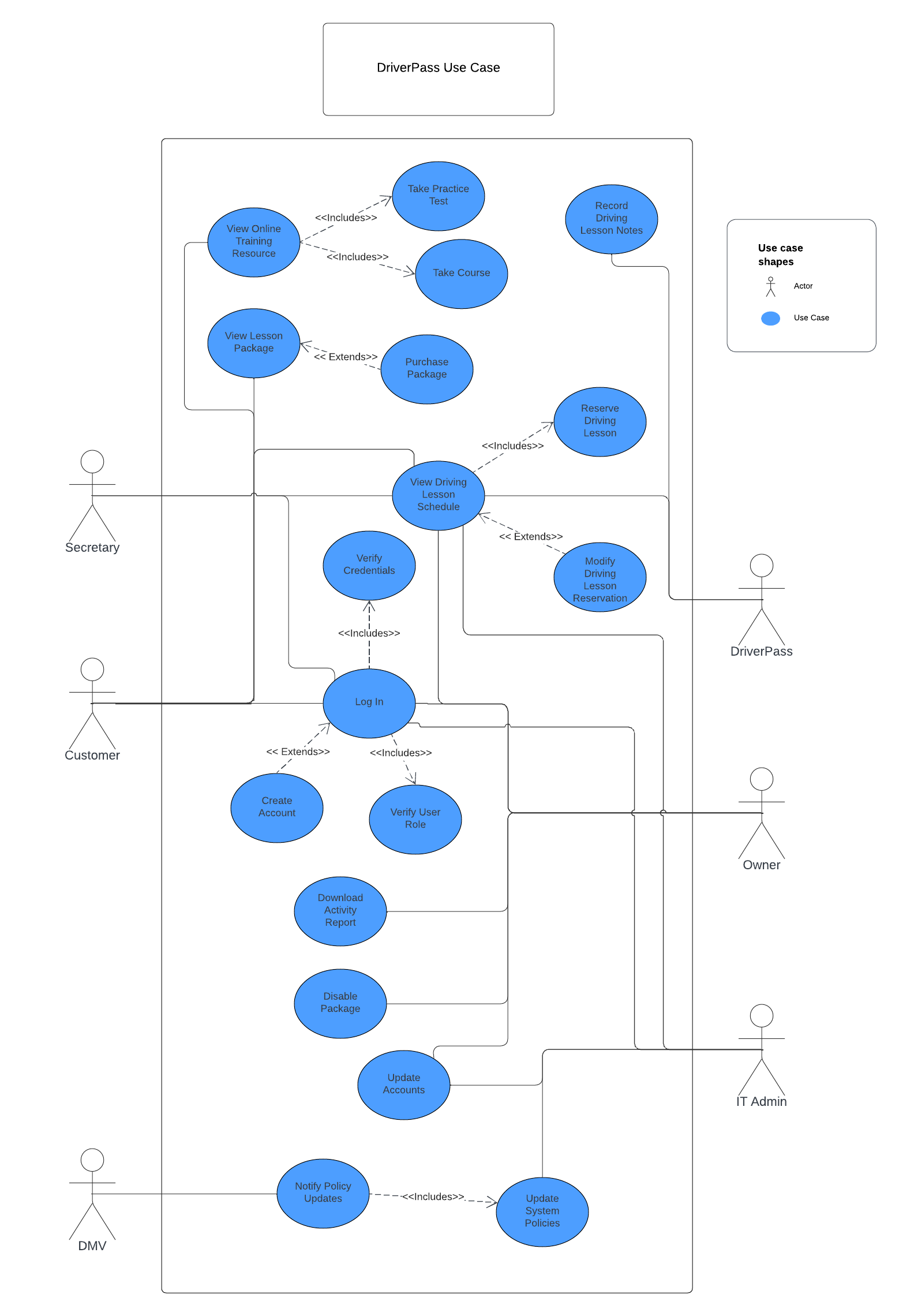
# CS 255 System Design Document

.

