/8635/object-model#:~:text=An%20object%20model%20is%20a,prior%20to%20development%20or%20programming.

Tegarden, D., Dennis, A., &amp; Wixom, B. H. (2013). System analysis design UML version 2.0: An object-oriented approach. Wiley.