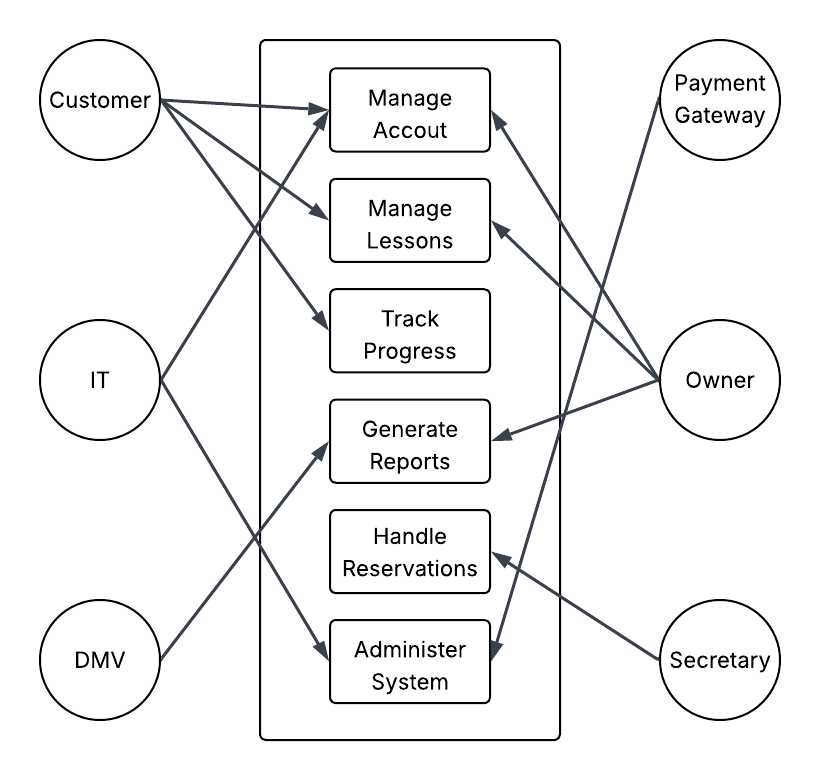
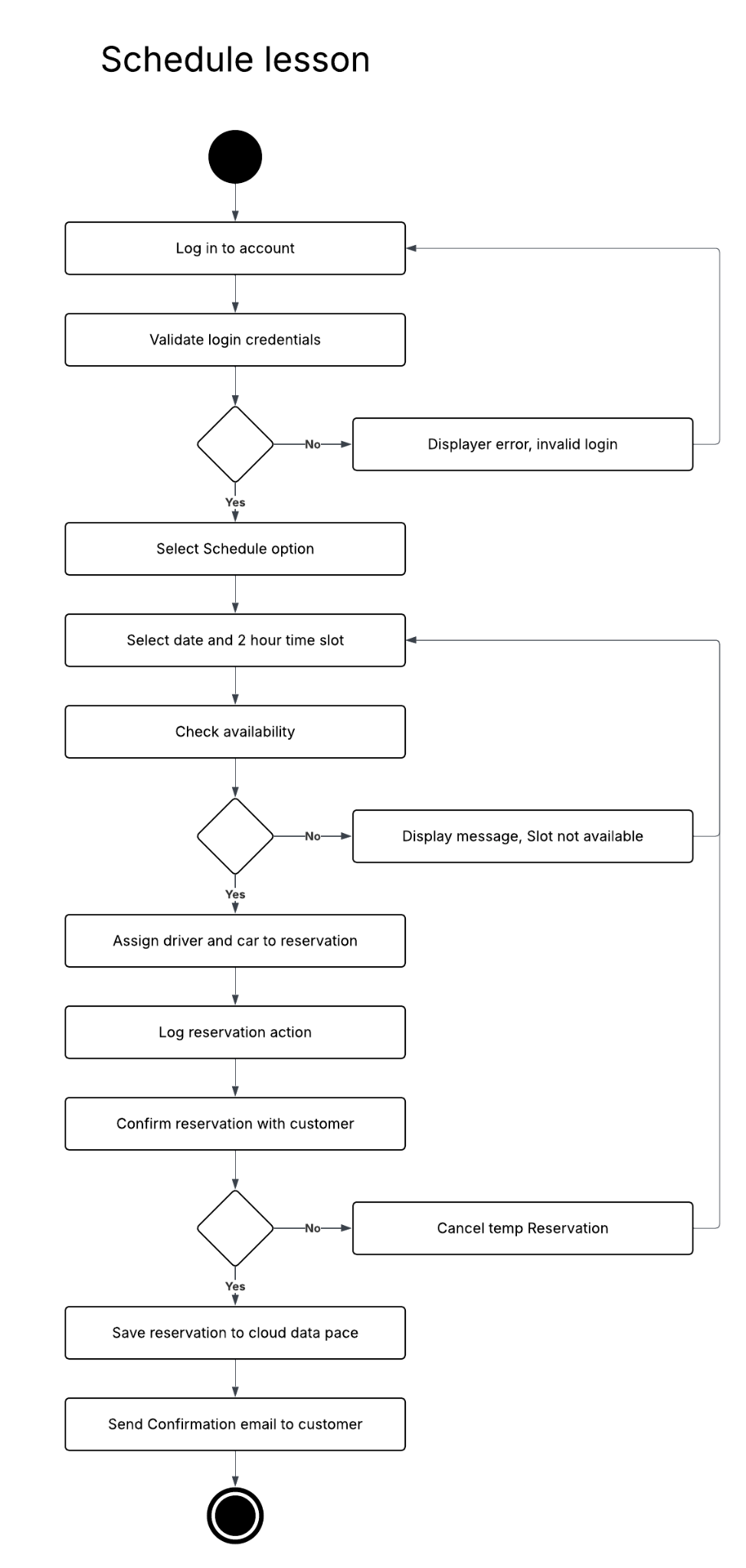
