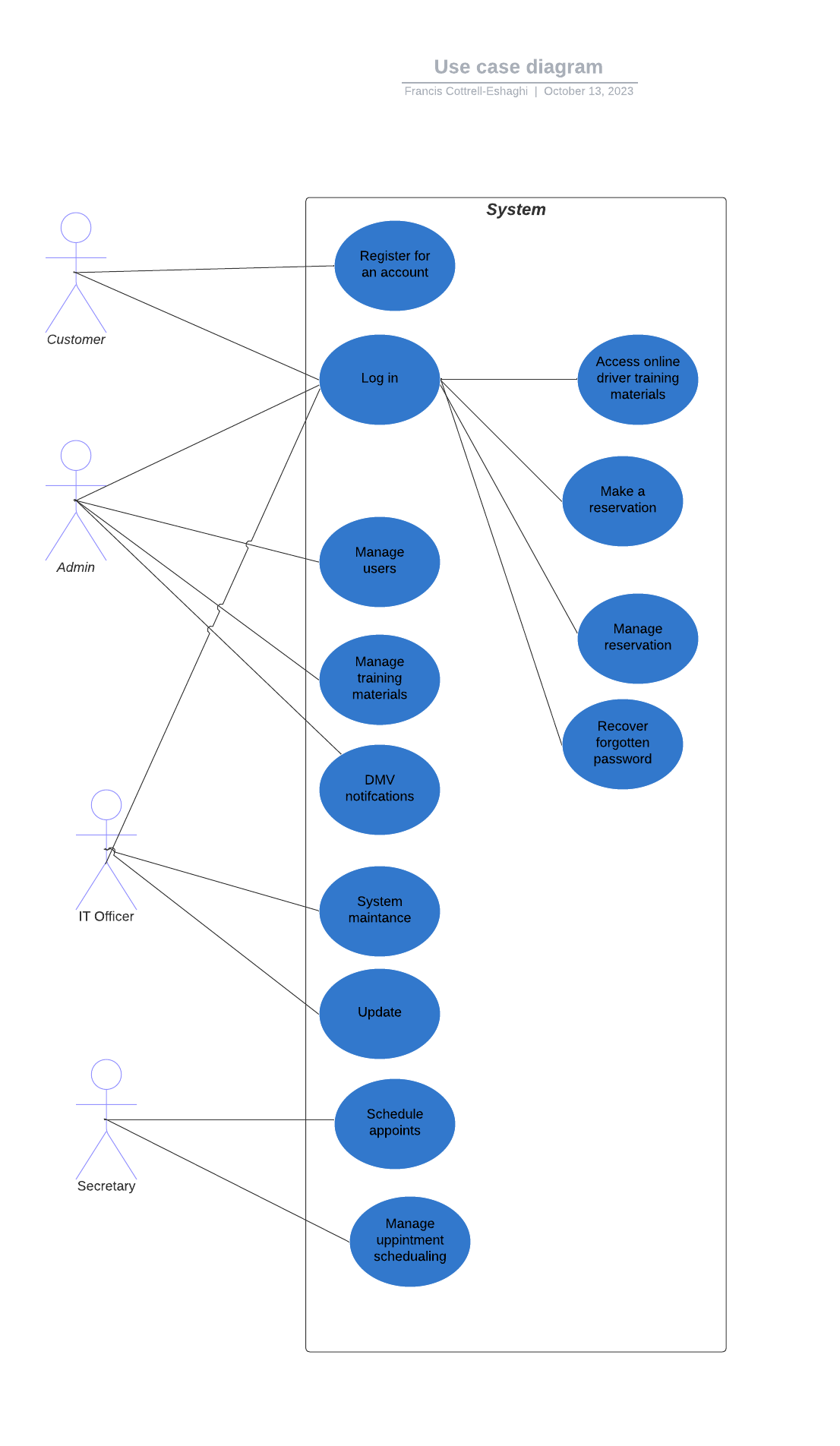
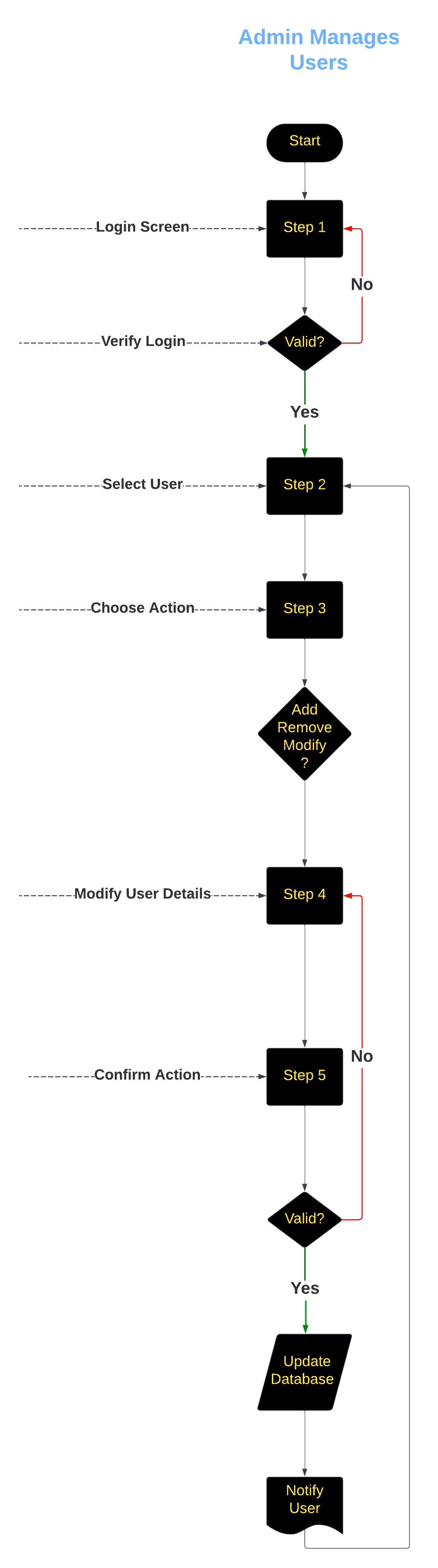