## UML Diagrams

### UML Use Case Diagram

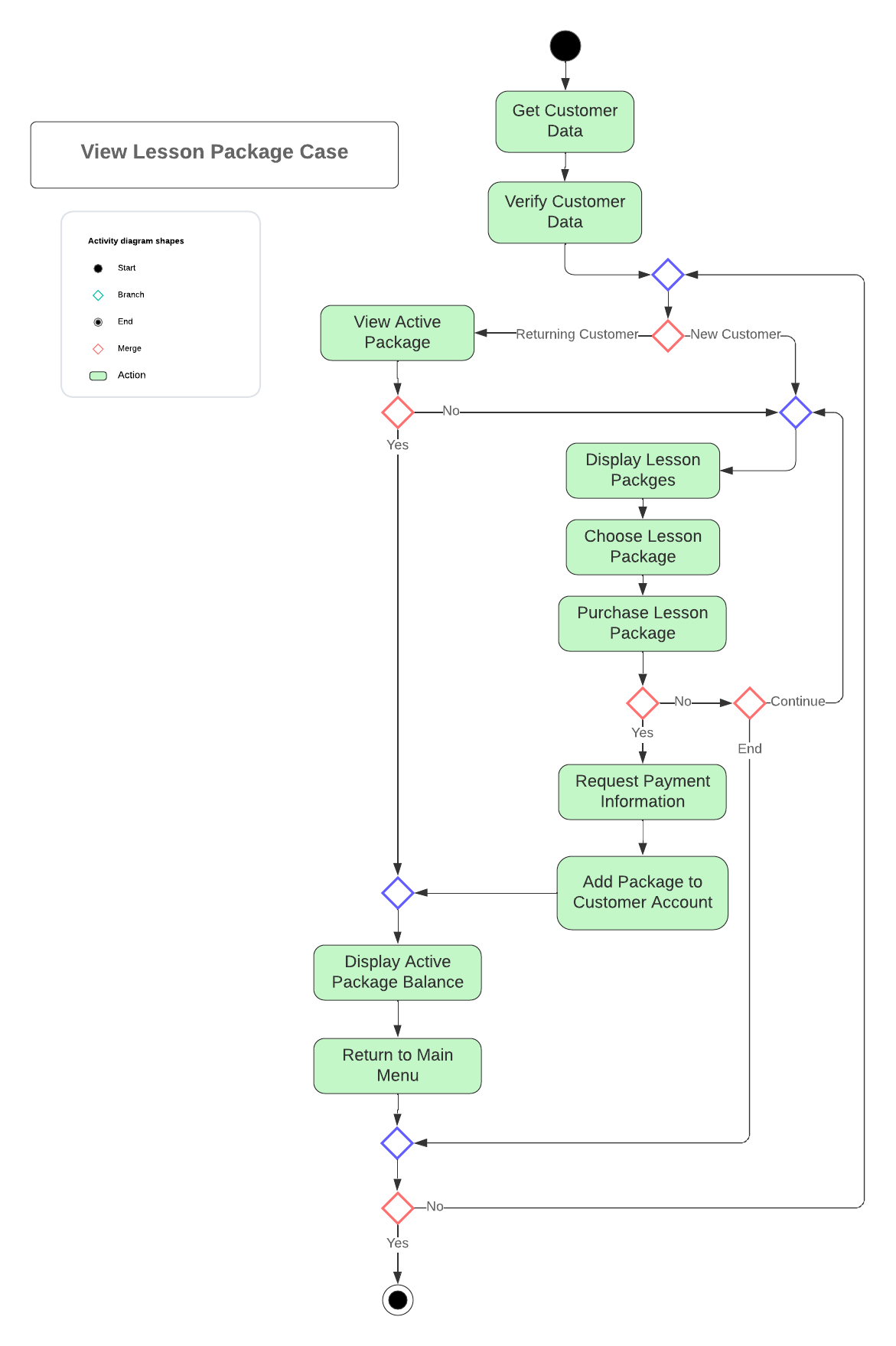
A top-view interpretation of behavior is essential for the development and design of a system. This can be achieved with Use Case diagrams that show how outside and internal processes of a system behave. Specifically, “Use Cases are the primary drivers for all the UML diagramming techniques”(Dennis et al., 2012) and “communicates at a high level what the system needs to do”(Dennis et al., 2012). Both points sum up the importance of a Use Case and how they are used as the central perspective in the UML diagramming method. They list all the functional requirements of a system as individual Use Cases and show how actors can interact with them. The creation of Use Case Diagram is the first step in the UML design process and is used as the focal point for the construction of the following models.

