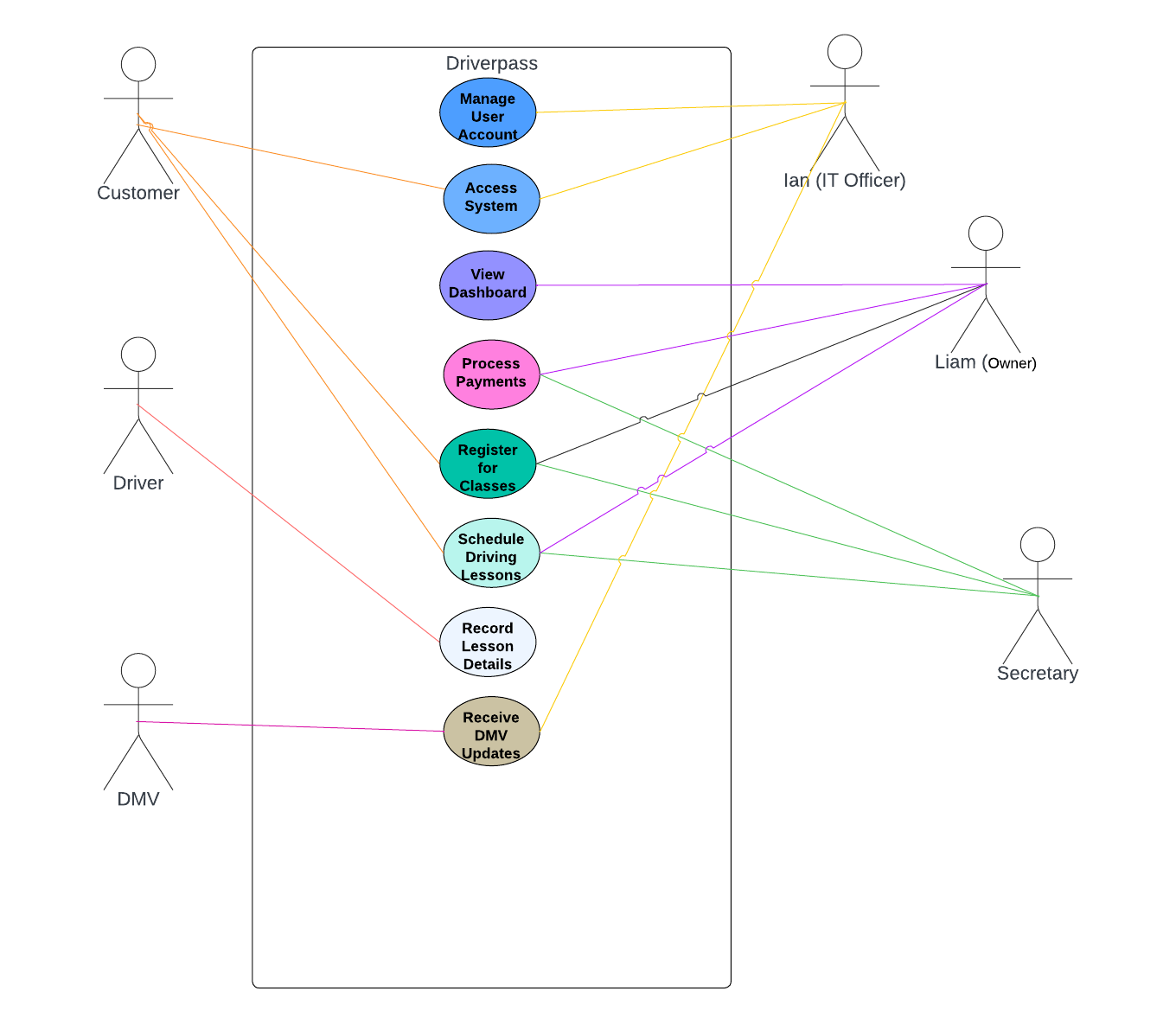
# CS 255 System Design Document Template

