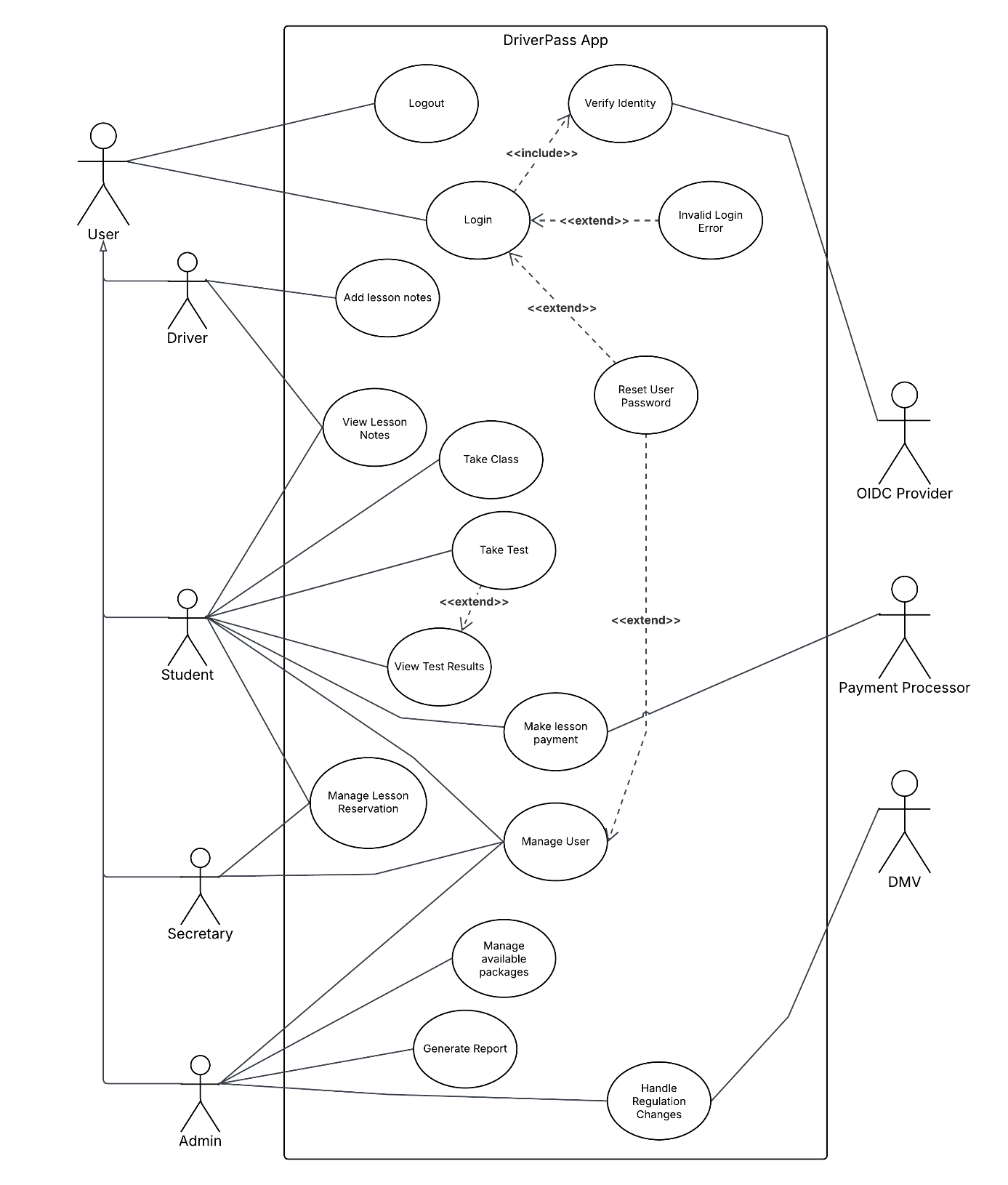
# CS 255 System Design Document: DriverPass