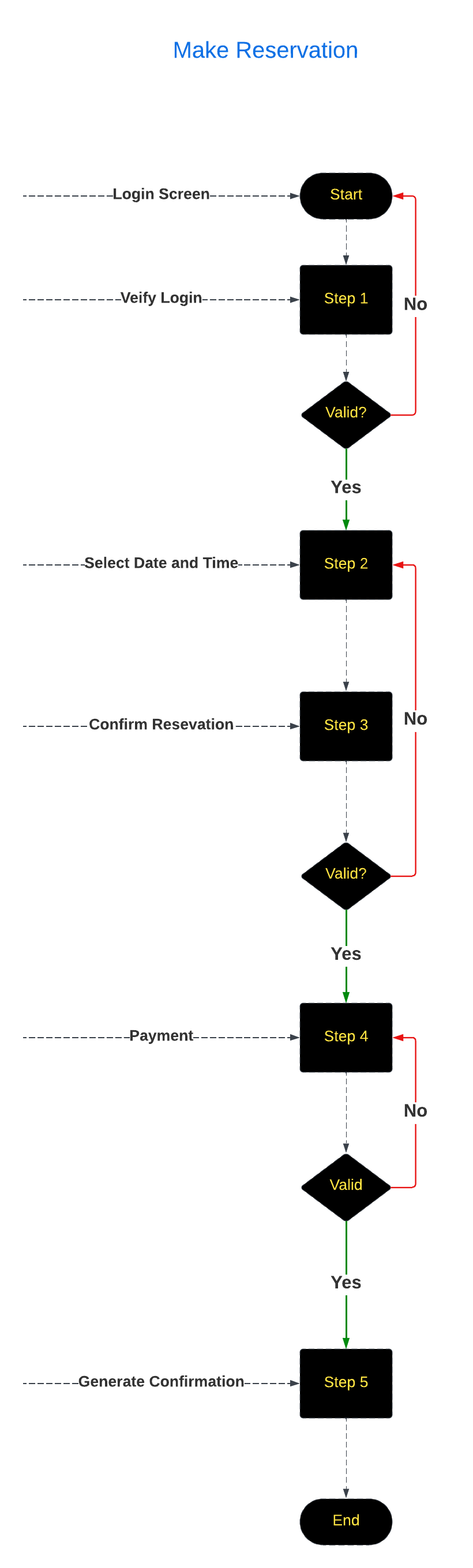
# CS 255 System Design Document Template

