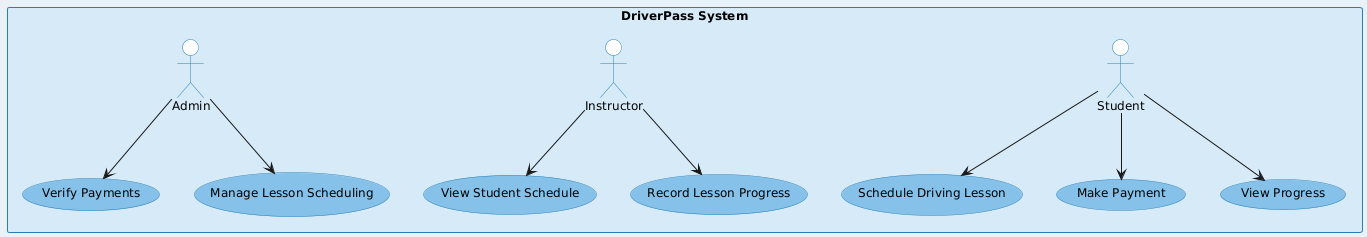
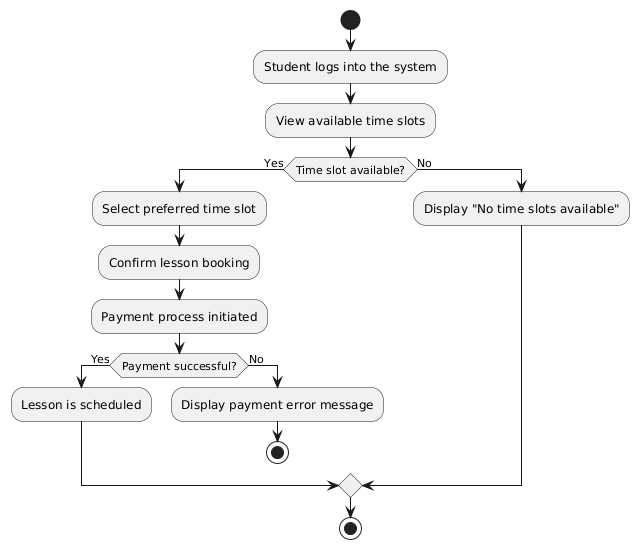
# CS 255 System Design Document

