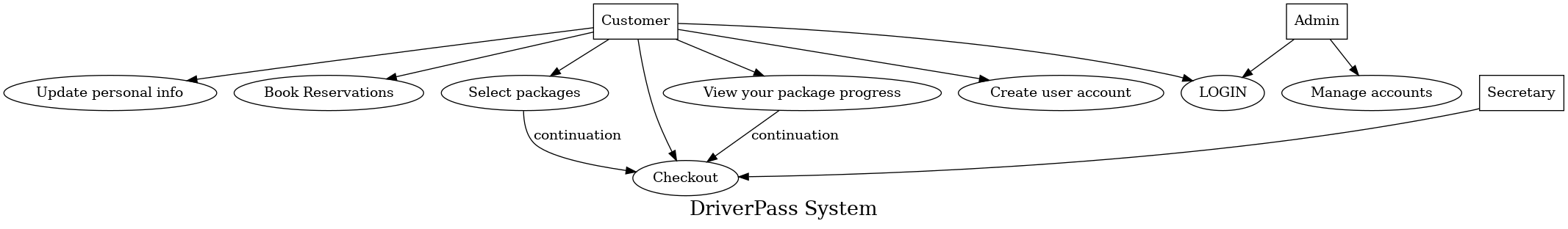
# CS 255 System Design Document Template

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.