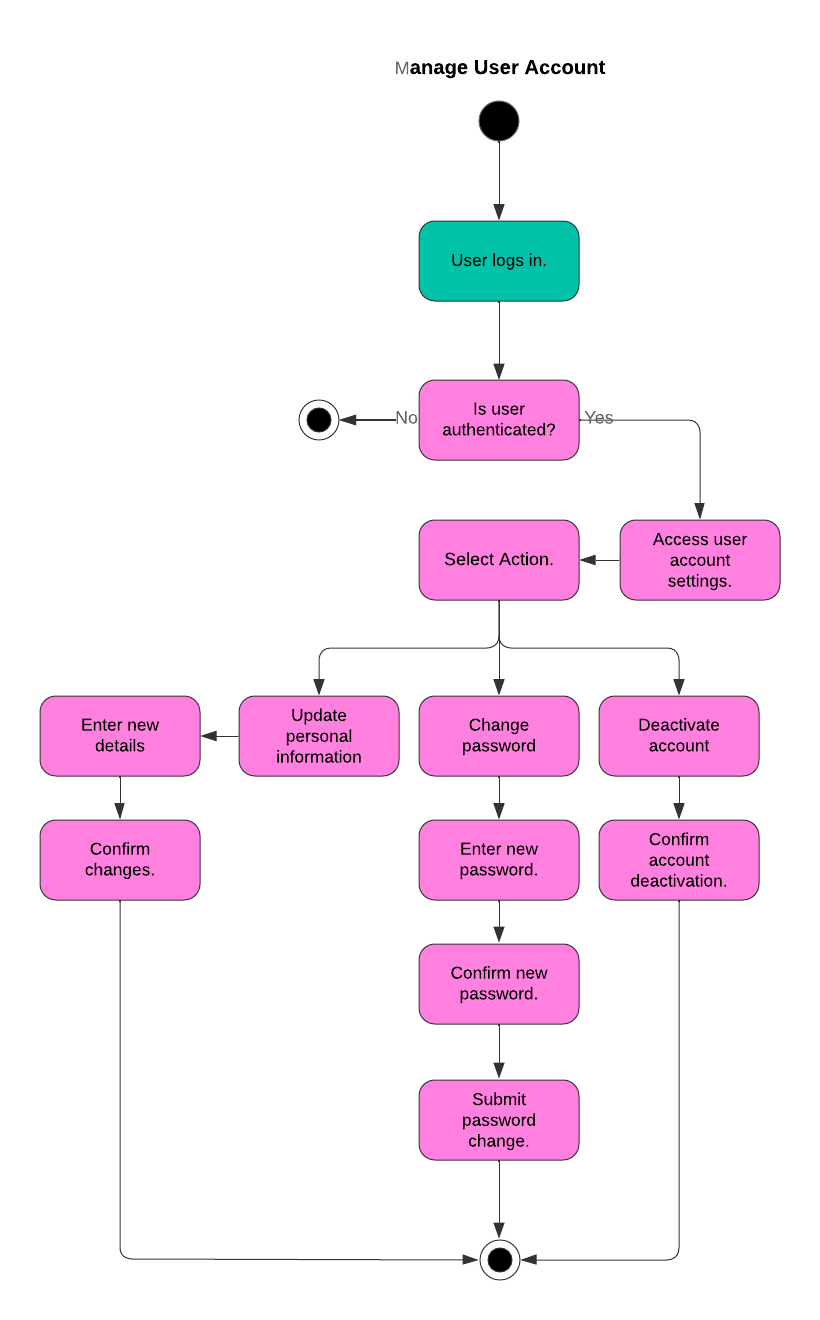
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead the goal is to complete each section based on what your client’s needs are. Remove this note when you are finished, and replace all bracketed text with the relevant information.