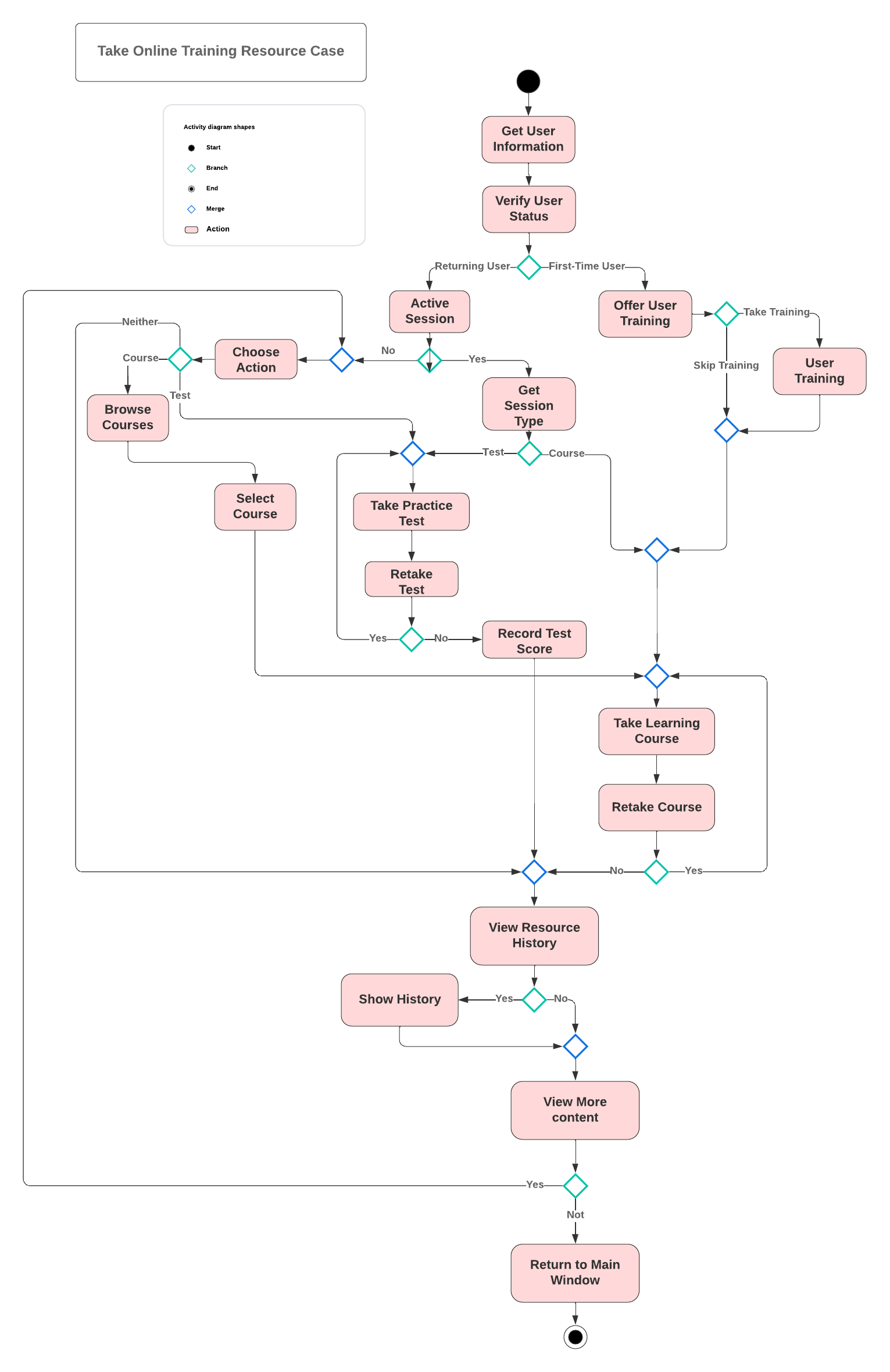


### UML Activity Diagrams

An individual Use Case represents a specific function or process of the system. The individual case can be represented with activity diagrams that demonstrate the flow between the different subprocesses. A good way of describing activity diagrams is as “sophisticated data flow diagrams” ”(Dennis et al., 2012). This statement is demonstrated using a logical decision operator that allows for the flow to be handled on a case-by-case basis. These diagrams are essential for understanding the

