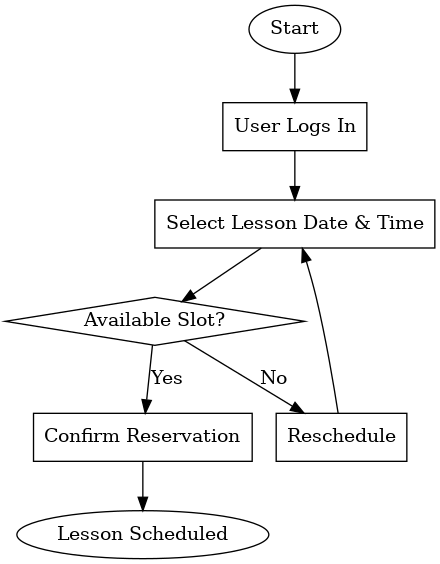
# **CS 255 System Design Document**

## **UML Diagrams**

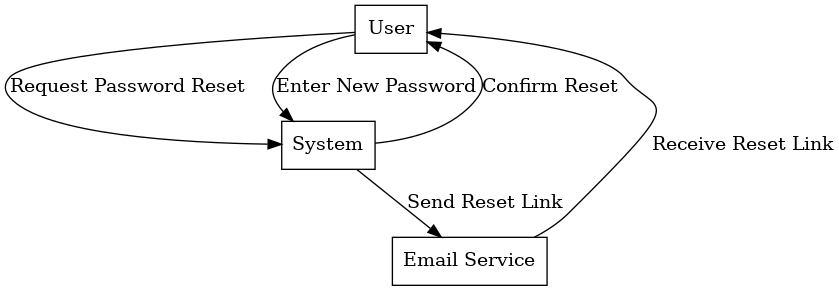
### **UML Use Case Diagram**

### 

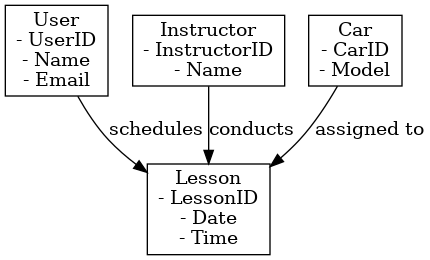
### **UML Activity Diagrams**



### **UML Sequence Diagram**



### **UML Class Diagram**



## **Technical Requirements**

### **Hardware Requirements**

* **Server:** Cloud-based or Local Server that hosts the system backend.
* **Storage:** Database with high-level security for storing user data and transaction records.
* **Network:** Stable internet connection with sufficient bandwidth.
* **Client Devices:** Desktops, laptops, and mobile devices for users.

### **Software Requirements**

* **Operating System:** Windows/ Linux for server. Clients will use Windows, MacOS, iOS, and Android.
* **Database:** MySQL, PostgreSQL, or MongoDB for user and transaction records.
* **Backend Framework:** Java, Python, or JavaScript for backend integration.
* **Graphic Framework:** User Interfaces built using React.js, Angular, or Vue.js.
* **Web Hosting:** For scalable deployment, use AWS, Azure, or Google Cloud.

### 

### **Tools**

* **Lucidchart:** For creating UML diagrams.
* **Development IDE:** VS Code, Eclipse, or IntelliJ for coding.
* **Version Control:** GitHub or GitLab for source control.
* **Testing Tools:** Selenium or JUnit for testing purposes.

### **Infrastructure**

* **Cloud Hosting:** Managed services to ensure high availability.
* **Security Measures:**
  + Access Control by Policies: Role-based access control, e.g., Admin, IT Officer, Secretary, and Customers.
  + Encryption of data during transmission using SSL.
  + Two-step verification (2FA) for users.
  + Audit logs are used to keep track of changes made to the records.
* **Backup System:** Automated backups at regular intervals to mitigate data loss.
* **APIs:** Communication between parts of different systems is done through RESTful APIs.
* **Scalability:** The capacity to cope with ever-increasing customer usage and additional data.

## **Functional Requirements**

* Users are supposed to register, log in, and safely recover their passwords.
* Customers are expected to be able to book driving lessons online as well as cancel or change them.
* The system shall be able to identify the driver, car, time slot, and other reservation particulars.
* Administrators are expected to manage user accounts with some restrictions: password reset and account deactivation.
* The system shall notify users and capture new DMV policies and test updates.

## **Non-Functional Requirements**

* **Security**: The system shall implement access control using roles and store all sensitive data in an encrypted format.
* **Performance:** The system shall support at least 500 concurrent users or 1,000 active users simultaneously with no lag.
* **Availability:** The system shall provide at least 99.9% operational uptime, with an automated backup procedure for retrieval.
* **Flexibility:** The system shall facilitate changing the composition of lesson packages and test structures in the future without expending too many development resources.