# CS 255 Model Application Short Paper

Brett Nottmeier

Brett.Nottmeier@snhu.edu

Southern New Hampshire University

## Process Model Application

Having a process model made for the DriverPass application can be a great aid in software development. It can help us understand the system's flow and the functions we would need to define to process, manipulate, and store data. Having these processes defined can be a great benefit to the project and enable a deep understanding of the DriverPass application system and how it will be designed.

To define the process model, we will need to think about the tasks that are going to be performed in this model and the data that will be used to create this flow. Understanding the requirements set by DriverPass is key to this success and requires analysis and understanding of the system that needs to be created. After understanding the tasks that need to be performed for the project, we will need to think about the input and output of each task and the data that will be required. For example, when a user creates an account, we will need to think about the input necessary to create an account, their username and password, and the output would be an account creation that would be stored in the database for the user to log in. The conditions for each task should be thought out when the user logs in. They should be required to input the unique username and password that the account creation specified. All conditions for each task need to be looked at to create the program's flow. Lastly, after understanding all the requirements, tasks, and conditions, we can lay out the flow and sequence of each task, create the process model for DriverPass, and understand the systems and flow that will be needed for the project.

## Object Model Application

Having an object model in the design of the DriverPass system can be helpful and have a visual representation of a system's object, attributes, and behaviors, creating a clear understanding of the system's structure. Using an object model can create a consistent experience in the system. It will help in the development phase by using it as a reference and ensuring all project requirements for DriverPass.

To create an object model for DriverPass, we would need to identify the system's objects. For example, User, Appointment, and Course are all objects required within the system and have data needed for DriverPass to function correctly. Next, we would have to identify all relationships between objects. Administrator account creation would have a relationship with user registration since their accounts would need to be created in the same way as users. Object actions and attributes would also need to be defined to know how each object will function and what will be required to create the attributes for the object model accurately. This defining of attributes will clearly represent how each object works and the attributes that will be attached to them. Identifying objects, defining the relationship, and identifying objects' actions and attributes are how I would apply an object model to the design of the DriverPass system.

## Process and Object Model Comparison

Each model for the DriverPass system has its weaknesses and advantages when applying it to this scenario. A process model can be used to understand the system’s flow, functions, and tasks needed for the system to operate correctly. However, the disadvantages of this model can be shown when trying to understand the relationship of all components and how they are related. Not understanding the key relationships in the model can result in frustration in understanding how each one relates to another and defining relationships.

Object models’ advantages are that they are fantastic for understanding the relationship of the objects in the system and can be a great tool for seeing how objects relate to one another. A downside is this relationship can cause complexity when developing a large project. When creating a system for DriverPass, many objects will be related to one another, which could be complex when trying to figure out the relationships during the development of the system.

## References

Adkisson, H. (2019, March 30). Object modeling for designers: An introduction. Medium. <https://hpadkisson.medium.com/object-modeling-for-designers-an-introduction-7871bdcf8baf#:~:text=This%20is%20where%20an%20object,a%20system’s%20higher%2Dlevel%20constructs>.

Sulemani, M. (2021, January 4). *What is a Software Process Model? top 7 models explained*. Educative. https://www.educative.io/blog/software-process-model-types

*Valacich, J. S., George, J. F., & Hoffer, J. A. (2019).*Modern Systems Analysis and Design*(9th ed.). Pearson Education (US).*[*https://mbsdirect.vitalsource.com/books/9780135172827*](https://mbsdirect.vitalsource.com/books/9780135172827)