# CS 255 Model Application Short Paper

Dylan Dunagan

Dylan.Dunagan@snhu.edu

Southern New Hampshire University

## Process Model Application

When designing the system for DriverPass utilizing a process modeling approach, the goal is to map out the actual workflows the business follows. This approach focuses on how tasks are completed and how users interact with the system at every step. Since DriverPass is offering both online driving test preparation and in person lessons, a process model would clearly outline every step a student, administrator, or instructor would take from start to finish. Creating us case diagrams would show actions like “Schedule a Driving Lesson”, or “Reset Password” very easy to understand. It would define the interactions between the users and the system. Activity diagrams would then be used to break each of those use cases down even further to understand exactly how the system is supposed to act when performing the tasks. This is a major way to identify bottlenecks and inefficiencies. Overall, process modeling helps DriverPass ensure all team members understand how users move through the system, making it easier to plan, refine, and improve the design before development begins.

## Object Model Application

Applying an object modeling approach to the DriverPass system would focus more on the development of the system. It organizes each entity involved in the system into objects that represent different users and functions. Each object would have its own data and have behaviors related to its role in the system. For example, a “Student” object would require their personal information, such as name, email, phone number, and password, but would also be linked to various functions, such as scheduling, viewing progress, or taking practice tests. Object modeling is ideal for DriverPass because it aligns well with object-oriented programming languages which is why it would go hand in hand with the development process. It allows the development team to create modular code that is very adaptable, reusable, and easy to update. If DriverPass wants to add new packages in the future, it can be done without changing the entire program.

## Process and Object Model Comparison

Both process and object modeling approaches bring important benefits to Dreiverpass, but each serves very different purposes. Process modeling is a great starting point because it visually represents how users interact with the system. For DriverPass, this helps ensure that important steps like scheduling lessons, managing instructors, and processing payments are organized logically. It also provides a clear way to communicate with stakeholders who may not have a technical background but need to understand how the system supports their requests. Object modeling takes over, when the focus shifts from the workflow to the data and software framework. Object modeling is better suited for defining the system’s structure and how different parts of it interact with itself. It provides a clear and understandable framework for the coding involved, supporting scalability and reusability. The downside of object modeling is that it can be harder for someone who is not technically savvy to follow along, while a software developer would probably understand it better than a process model. For a project like DriverPass, the best solution is to use both approaches for different audiences. The process model is for everyone to understand what the system needs to do and ensures that all the stakeholders’ requests are accounted for. The object model for helping developers know exactly what needs to be in the system on a much deeper and specific level. Together they provide a complete picture that supports both communication, development, and functionality.