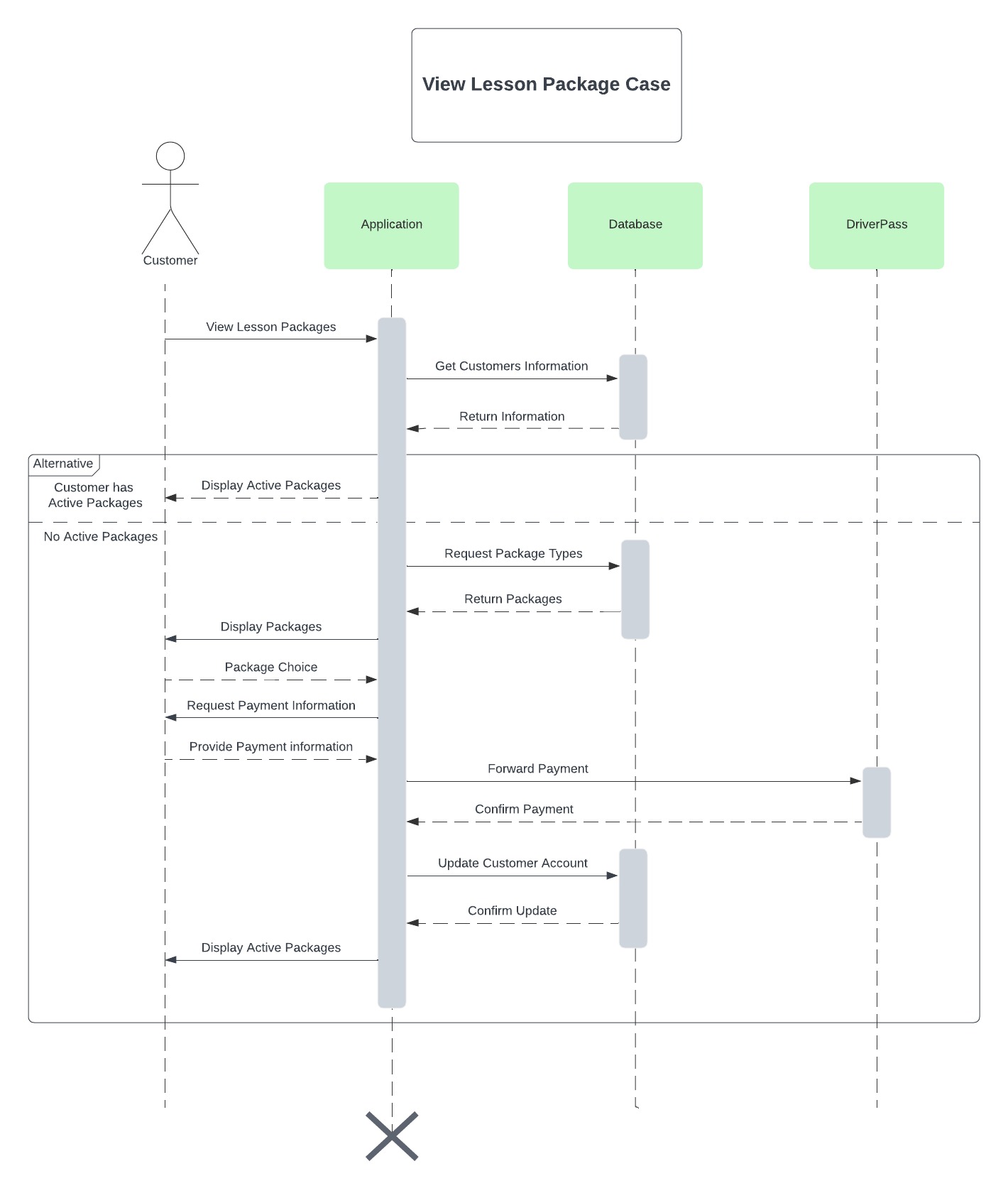
complex details of a larger process in a controlled manner.