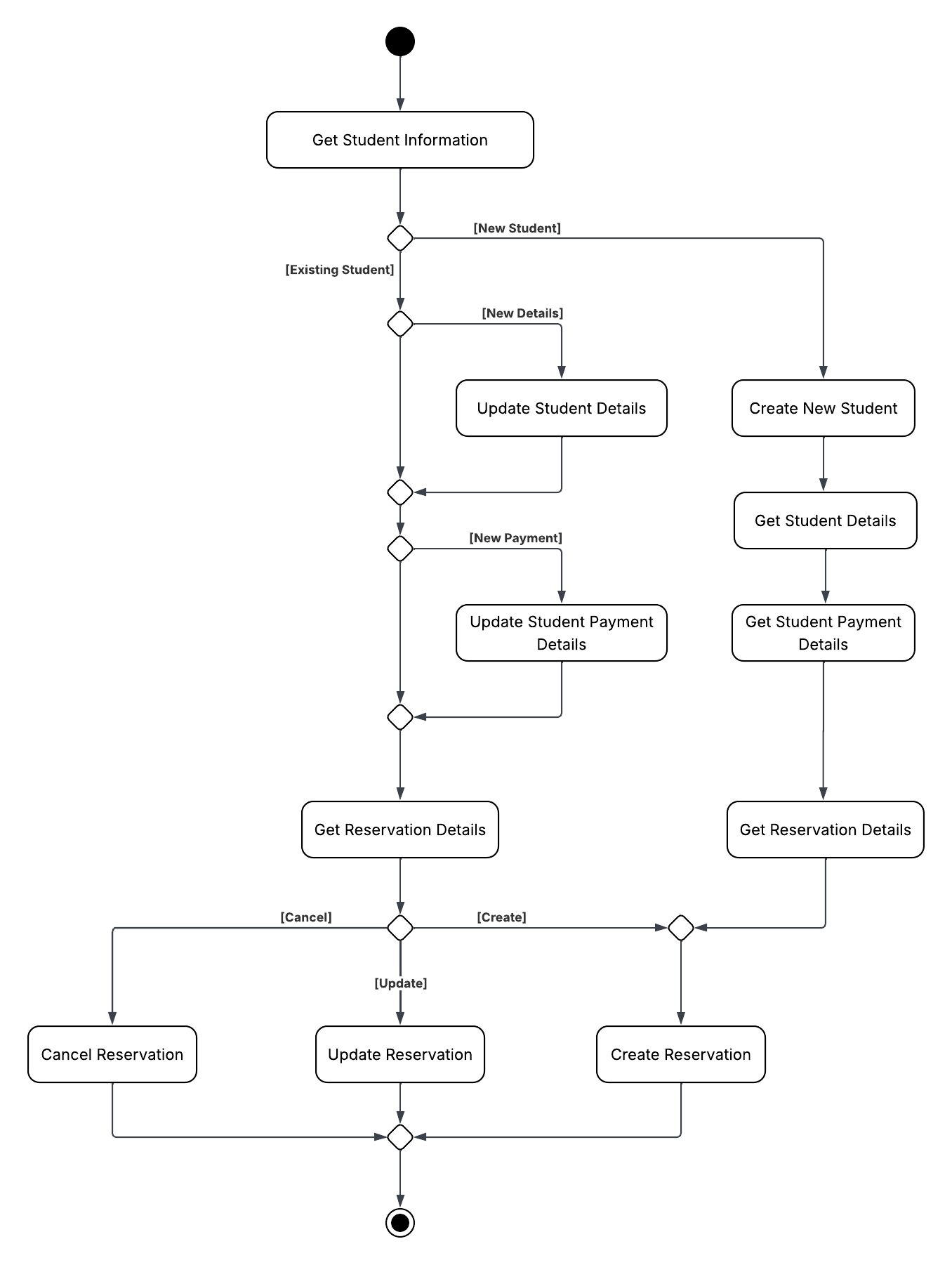
## UML Diagrams

### UML Use Case Diagram

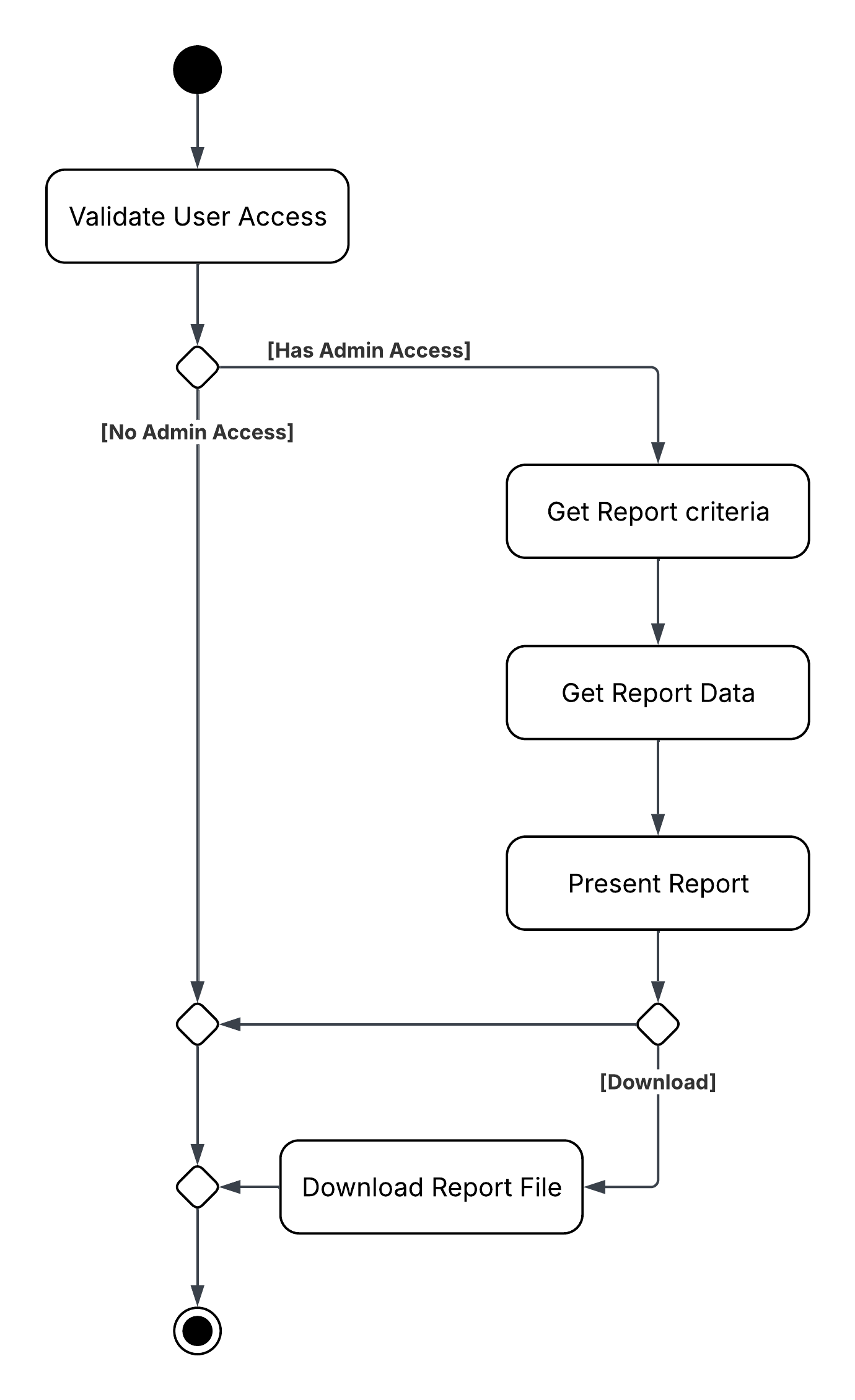


### UML Activity Diagrams

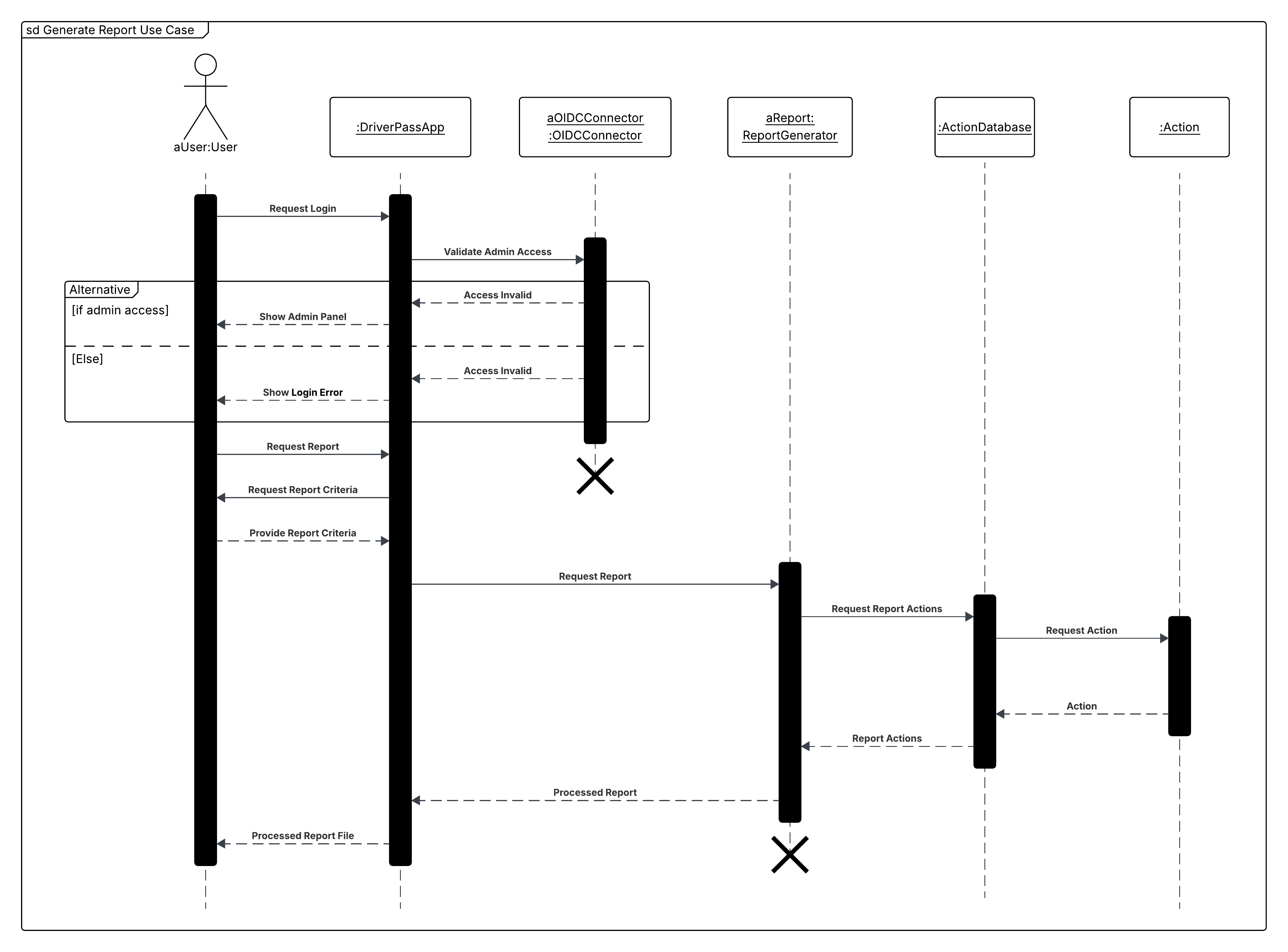
Manage Reservation



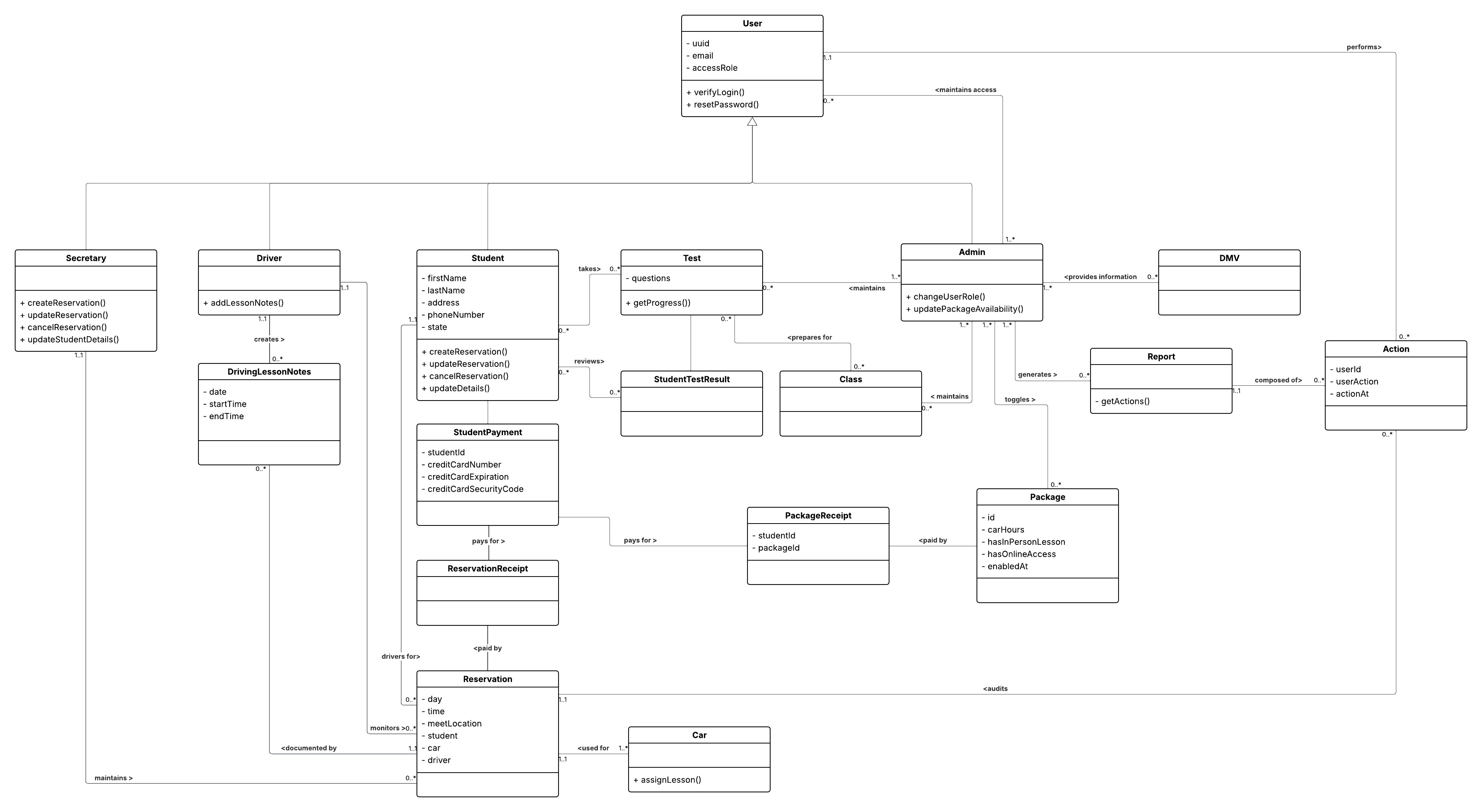
Generate Report



### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

Based on the information provided by client the following are technical requirements of this system:

* **Amazon Web Services Cloud-Based Infrastructure**: The client indicated their involvement for the infrastructure should be limited and explicitly stated that the infrastructure should be hosted in a