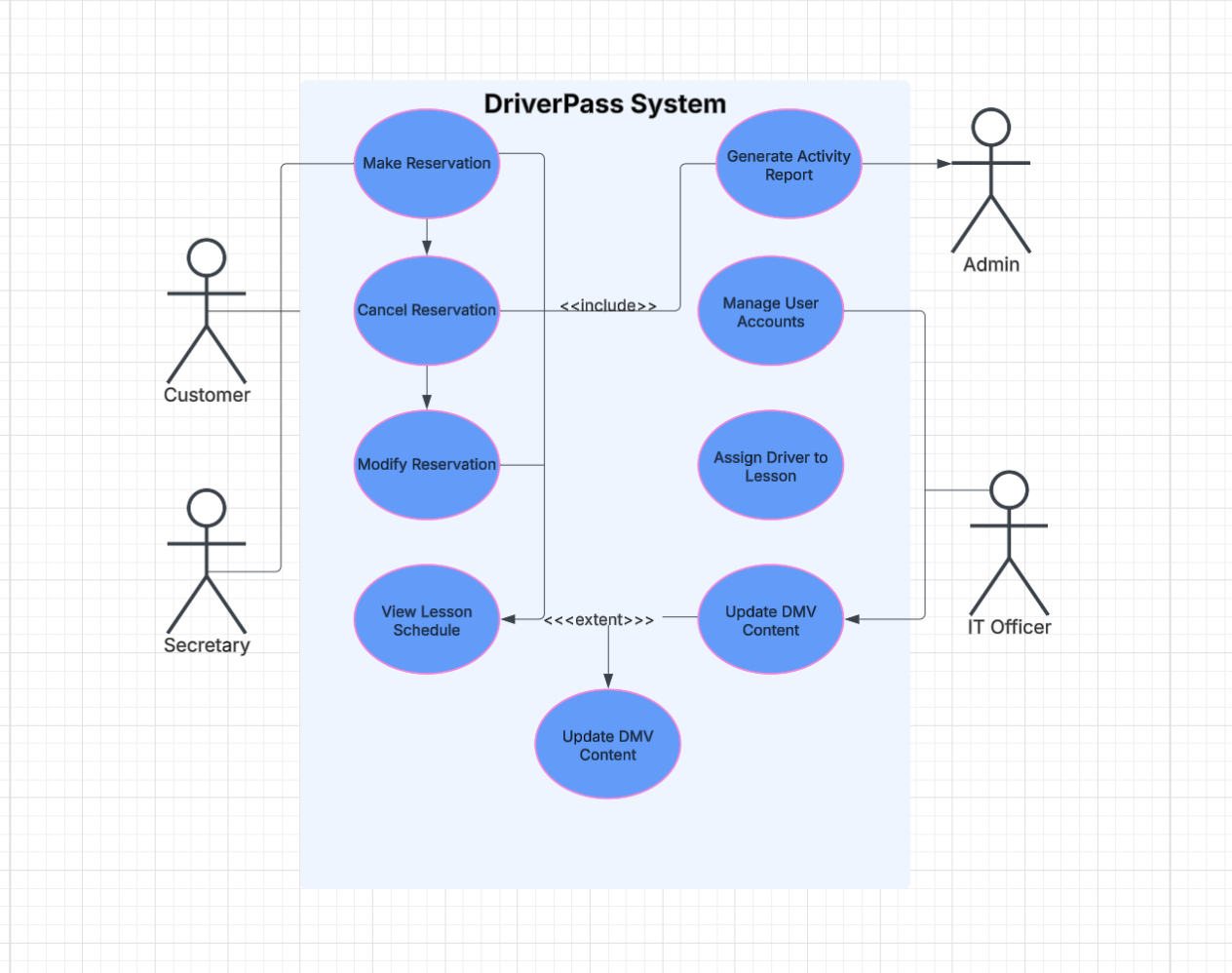
# CS 255 System Design Document Template

