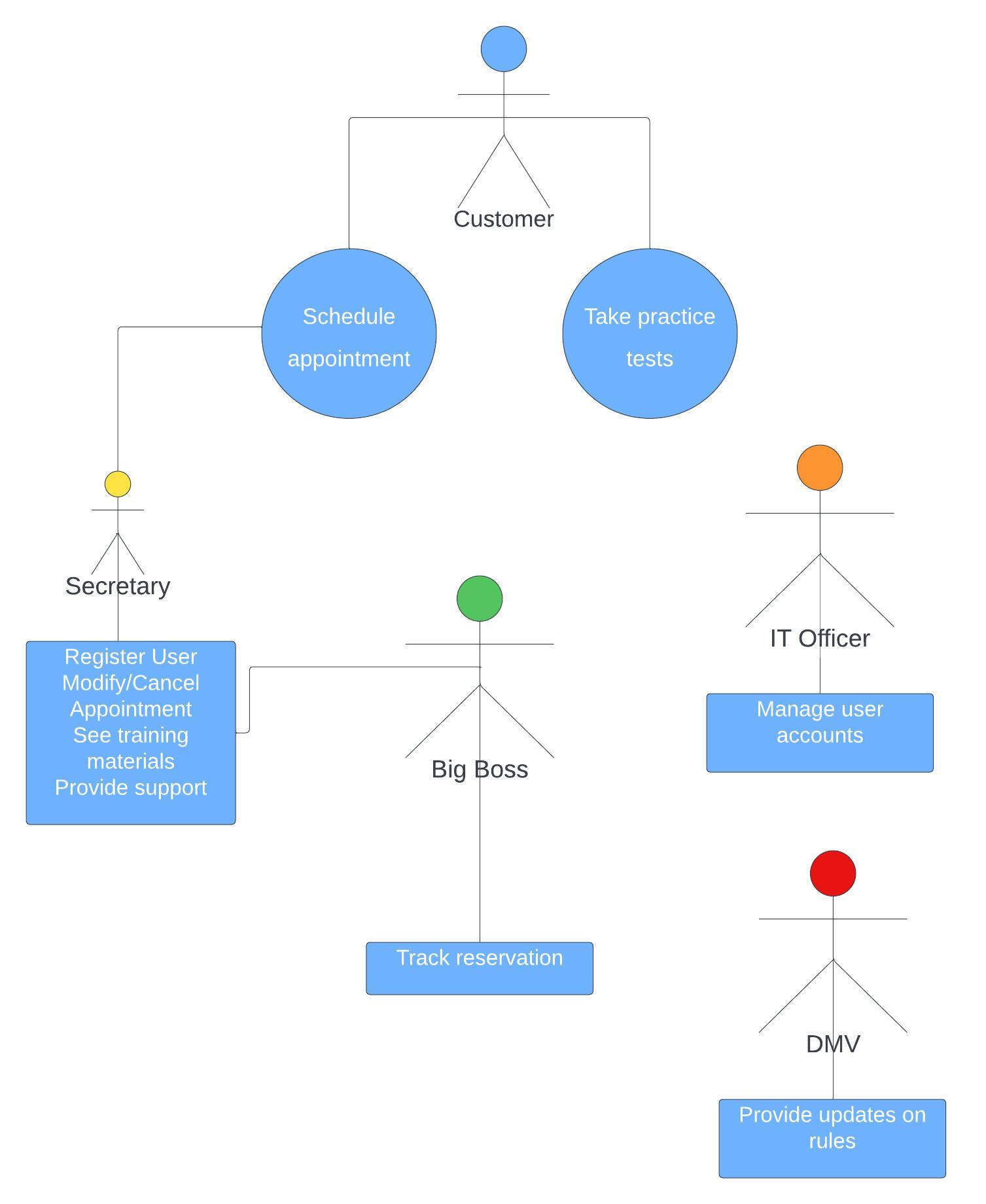
# CS 255 System Design Document Template – Kemal Cankurt