cloud environment. This will reduce the amount of overhead required for client maintenance and monthly cost. Additionally, creating a scaling architecture is well supported through features such as the Elastic Container Service.
  + **Cognito** - Will be utilized to handle user access and account management
  + **S3 -** Utilize to store static files for the site and files containing information supplied by the DMV
  + **CloudFront** - Will be utilized to handle distributed access by users and prevent server load in preparation for scalability
  + **Elastic Cloud Compute running Ubuntu Server LTS Version**: The backend operating system for the application server will be on the latest Ubuntu LTS hosted on a virtual server within the AWS infrastructure
  + **Relational Database Service (PostGreSQL)**: The database for the application will utilize PostGreSQL for storing application information via Amazon RDS service. This information will include all classes outlined in the UML class diagram.
* **Supported Browsers:** Due to widespread availability and for greatest user access the following browsers will be supported with versions not to exceed 6 months of age:
  + Microsoft Edge
  + Google Chrome
  + Mozilla Firefox
  + Safari
* **REST Based API layer**: Due to the information being relatively static and low update frequency, the backend server will be implemented with a REST-ful API over a GraphQL based approach.
* **Require HTTPS:** To maintain secure connection between the client-server communication over HTTP will be disabled in the infrastructure.
* **HTML/CSS/JavaScript Client**: Due to the limited reactivity required on the client interface, a simplistic client will be created without the utilization of a UI framework.
* **Private GitHub Repository:** The source code will be hosted in a private GitHub repository that handles deployments via GitHub Actions for code changes to the system.