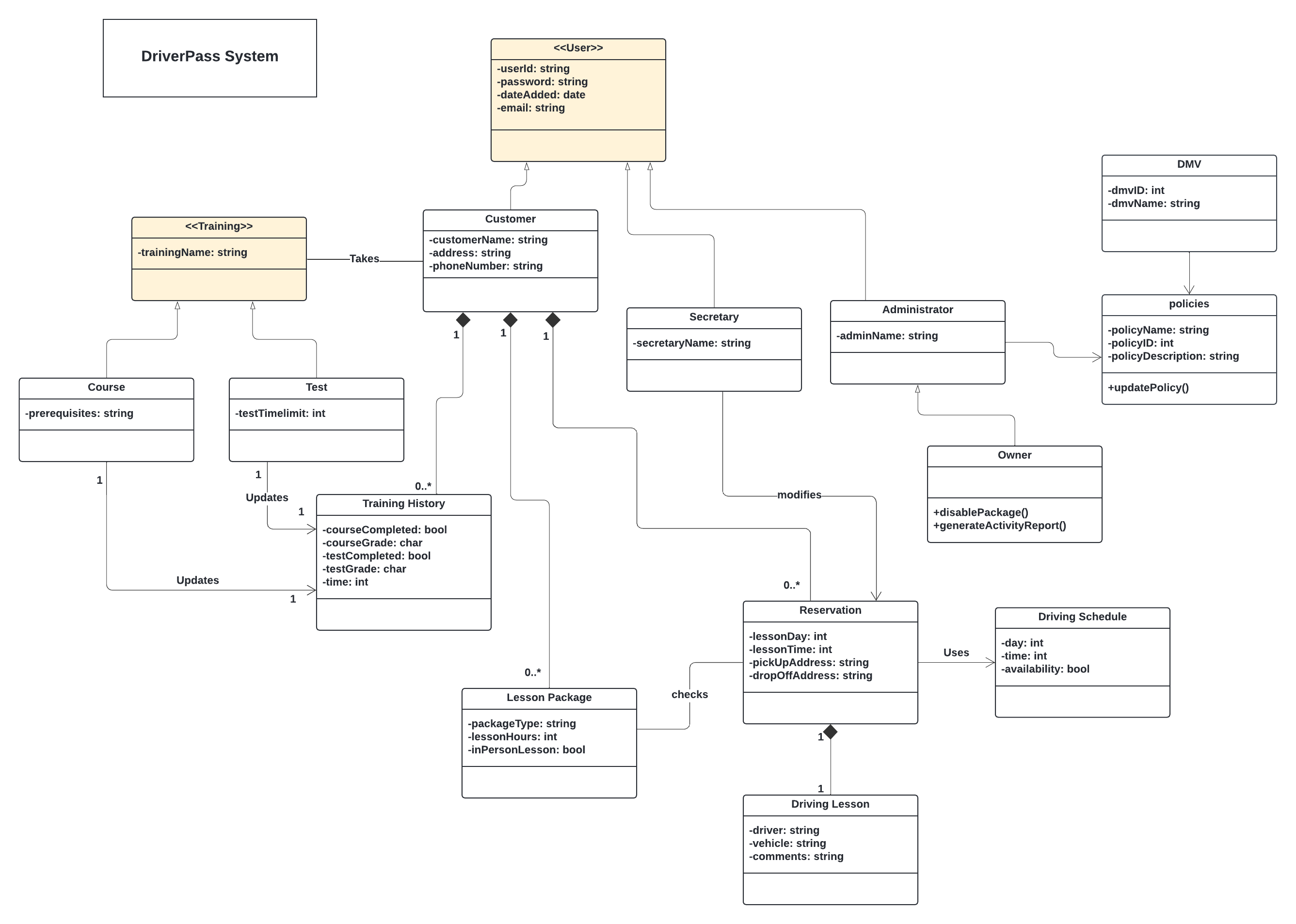
### UML Sequence Diagram

An important aspect of explaining a single Use Case is to visualize the behavior from a dynamic perspective, showing how the system and the user handle communication. We can do this with a behavioral model called a sequence diagram. These diagrams show the communication paths between each object and actor in the order in which they occur. One way of describing them is that they “emphasize the time-based ordering of the activity”(Dennis et al., 2012). Overall, these models offer a different perspective that highlights the time and objects that are not seen in your typical activity diagram.



### UML Class Diagram

During the development and design of a system, UML Class Diagrams are used as object models to explain a system on a structural level. These models are used to plan how the various classes and objects will interact with each other. This is achieved by showing different types of relationships, multiplicity of instances, class attributes, and operations. The classes themselves represent the objects that will be used by the system. A good explanation of an object is that an “object is a single occurrence of a class”( Valacich & George, 2019, p. 276) and that an “object’s state is determined by its attribute values and links to other objects”( Valacich & George, 2019, p. 276). This directly highlights the importance of the diagram and how it describes the attributes, and operations within each class as well as shows how the objects are linked to others through various types of relationships and associations. The UML diagram is an essential structural model that can be used for the entire development of the system.



## Technical Requirements

Technical requirements are limitations of a system that are related to hardware, software, and the physical environment. They limit design choices and alter the functionality of the system, forcing developers to build around them. Understanding these requirements is essential when discussing limitations and non-functional requirements.

**Hardware**

* The system will need to run on a cloud-hosted server.
* The system will need to allow for connection between desktop and mobile devices.
* Employees will need computers to access the application as well as internet access.

**Software**

* The system will require the webserver to make real-time updates for accurate driver lesson scheduling.
* The system will need to host a website.
* The system will need to receive information from the DMV through an API.
* The system will need to protect sensitive custom information including payment data.

***References***

*Valacich, J. S., & George, J. F. (2019). Modern systems analysis and design (9th ed.). Pearson UK.*

*Dennis, A., Wixom, B. H., & Tegarden, D. (2012). Systems analysis and design with UML (4th ed.). John Wiley & Sons.*