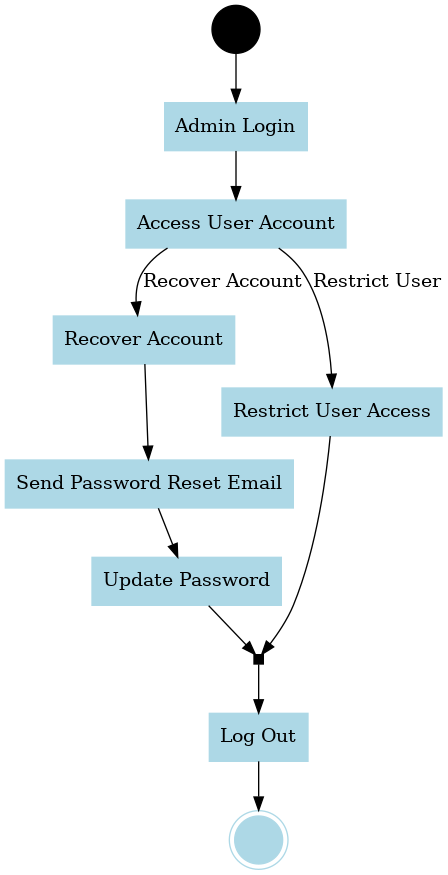
## UML Diagrams

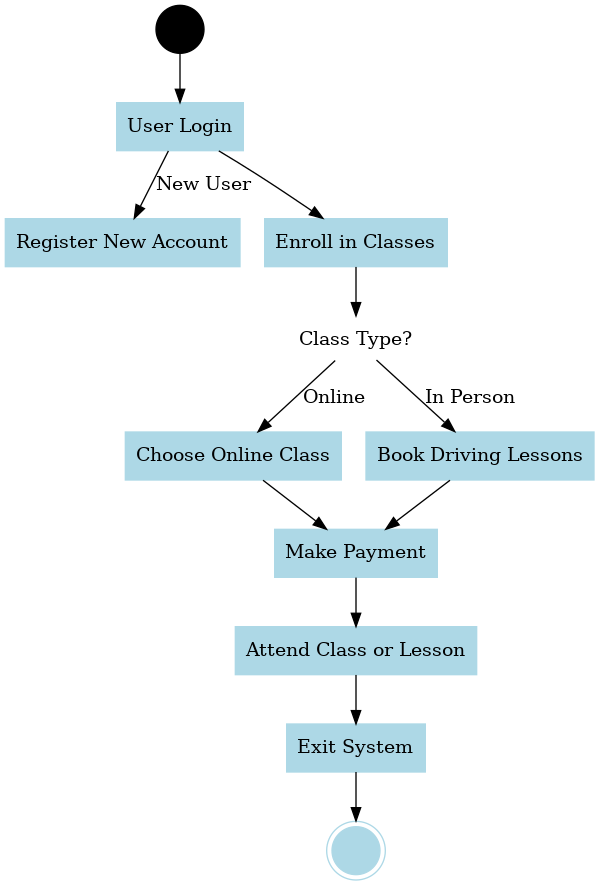
### UML Use Case Diagram



### UML Activity Diagrams

*[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

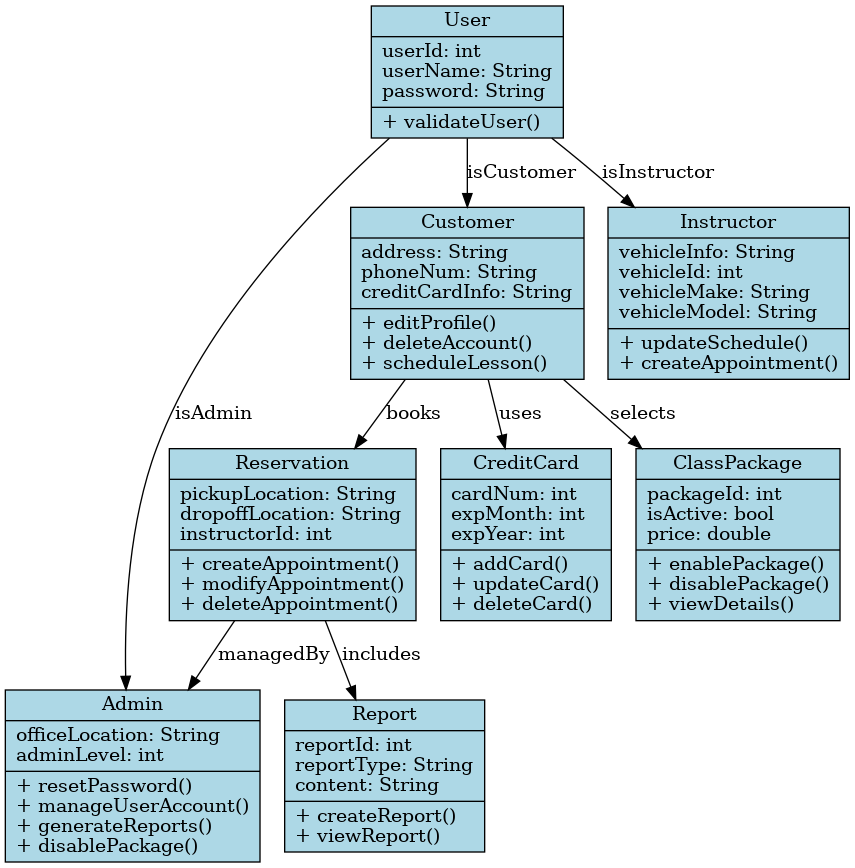
*[You were asked to create a sequence diagram based on* ***one*** *of the use cases you chose. Please insert your sequence diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]*

*A diagram of a driver pass system

Description automatically generated*

### 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

*[Based on the diagrams you have created, describe the technical requirements of your system. These requirements should address the required hardware, software, tools, and infrastructure necessary for your system design.]*

***Hardware Requirements:***

1. ***Server Hardware****: A physical or cloud-based server to host the system, networking equipment such as routers and switches for connectivity, and power sources, including backup power for server uptime.*
2. ***Client Hardware****: Desktop or laptop computers for system access, mobile devices (iOS and Android) for application compatibility, and internet access via wired or wireless connections.*

***Software Requirements:***

1. ***Server Software****: Windows Server or Linux-based server (Ubuntu, Red Hat, etc.), MySQL or PostgreSQL for storing user data, reservations, and reports, AWS, Azure, or Google Cloud to host the application securely, and Apache or NGINX for handling HTTP/HTTPS requests.*
2. ***Client Software****: Web browsers (Chrome, Edge, Firefox) for system access and mobile OS support for Android (v8.0+) and iOS (v12+).*
3. ***Security Tools****: SSL/TLS for encrypted communication and firewall/anti-malware tools to protect against attacks.*

***Development Tools:***

1. ***Programming Languages****: Java, Python, or C# for backend development.*
2. ***Front-end Frameworks****: React.js or Angular for an intuitive user interface.*
3. ***IDE/Editors****: Visual Studio Code, IntelliJ IDEA, or Eclipse.*
4. ***Version Control****: Git for source code management with GitHub or GitLab repositories.*
5. ***Design Tools****: Lucidchart or Draw.io for UML diagrams.*

***Infrastructure Requirements:***

1. ***Cloud Hosting****: Utilize cloud platforms like AWS EC2, Azure App Services, or Google Cloud Compute for high availability and scalability.*
2. ***Database Management****: Cloud-based managed databases (e.g., Amazon RDS, Google Cloud SQL) for improved reliability and backups.*
3. ***Backup and Recovery****: Daily backups of system data to a secure location to ensure data integrity in case of failures.*
4. ***Network Infrastructure****: High-speed internet connectivity for seamless user interactions and system operations and load balancers to distribute incoming traffic efficiently.*