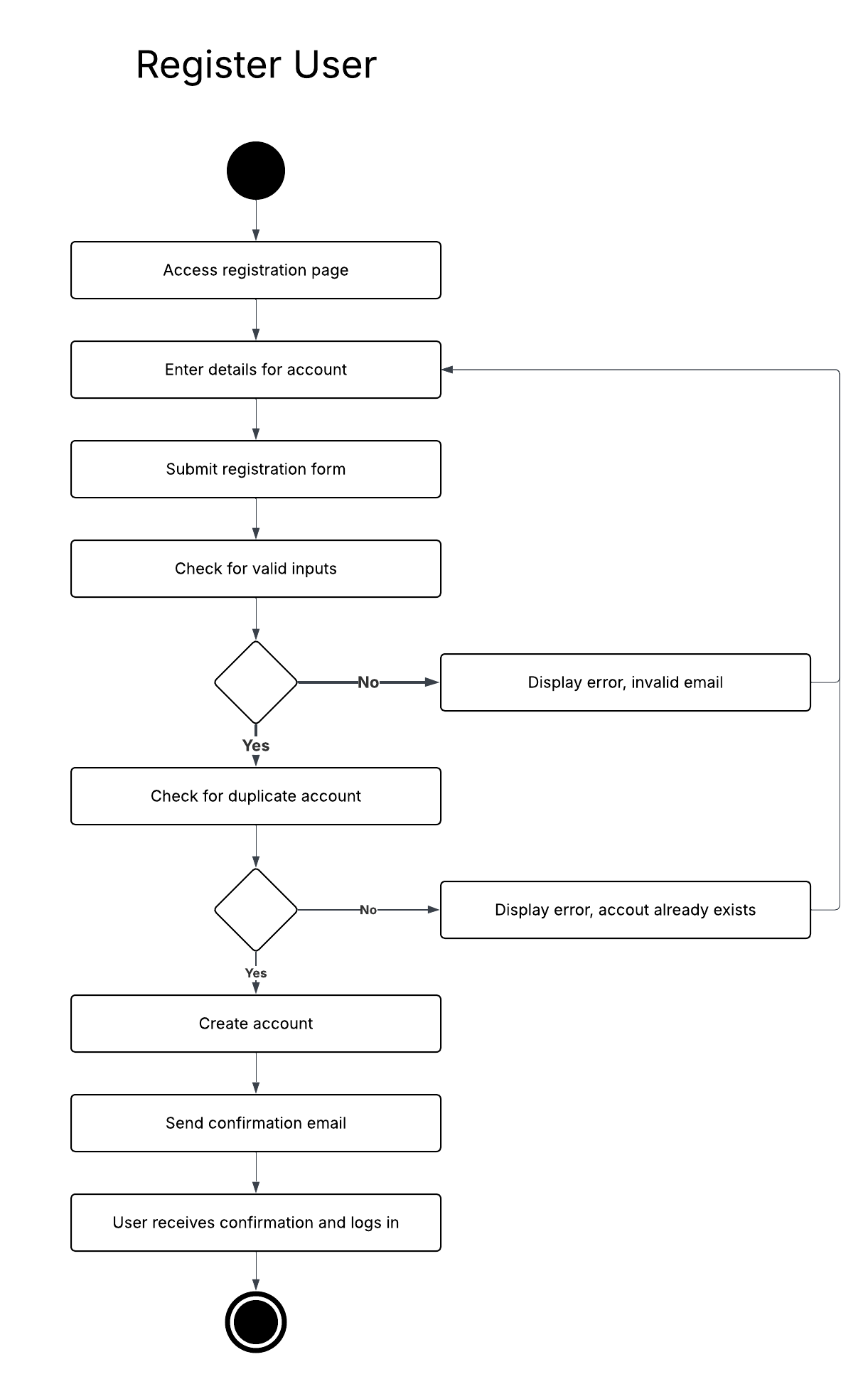
# CS 255 System Design Document Template