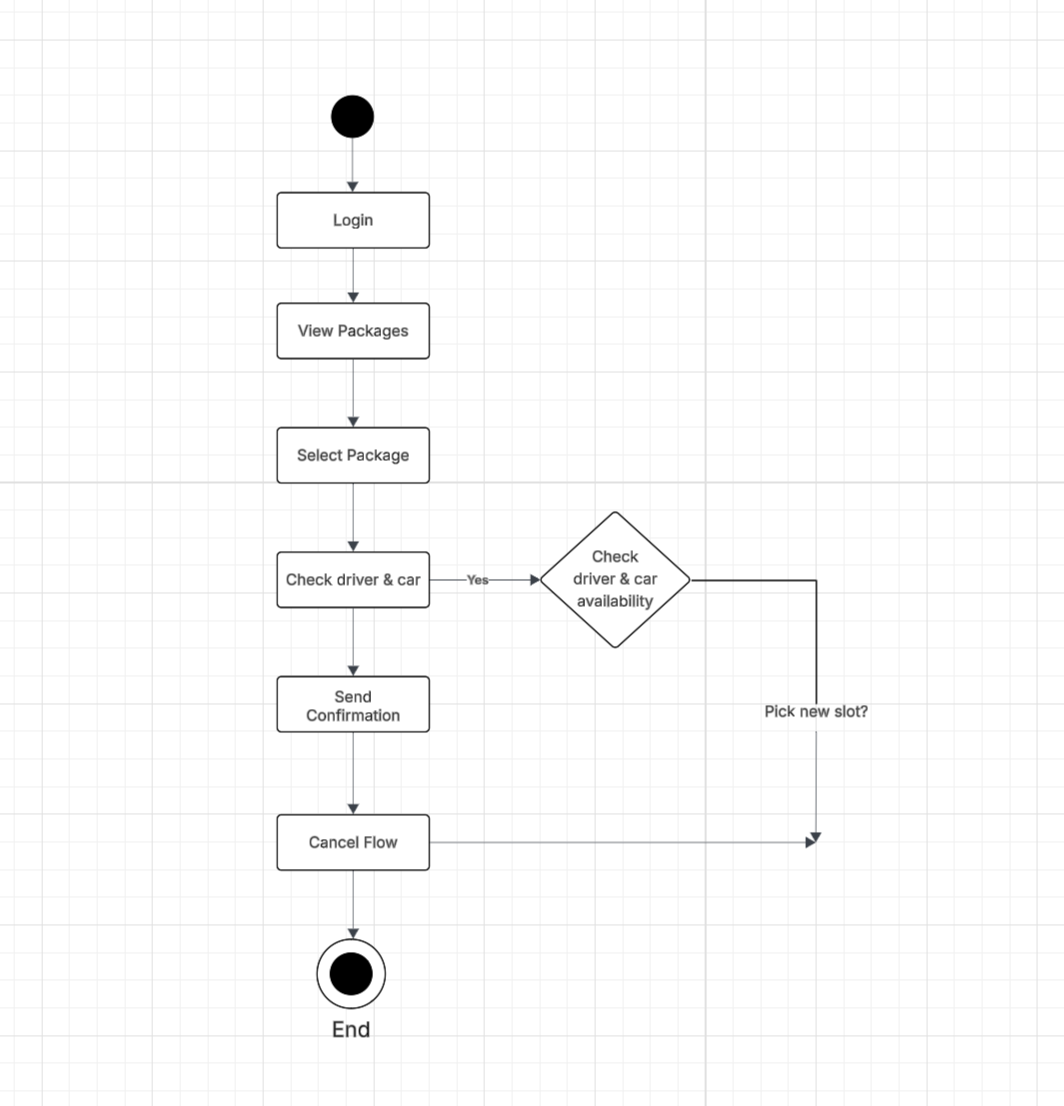
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length 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