## UML Diagrams

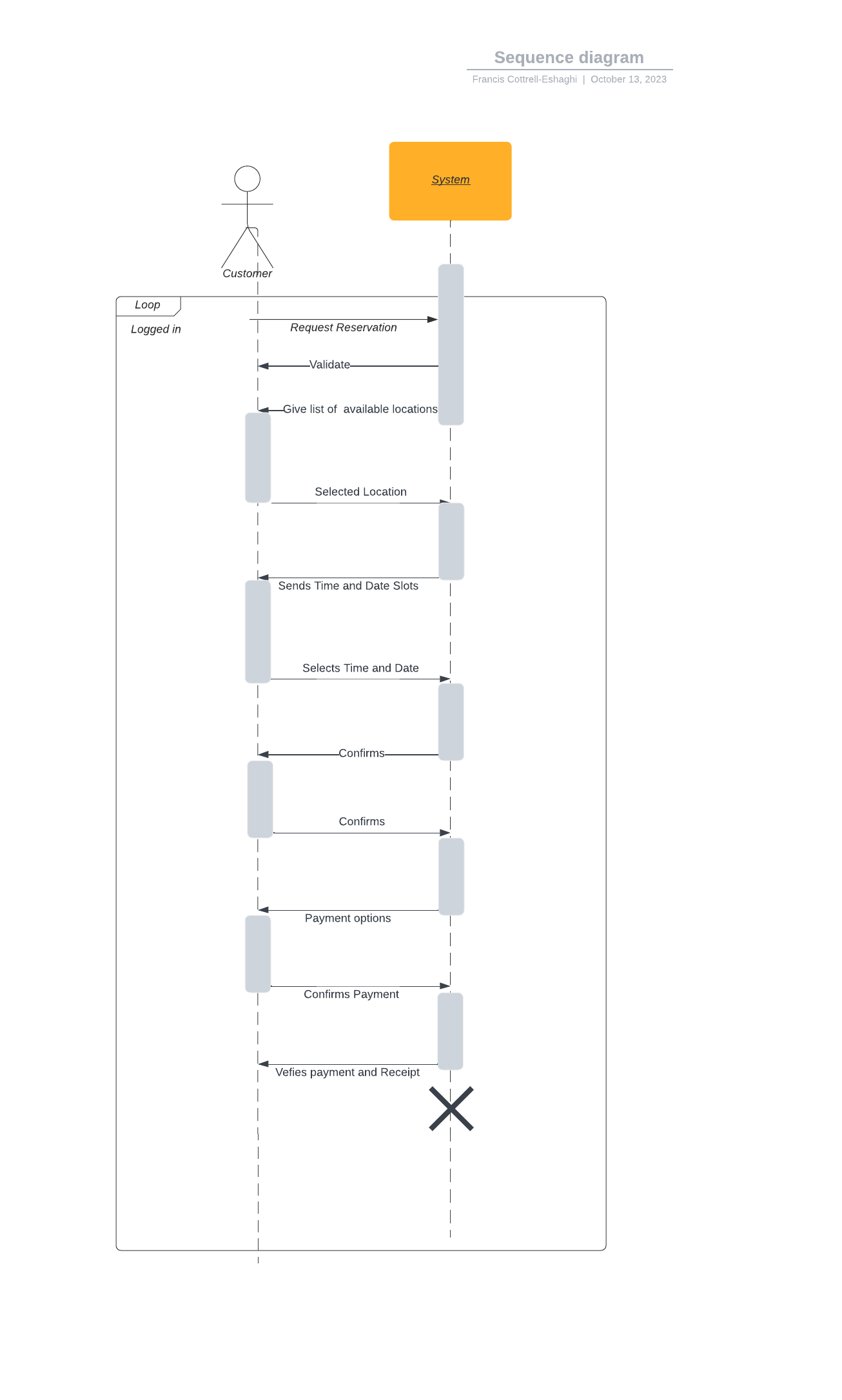
### UML Use Case Diagram



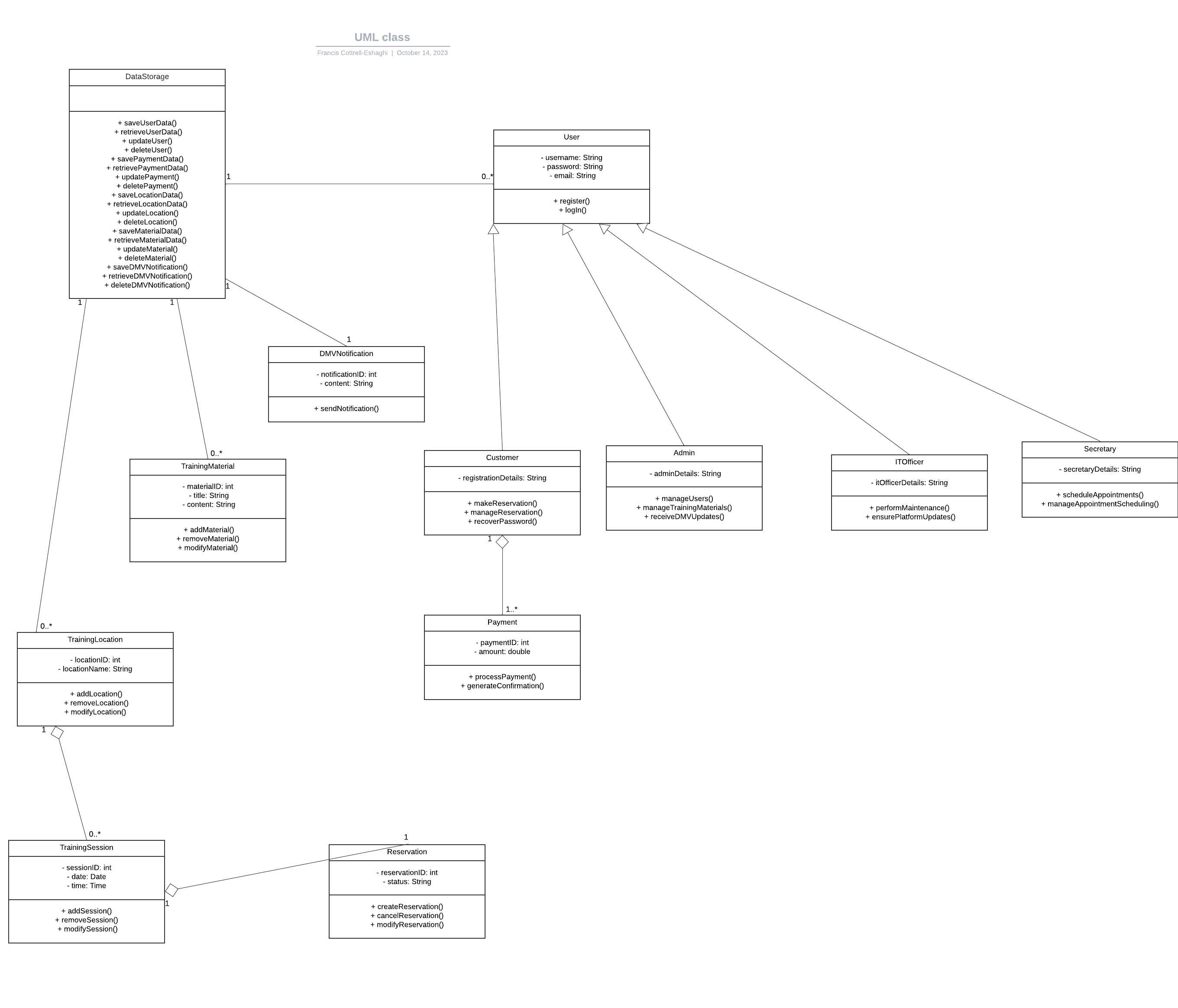
### UML Activity Diagrams



### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

*Hardware Requirements: The system requires server infrastructure capable of hosting the application and the associated database, with an emphasis on efficient handling of user loads and database operations. Additionally, a reliable network infrastructure is crucial for seamless data communication between components. End-users should be able to access the system from a variety of devices, including desktop computers, laptops, tablets, and smartphones.*

Software Requirements: The server infrastructure must run on a suitable operating system, Linux, or another compatible system. A web server, such as Apache, is essential for serving the web application. A relational database management system (RDBMS), such as MySQL, is required to efficiently store and manage data. The choice of a development framework and programming language, such as Python, Java, Ruby on Rails, or PHP, will depend on the development team's expertise. Web development tools, integrated development environments (IDEs), code editors, and version control tools are essential for application development. Security software and protocols must be in place to ensure data security, user authentication, and encryption of sensitive information.

Tools and Libraries: Front-end frameworks, including HTML, CSS, and JavaScript frameworks (e.g., React, Angular, or Vue.js), are used for building the user interface. Back-end frameworks or libraries are employed for server-side operations. Libraries and drivers for database connectivity are required, as well as user authentication libraries to ensure secure user access. If payment processing is involved, integration with payment gateways or services may also be necessary.

Infrastructure Requirements: Cloud services like AWS, Azure, or Google Cloud can be considered for scalability, reliability, and data redundancy, although this is optional. Load balancing can be implemented to distribute incoming web traffic across multiple servers, ensuring better performance and redundancy. Data backup and recovery procedures are essential to safeguard against data loss or system failures. The system should be designed with scalability and redundancy in mind to handle increased user loads and provide high availability. Compliance with DMV data sources is crucial, ensuring that the system can access and process DMV updates and notifications in accordance with regulations. Finally, monitoring and logging tools should be implemented to track system performance, identify issues, and facilitate troubleshooting.