# CS 255 System Design Document

## UML Diagrams

### UML Use Case Diagram

*A computer screen shot of a diagram

AI-generated content may be incorrect.*

### UML Activity Diagrams

*[You were asked to choose* ***two*** *use cases and create* ***two*** *activity diagrams, one for each use case. Please insert* ***both*** *of your activity diagrams here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s needs.]*

*A diagram of a credit card

AI-generated content may be incorrect.*

*A diagram of a computer program

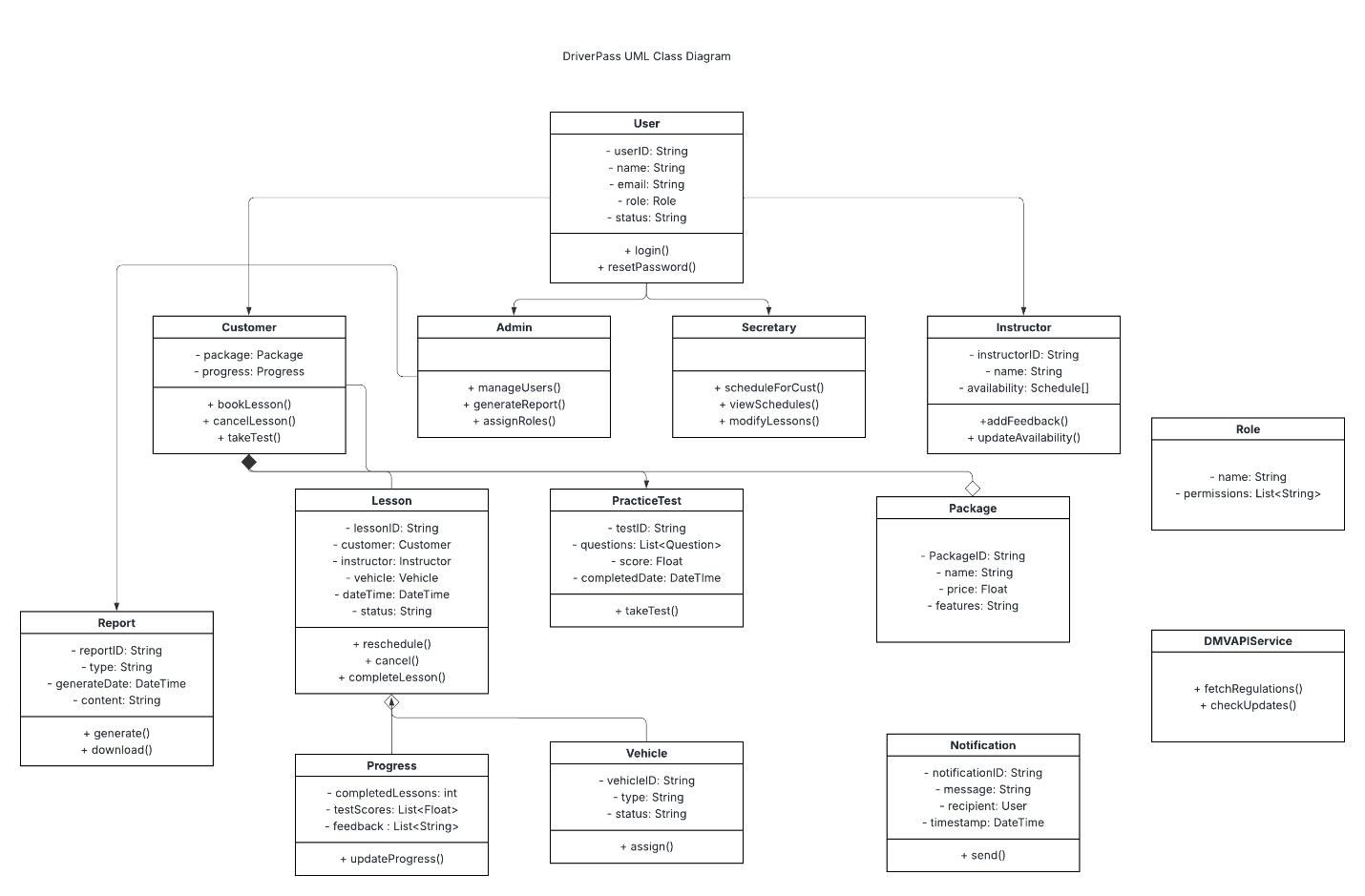
AI-generated content may be incorrect.*

### UML Sequence Diagram

A screenshot of a computer

AI-generated content may be incorrect.

### UML Class Diagram



## Technical Requirements

1. *Hardware Requirements*
   1. *Device with internet capability (Desktop, Laptop, Tablet, or Smartphone)*
   2. *Cloud-hosted server or dedicated on-prem server*
2. *Software Requirements*
   1. *Supported browsers: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge*
   2. *Backend Language: Node.js or Java*
   3. *RESTful API design for frontend-backend communication*
   4. *Database: MySQL or PostgreSQL*
   5. *Authentication & Role-based Access Control (RBAC)*
3. *Development Tools*
   1. *Visual Studio Code / IntelliJ IDEA*
   2. *Postman (API testing)*
   3. *Git & GitHub/GitLab for version control*
   4. *Lucidchart*
4. *Hosting & Deployment*
   1. *Cloud service provider: AWS, Azure, or Google Cloud*
   2. *Compute: EC2 (AWS) or App Service (Azure)*
   3. *Storage: S3/Blob for documents/reports*
   4. *Database: RDS (AWS) / Azure SQL*
5. *Security Infrastructure*
   1. *HTTPS using SSL/TLS*
   2. *Authentication system with password encryption*
   3. *Brute-force protection and activity logging*
   4. *Role-based access using the Role class and permissions*
6. *APIs & Integrations*
   1. *DMV API integration (as shown in DMVAPIService)*
   2. *fetchRegulations()*
   3. *checkUpdates()*
   4. *Notification system (Email/SMS integration using SendGrid or Twilio)*
7. *Monitoring & Maintenance*
   1. *Logging & auditing tools (e.g., Loggly, ELK stack)*
   2. *Scheduled backups (daily) and restore mechanisms*
   3. *CI/CD pipeline using GitHub Actions / GitLab CI / Azure DevOps*