## UML Diagrams

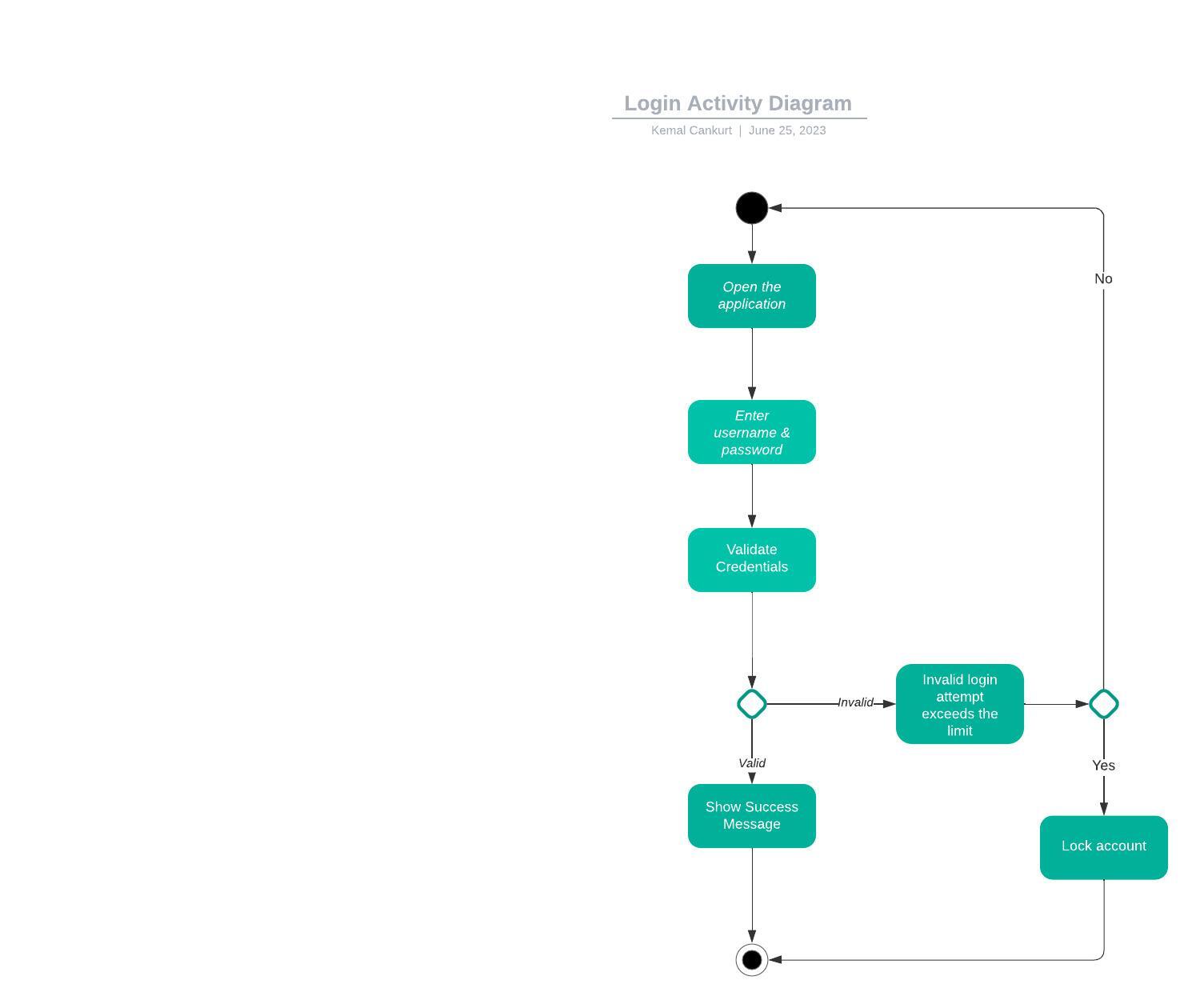
### UML Use Case Diagram

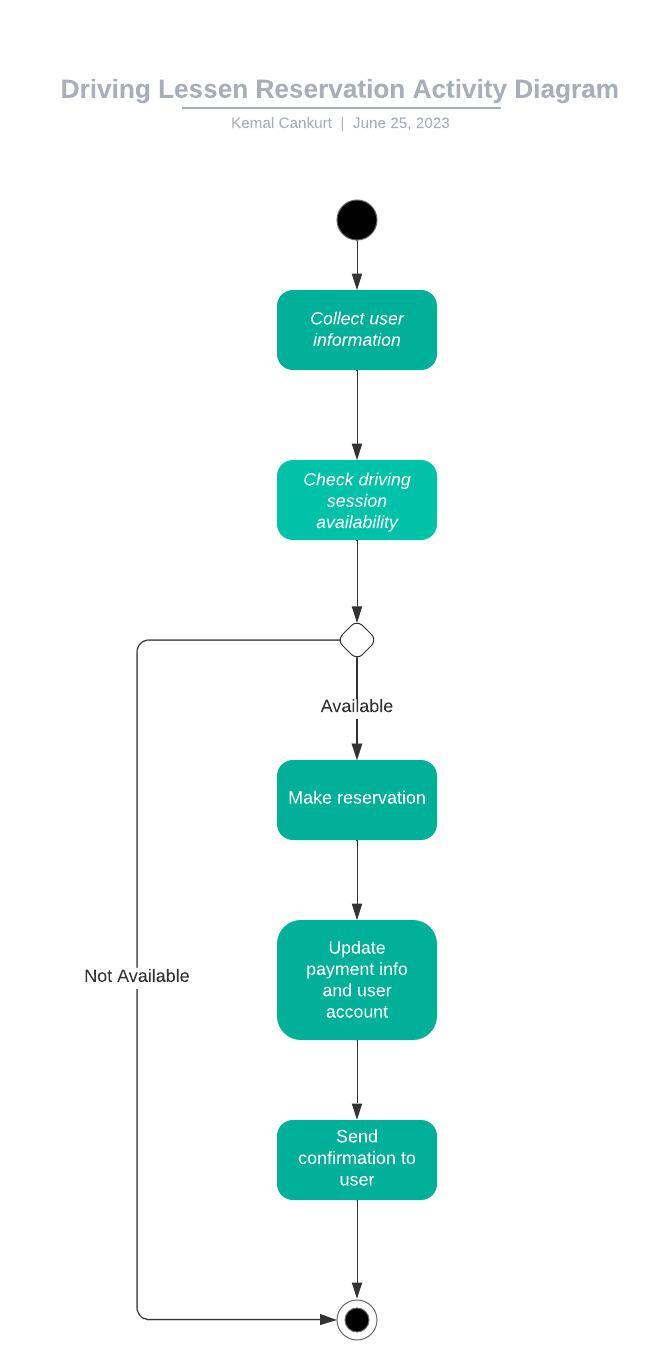
*[In Module Six, you were asked to complete a use case diagram based on your system design. If you would like to make any adjustments to your diagram, please do so. Please insert your use case diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]*