## UML Diagrams

### UML Use Case Diagram

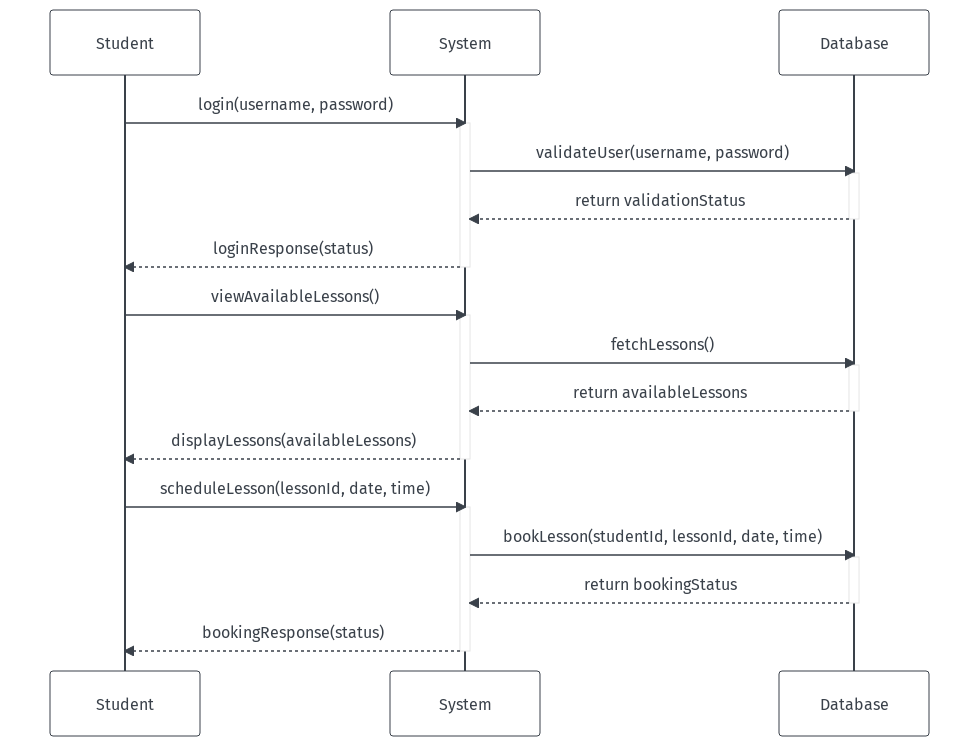


### UML Activity Diagrams

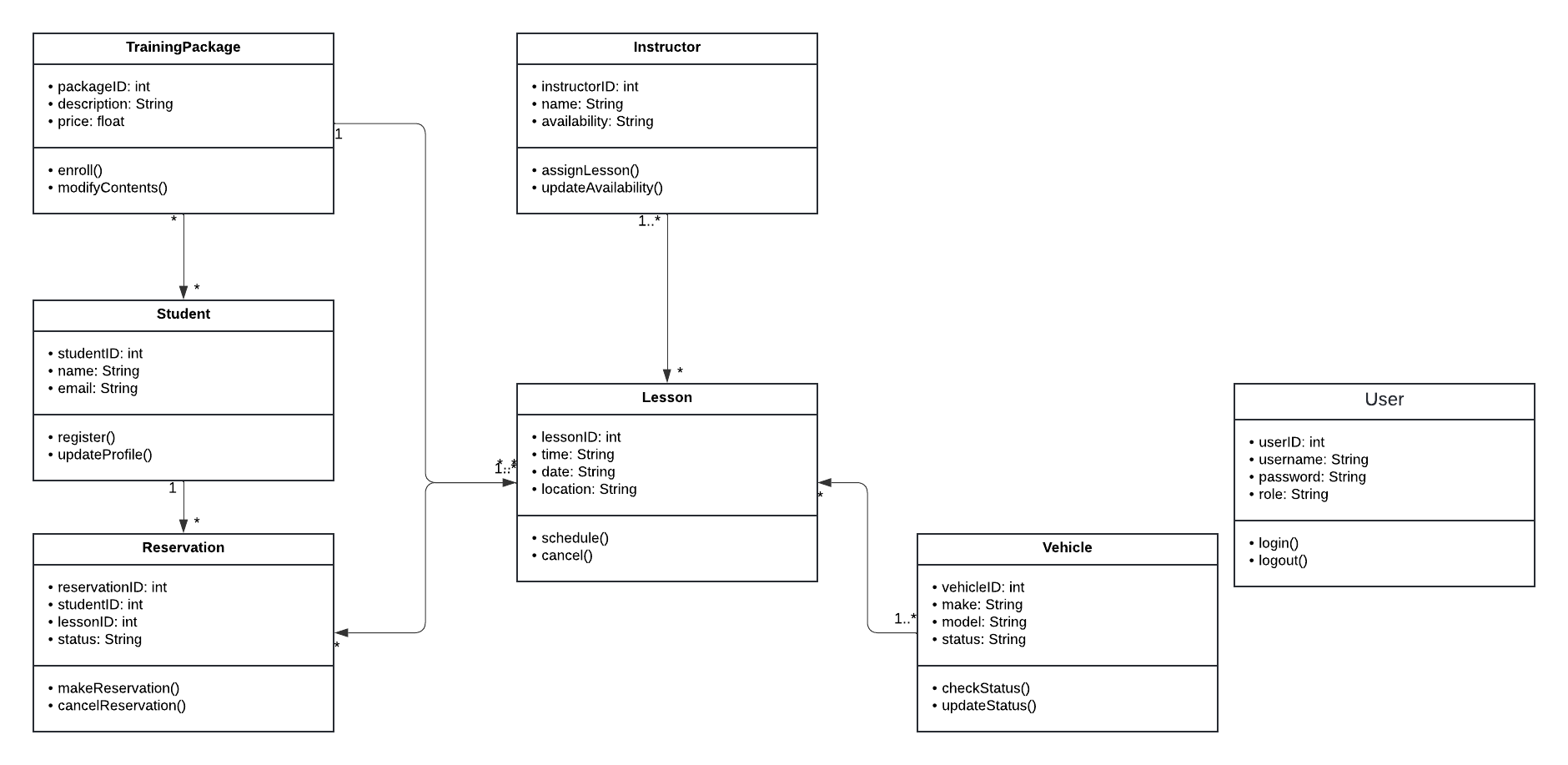


### 

### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

#### *Hardware Requirements*

* **Servers**: Use cloud-based services to simulate server environments (e.g., AWS Free Tier for educational purposes).
* **Client Devices**: Any device capable of running a modern web browser will be suitable for accessing the system.

#### Software Requirements

* **Operating System**: Servers can run on any standard OS, such as Linux or Windows, available on the cloud service.
* **Database**: Use a simple relational database like SQLite for ease of setup and use, suitable for educational projects.
* **Back-End Technology**: Node.js with Express framework, which is lightweight and beginner-friendly.
* **Front-End Technology**: Use React.js for the user interface to practice building dynamic web applications.

#### Tools and Infrastructure

* **Development Tools**: Visual Studio Code is recommended for coding due to its extensive support for web development.
* **Version Control**: Git for source control, hosted on platforms like GitHub, which also facilitates project tracking and collaboration.

#### Security Measures

* **Basic Authentication**: Implement basic username and password authentication.
* **Data Security**: Introduction to using HTTPS for securing data in transit.

#### Monitoring and Maintenance

* **Basic Monitoring**: Use simple tools like console logs in development and basic cloud monitoring tools provided by the cloud service for keeping track of the system's status.

#### User Interface Requirements

* **Design**: Focus on creating a clean and simple interface using HTML, CSS, and Bootstrap for responsiveness.