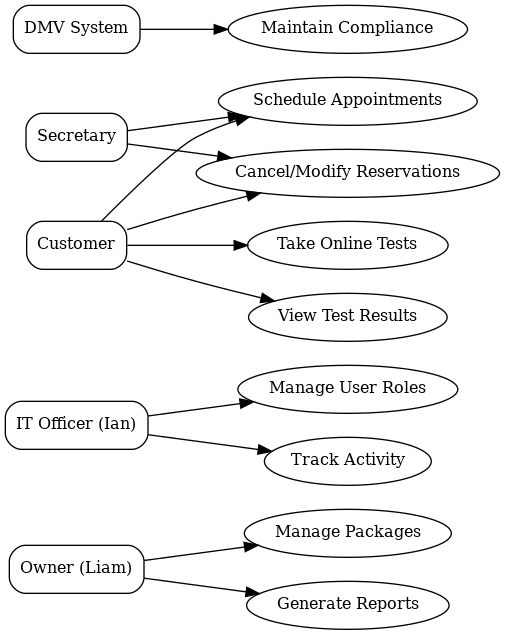
# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram



### UML Activity Diagrams

***UML Activity Diagram 1 (Use Case: Schedule Driving Lessons)***

A diagram of a system

Description automatically generated

**UML Activity Diagram 2 (Use Case: Take Online Test)**

A diagram of a system

Description automatically generated with medium confidence

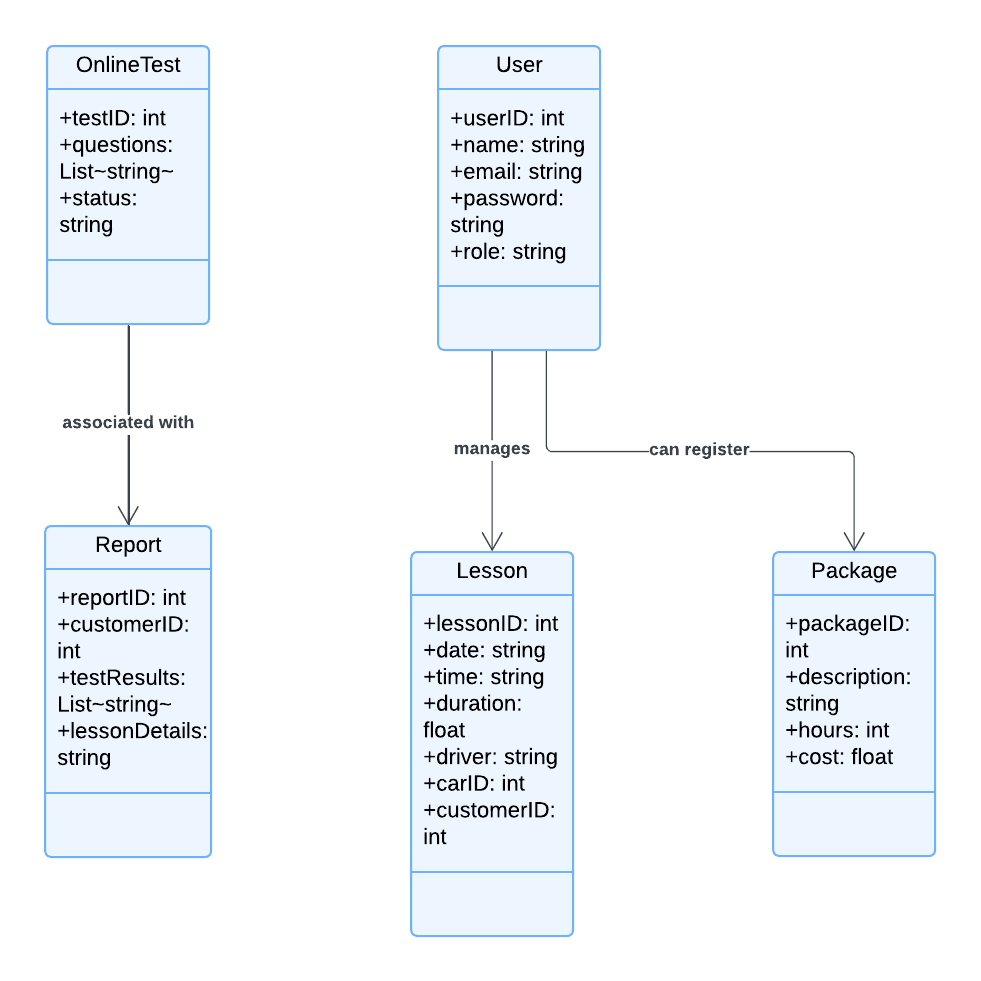
### UML Sequence Diagram

*UML Sequence Diagram (Use Case: Schedule Driving Lessons)*

A diagram of a software process

Description automatically generated

### UML Class Diagram



## Technical Requirements

**Hardware Requirements**

**Client Devices:**

The system requires hardware from client devices. Desktop and laptop computers should have at least 4GB of RAM and a dual-core CPU to simplify scheduling, communication, and resource management for customers, secretaries, administrators, and trainers. Cloud and real-time functions demand a constant internet connection. Cellphones and tablets need 2GB RAM for mobile access. For flexible use, the platform is available on iOS and Android smartphones.

**Server Infrastructure:**

Server infrastructure is essential for reliable and scalable systems. The platform will need dedicated or cloud-hosted servers with enough processing power for many concurrent users. For real-time user interactions, database queries, and content delivery, servers need 16-core processors and 32GB RAM. Manage the growing number of class schedules, user accounts, and test results with at least 2TB of scalable storage. Backup servers will provide redundancy and data recovery in case of system failure, improving platform resilience and disaster recovery.

**Software Requirements**

**Client-Side Software:**

To provide multi-device access, the platform must be client-side compatible with Chrome, Firefox, and Safari. For responsive, dynamic websites, these browsers should support HTML5 and CSS3. Android (8.0+) and iOS (12.0+) apps will enable platform access on the go. On smartphones and tablets, these apps will let users access class schedules, interact with educational content, and complete assignments, expanding the system's reach.

**Server-Side Software:**

Linux, especially Ubuntu Server for reliability, security, and cost, will power the platform. The platform will use MySQL or PostgreSQL to store user profiles, lectures, and exam results. These databases are great for scaling and querying as users grow. Apache and Nginx, popular and trustworthy web servers, deliver online pages and handle massive traffic. Backend development uses Python or Java frameworks like Django or Flask for security, maintainability, and scalability.

