The purpose of this essay is to summarize and evaluate the different system design approaches we could use to build functionality for our client. Our client, Driver Pass, would like a system that would allow them to easily help students study for their driver’s test. The system we have outlined to make this happen is a Linux server hosted web application with multiple pages and a backend database to handle the desired functionality outlined in the design document. This document will serve as an exploration and comparison between the process and object modeling approach to building the system. By the end of this discussion, we will find the design approach best suited to the task.  
 Project Modeling and Object modeling are both ways of designing systems with a focus on how to provide system functionality. In an object modeling approach, properties and attributes that share a common functionality are encapsulated into objects that organize code. Through object modeling, engineers can get an overview of object relationships, properties, and functionality. In an object model, code is encapsulated in classes and functionality is created via the interaction between objects instances. The benefits of using an object model are that it is easy to understand entities within a system and data is secured and accessed only when needed.

In a process model behavior and data is largely handled sequentially and functionality is created and executed when required. Process models are largely reliant on good control flow design and are easier to understand for stakeholders. A high-level process diagram for a system implemented using a process model can often give a pretty good description of how that system behaves to a non-technical entity. Another major benefit of a process modeling approach is that often engineers can highly optimize functionality and speed of execution. Because of this the process modeling approach should be considered when dealing with a time sensitive systems.

Implementing a process modeling approach to our system design for our client would largely be done server side. For example if we implemented our functionality via some sort of service oriented architecture we would be able to easily modify and handle request in Realtime as services are needed. For example if a user of our web application were to request some sort of functionality for or from the database then a real-time request would be sent from the clients homepage to the database. The homage would need to be static and replicated based on different data that may be present from different user. The main reason I want to stay away from a process model for our client is because the runtime for this system will not be a constraint and with a web application that has so many attack vectors the security of the system may be compromised if a vulnerability is found and exploited.

The Object modeling approach on the other hand is an intuitive solution to implementing the system for our client driver pass. To start we would use a façade design pattern to provide the interface object for users of different roles to use. This interface pattern would serve as the core functionality and information will be interchangeable and modular via the user object instances. All User entities be they customer, teacher, or admin will inherit from this user class which will require a login and multi factor authentication. Essentially if we implement this system using a object model we would be able to abstract away functionality such as the server, database and authentication when needed keeping the integrity of our application under control. We will have encapsulated our functionality and made our application easier to debug and patch across the whole system which will be necessary to building a resilient scalable web application.

for the system outlined in the system design document I recommend we move forward with an object focused approach to providing functionality. The modularity of class objects and the use of the façade design pattern allow us to keep the code simple and sweet while also being as secure as possible. The Process focus should be reserved for a more embedded system with a singular unifying purpose and low security attack possibilities such as an internet facing web application. With an application that will have many entities A process model would better suit our implementation of this system.

In conclusion, while for the purposes of this system I recommend using an object focused approach to creating system functionality it is important to remember our chosen approach is largely dependent on the underlying requirements. If we needed a fast embedded system to handle a small IOT device or a system with less variables such as an ATM machine a process model would serve much better. I think overall it comes down to the static nature of our application in general. Teachers, students, and our administrators will need persistence throughout the system in order for us to provide that functionality. The overall need for persistent states throughout our system is why I feel and object oriented approach is necessary.