## UML Diagrams

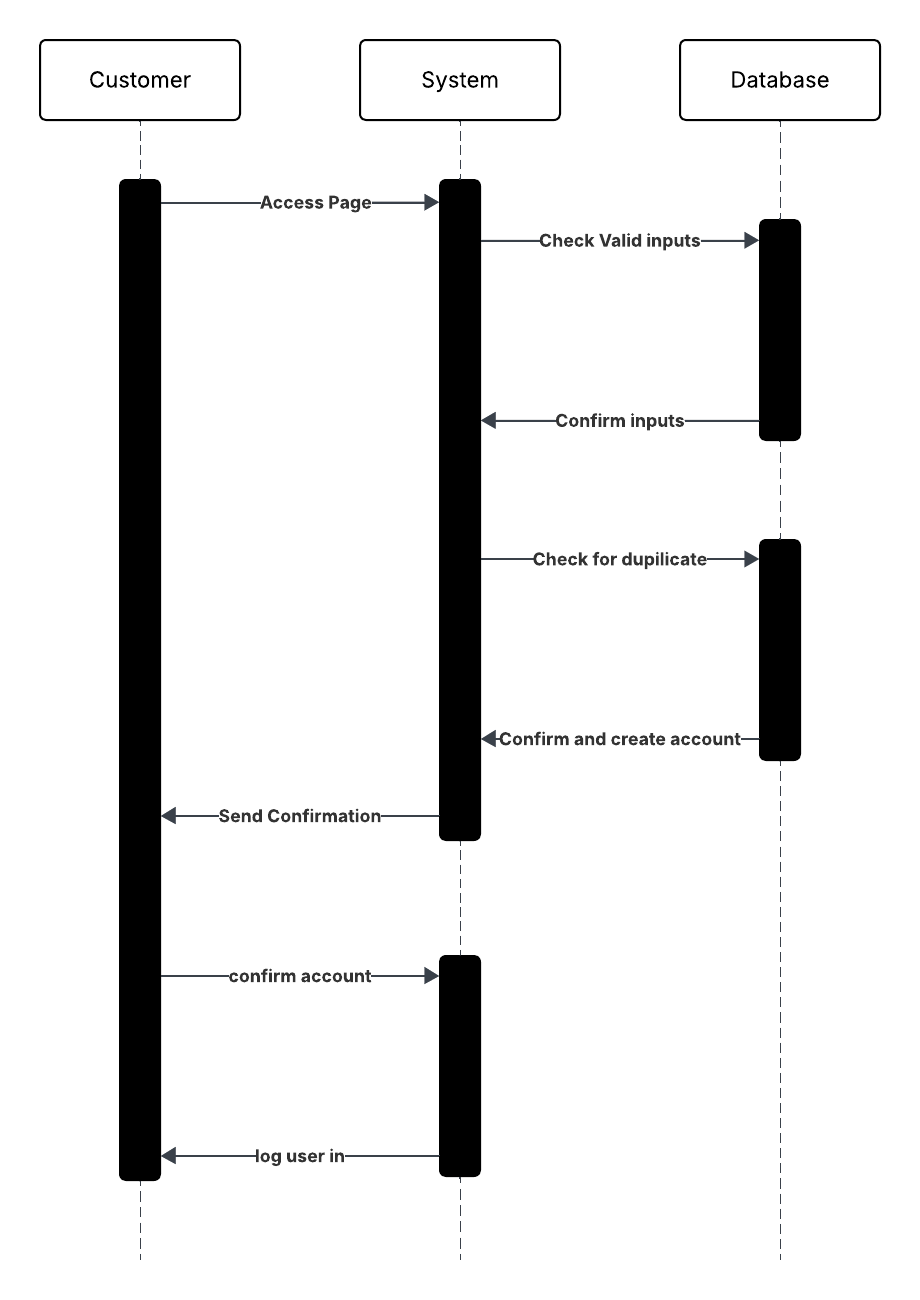
### UML Use Case Diagram

**

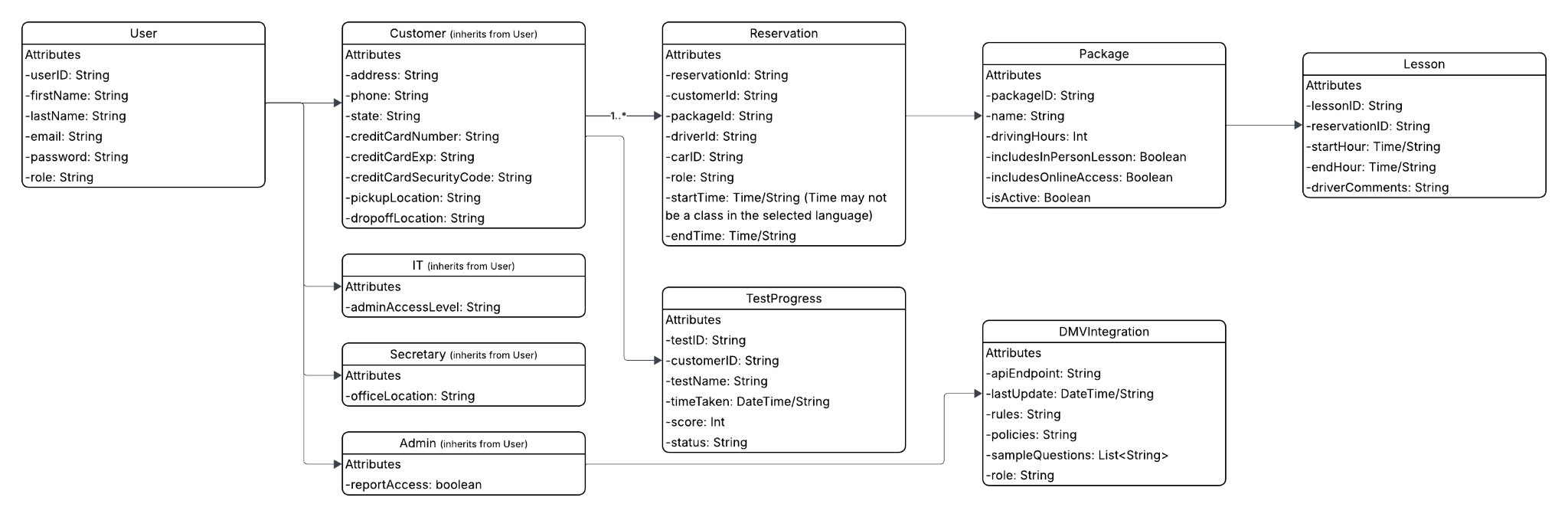
### UML Activity Diagrams



### UML Sequence Diagram



### UML Class Diagram

**

## Technical Requirements

## Hardware

* **Cloud Servers**: The system requires scalable cloud servers to handle user traffic, data storage, and processing for registration, scheduling, and DMV integration. Servers must support a minimum of 10 concurrent drivers and an initial user base of 100 customers, scalable to accommodate growth.
* **No Local Hardware**: No on-premises hardware is needed, as the system is cloud-hosted to minimize technical management, aligning with Ian’s requirement for minimal technical overhead.
* **User Devices**: Users need devices with internet access to access the web based interface, assuming reliable internet.

## Software

* **Web Browsers**: The system must be accessible via modern web browsers with HTML5 and CSS3 support for responsive design, ensuring compatibility across devices
* **Operating Systems**: No specific OS is required for users, as the system is browser-based. The cloud platform must support a Linux-based server environment for hosting
* **Database Management System**: A relational database is required to store encrypted customer data (name, credit card), reservations, test progress, and activity logs, supporting the class diagram’s entities.
* **Web Server Software**: A web server is needed to handle HTTP requests and serve the web interface, supporting HTTPS for secure connections as specified in the security requirements.

## 

## Tools

* **Development Frameworks**: Use JavaScript frameworks to build the responsive web interface, supporting role specific dashboards as described in the interview and user interface requirements.