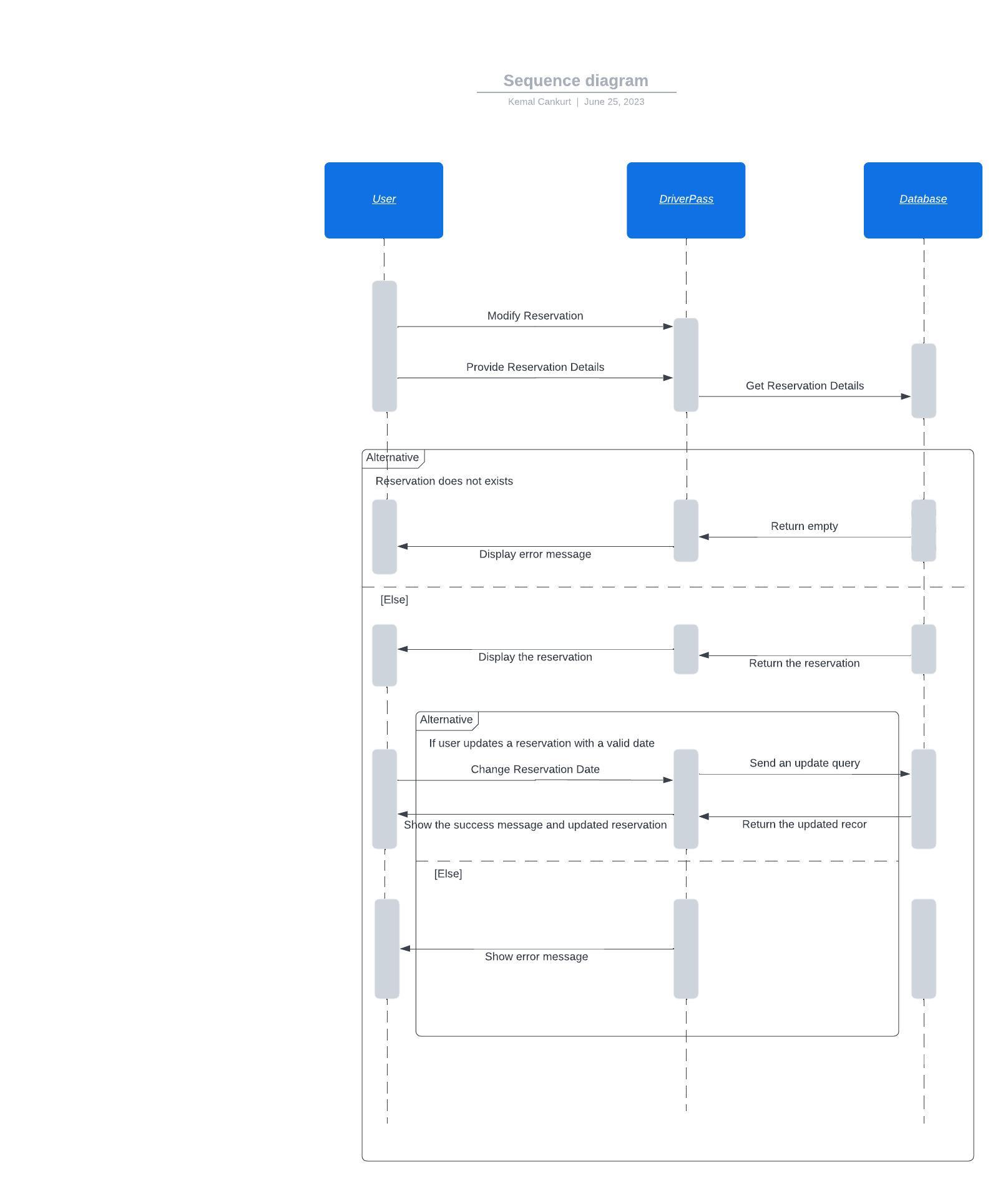
### UML Activity Diagrams

*[You were asked to choose* ***two*** *use cases and create* ***two*** *activity diagrams, one for each use case. Please insert* ***both*** *of your activity diagrams here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]*