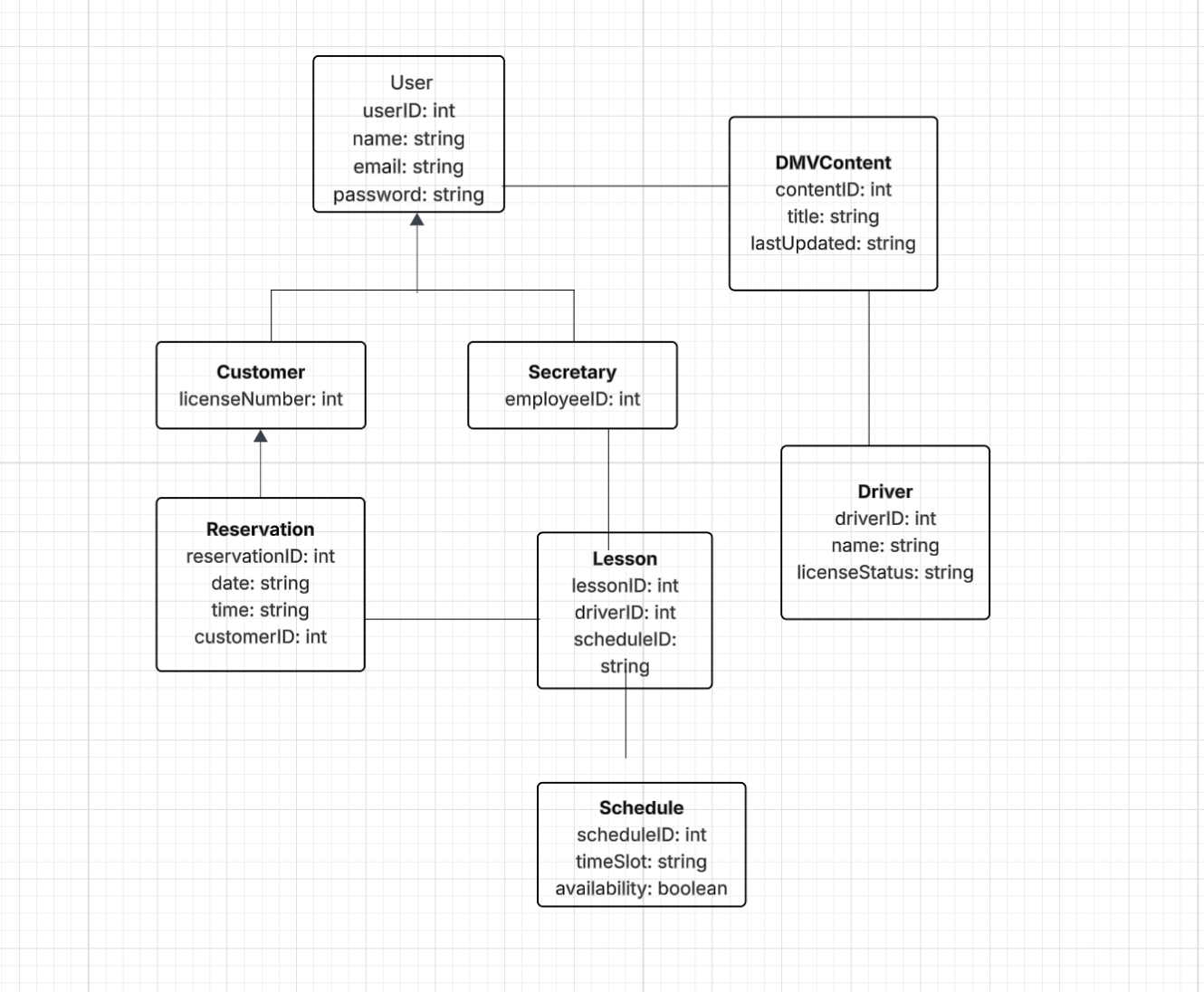


### 

### UML Sequence Diagram

### 

### 4. UML Class Diagram



## Technical Requirements

### Technical Requirements

To run the DriverPass system smoothly and securely, a few important technical components are needed. These include the right hardware, software, tools, and backend setup that align with the system design shown in the diagrams.

**Hardware Requirements**  
The system needs a reliable server (either cloud-based or local) with decent performance — ideally something with at least 8 GB of RAM, a quad-core processor, and 250 GB of SSD storage. Users like customers, secretaries, and admins can access the platform using regular desktops, laptops, or smartphones with an internet connection and a modern browser (like Chrome, Firefox, or Safari). A backup drive or cloud storage should also be in place to keep DMV-related data and reservation records safe in case anything goes wrong.

**Software Requirements**  
The platform itself can be built using a backend framework such as Python with Flask or Django, Java, or Node.js. On the frontend, simple technologies like HTML, CSS, and JavaScript (or a modern framework like React) can be used to make the interface user-friendly. A secure and reliable database system — like MySQL, PostgreSQL, or even MongoDB — will store user information, reservations, lesson schedules, and DMV content.

**Tools**  
To manage the development and system design, version control software like Git should be used (with GitHub or GitLab for collaboration). Tools like Postman can help test any APIs that may be used to integrate third-party systems like a DMV database. Lucidchart can be used for updating and maintaining the UML diagrams. Developers can work with any standard IDE like VS Code, IntelliJ, or PyCharm.

**Infrastructure Requirements**  
The system should be hosted on a secure and scalable platform — cloud providers like AWS, Google Cloud, or Microsoft Azure are great choices. All communication between users and the system must be encrypted using HTTPS with an SSL certificate. Role-based access must be in place so that only authorized users (like Admins, Secretaries, and IT Officers) can access certain functions. Lastly, automatic daily backups and a basic recovery plan should be implemented so that important data isn’t lost.