* **API Integration**: An API client is required for DMV integration to fetch real time updates on rules, policies, and sample questions. The API must support secure communication.
* **Payment Gateway**: A secure payment processing tool is needed to handle credit card transactions during registration, ensuring encryption as per security requirements.
* **Authentication Tools**: Implement OAuth 2.0 or JWT (JSON Web Tokens) for role based access control, distinguishing users.
* **Logging and Monitoring**: Use tools like ELK Stack or cloud native logging to log all actions for auditing.

## Infrastructure

* **Cloud Platform**: Deploy the system on a cloud provider to ensure scalability, automated backups, and 99.9% uptime, aligning with the nonfunctional requirement for cloud hosting and Ian’s focus on minimal technical management.
* **Security Infrastructure**: Implement HTTPS with SSL/TLS certificates for secure data transmission, and encrypt sensitive data in the database using AES-256. Firewalls and intrusion detection systems protect against brute force attacks.
* **Scalable Architecture**: Use a microservices architecture to separate functionalities, allowing independent scaling and maintenance, supporting the adaptability requirement for browser/OS updates.
* **Notification System**: Integrate an email service for sending confirmation emails and admin notifications, as required by the sequence diagram and interview.
* **Content Delivery Network (CDN)**: Use a CDN (such as Cloudflare) to optimize web interface performance, ensuring reservations and logins process within 5 seconds and DMV data pulls complete in under 10 seconds, as per performance requirements.