**Tools and Frameworks**

**Development Tools:**

System development will use IDEs like VS Code, IDEA, and PyCharm. These IDEs have code auto-completion, debugging, and version control to speed up coding. Version control with Git and platforms like GitHub or GitLab facilitates collaborative development and code management. Swagger or Postman can document and test API endpoints before production to assure security.

**Testing and Monitoring Tools:**

UI testing will use Selenium while backend testing will use JUnit to validate platform functionality. These methods enable continuous development testing to find bugs early and preserve system integrity. New Relic or Datadog will monitor platform health after implementation. These technologies monitor system performance in real time, allowing engineers to find problems and make proactive improvements.

**Infrastructure Requirements**

**Cloud Services:**

As user demand grows, AWS, Azure, or Google Cloud will host the platform, simplifying scaling and resource management. These flexible, affordable services let you add computer power or storage as needed. A Content Distribution Network (CDN) will speed up content distribution across geographical regions, reducing latency and improving user experience, especially globally.

**Network Infrastructure:**

SSL/TLS encryption protects sensitive data in all platform-user transactions. Platform firewalls and IDS will block unwanted access and detect threats in real time. This protects user data from hackers and breaches.

**Backup and Recovery:**

Daily cloud backups of databases and system files secure and retrieve platform data. This method reduces data loss and ensures business continuity after system failure. Disaster recovery plans involve platform restoration after major failures. This will reduce downtime and let users resume using the system without losing data.

**Security Requirements**

**Authentication and Access Control:**

To protect user data, the platform will use token-based OAuth 2.0 authentication. Only authorized users can access sensitive platform functionality and data. To restrict system access to important information for employees, administrators, and trainers, RBAC will use user roles. Security audits find and address platform vulnerabilities, lowering risks and preserving user data. Consumers may trust the platform without worrying about data breaches or unauthorized access with these security features.