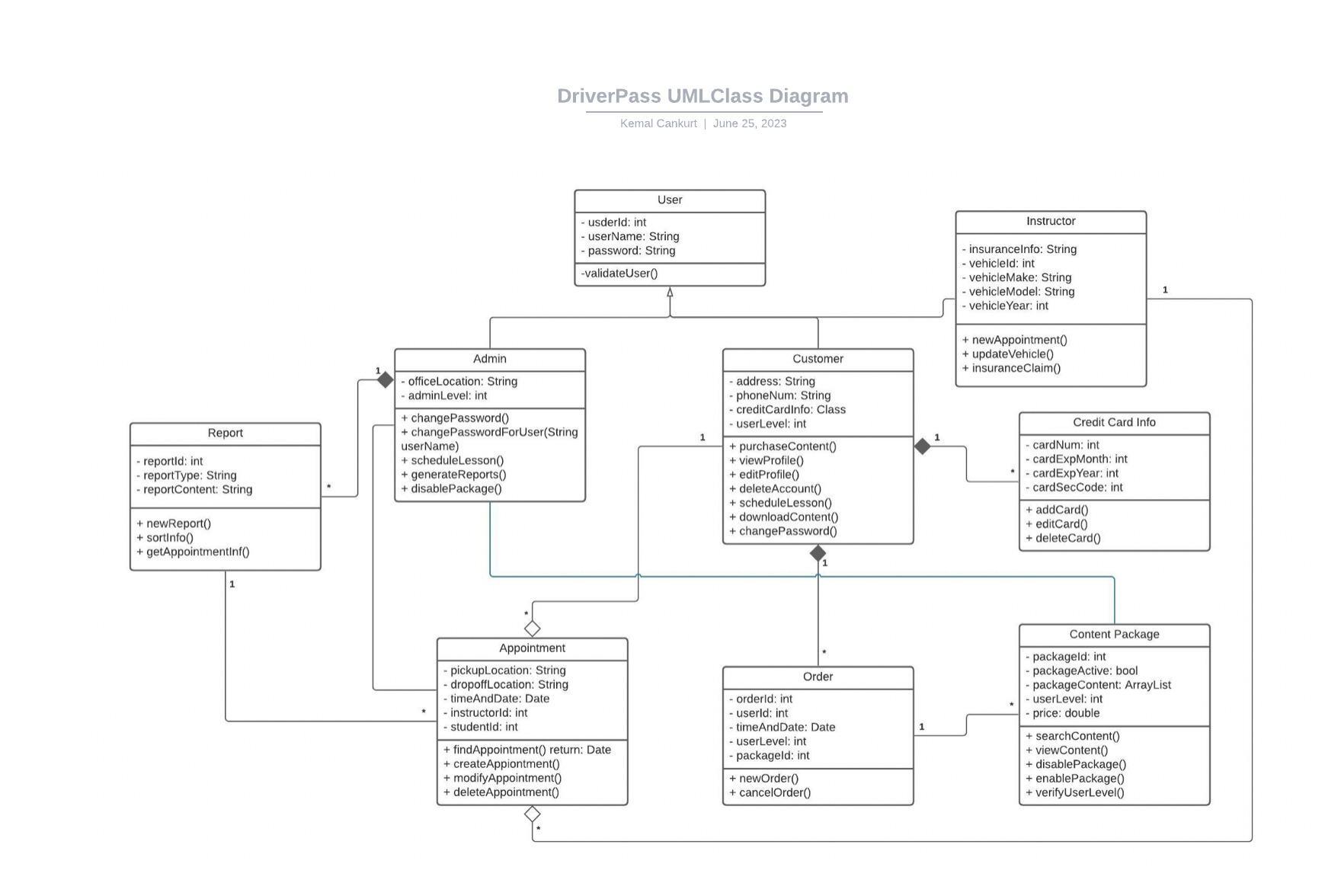


### UML Sequence Diagram

*[You were asked to create a sequence diagram based on* ***one*** *of the use cases you chose. Please insert your sequence diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]*

### UML Class Diagram

*[You were asked to create a class diagram based on the different classes and attributes needed for your system design. You are* ***not*** *required to include methods, but you may if you wish. Please insert your class diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s requirements.]*



## Technical Requirements

Hardware Requirements:

* The system needs a reliable server infrastructure to host the web application and effectively handle user queries. This includes having servers with enough processing speed, memory, and storage.
* The system should be accessible from various devices, such as desktops, laptops, and smartphones.

Software Requirements:

* The system needs a web application framework to manage the user interface, business logic, and database integration. A good example might be a framework like Django, Spring Boot, or ASP.NET, depending on the technology stack picked.
* The system's data must be stored and managed using a dependable and scalable database management system, such as MySQL, PostgreSQL, or MSSQL.
* System development will need programming to build the application logic and integrate multiple system components, languages like Python, Java, or C#.

Tool Requirements:

* IDEs will be needed during the development and testing stage, such as Visual Studio, Eclipse, or PyCharm.
* The development team's source code modifications must be tracked and managed using a version control system. Hence, Git could be used.

Infrastructure Requirements:

* The system may benefit from the scalability, flexibility, and simple deployment choices cloud services like AWS or Azure provide.
* A trustworthy backup and recovery strategy should be in place to guarantee data availability and integrity in the event of system failures or disasters.