## UML Diagrams

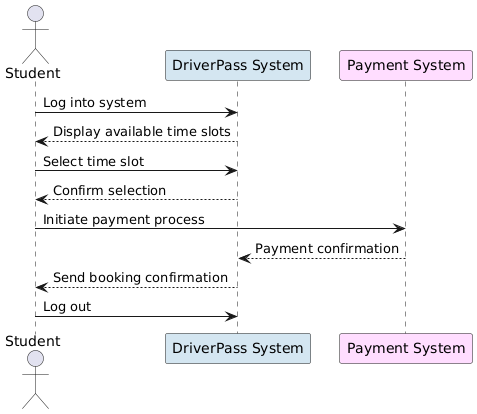
### UML Use Case Diagram



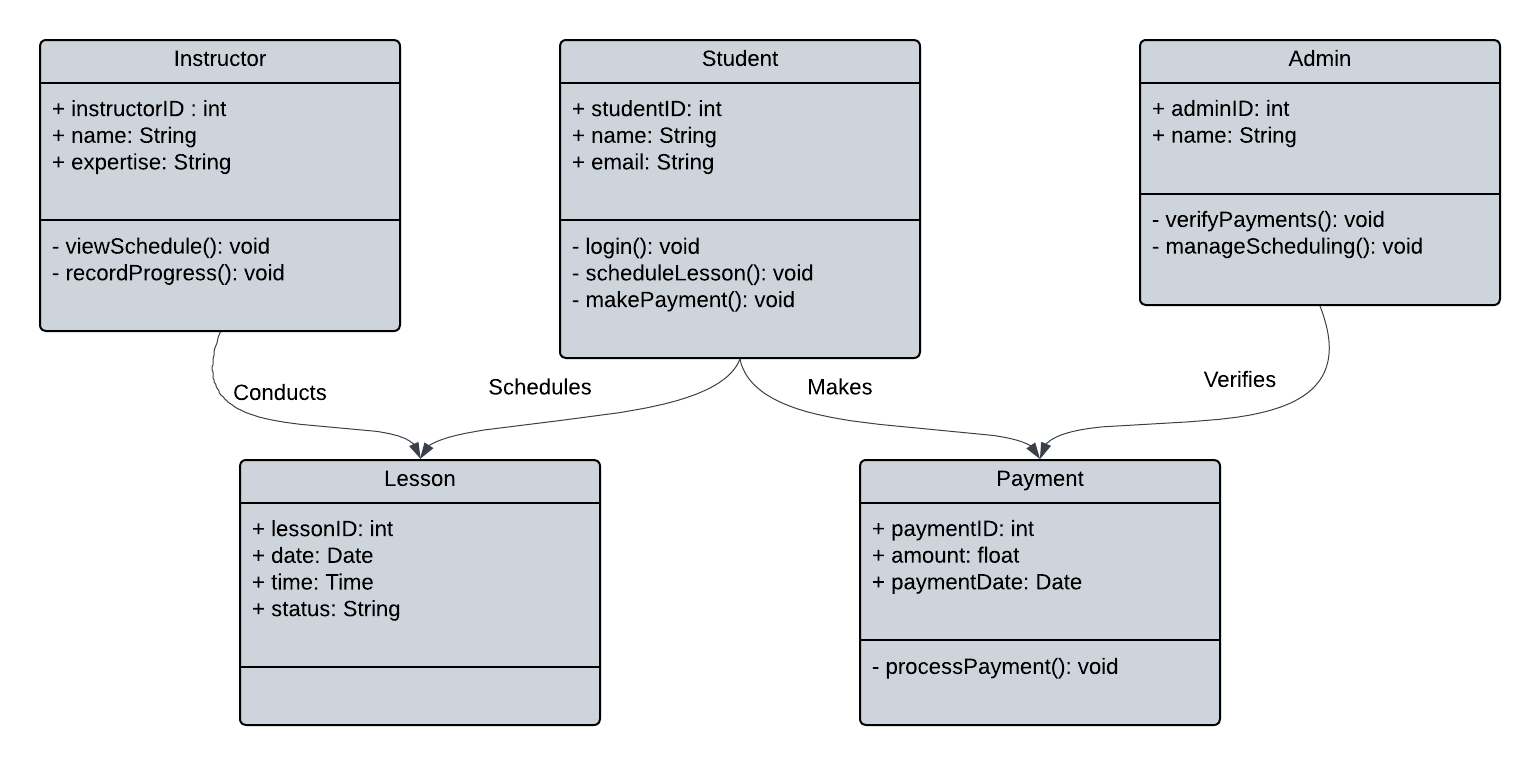
### UML Activity Diagrams



### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

**1. Hardware Needs**

To run the system, you’ll need:

* For the Server:
  + A powerful computer (server) to handle many users at once.
  + Recommended specs:
    - Processor: Fast, multi-core processor.
    - RAM: At least 32GB.
    - Storage: Fast hard drive with at least 1TB of space.
    - Internet: A strong connection (1Gbps or higher).
  + Backup servers to prevent data loss if something goes wrong.
* For Users:
  + A computer, laptop, tablet, or smartphone with internet access.
  + At least 4GB of RAM to run the system smoothly.

**2. Software Needs**

The software needed to build and run the system includes:

* For the Server:
  + Operating System: Linux because it’s reliable and can be secure.
  + Web Server: Nginx to handle requests.
  + Programming Language:
    - Java or Python.
  + Database:
    - MySQL to store user and system information.
  + Payment Processing: A payment system for handling payments securely.
* For the User Side:
  + Languages: HTML, CSS, and JavaScript.
  + Tools: React.js to build a user-friendly website.
  + Mobile Apps: Use React Native to make the system work on phones and tablets.
* For Security:
  + SSL/TLS certificates for safe, encrypted communication.
  + Secure login systems to protect accounts.
  + Firewalls to block unwanted access.
* Development Tools:
  + Visual Studio Code for coding.
  + Git to save and manage code changes.

**3. Infrastructure and Tools**

To run and manage the system, you’ll need:

* Cloud Hosting:
  + A provider like AWS or Microsoft Azure to store and run the system online.
  + Tools to scale resources up or down depending on how many people use the system.
* For Updates and Testing:
  + Tools like GitHub Actions to automatically test and update the system.
* For Monitoring and Logs:
  + Tools to check system performance.
  + Logging tools to track errors and fix issues.
* For Team Work:
  + Use Lucidchart to design diagrams.
  + Jira to manage tasks and team progress.