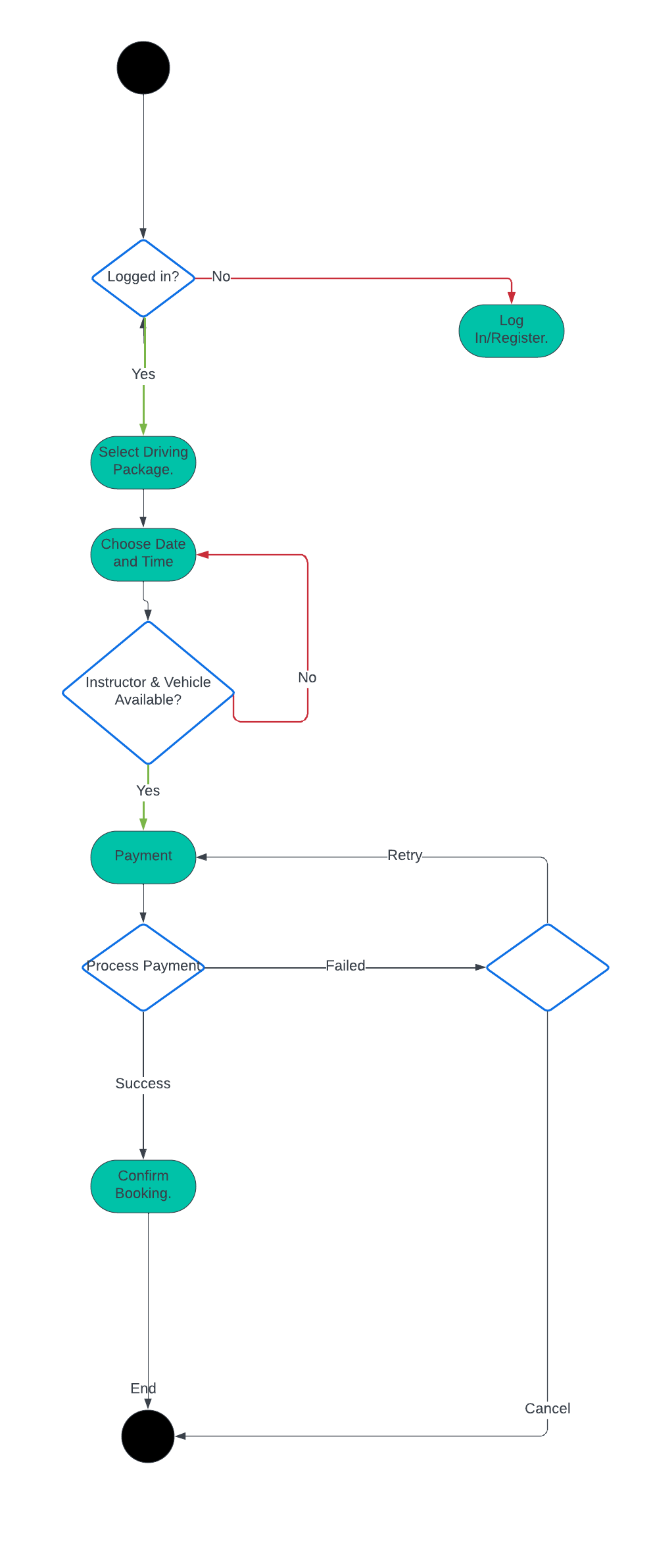
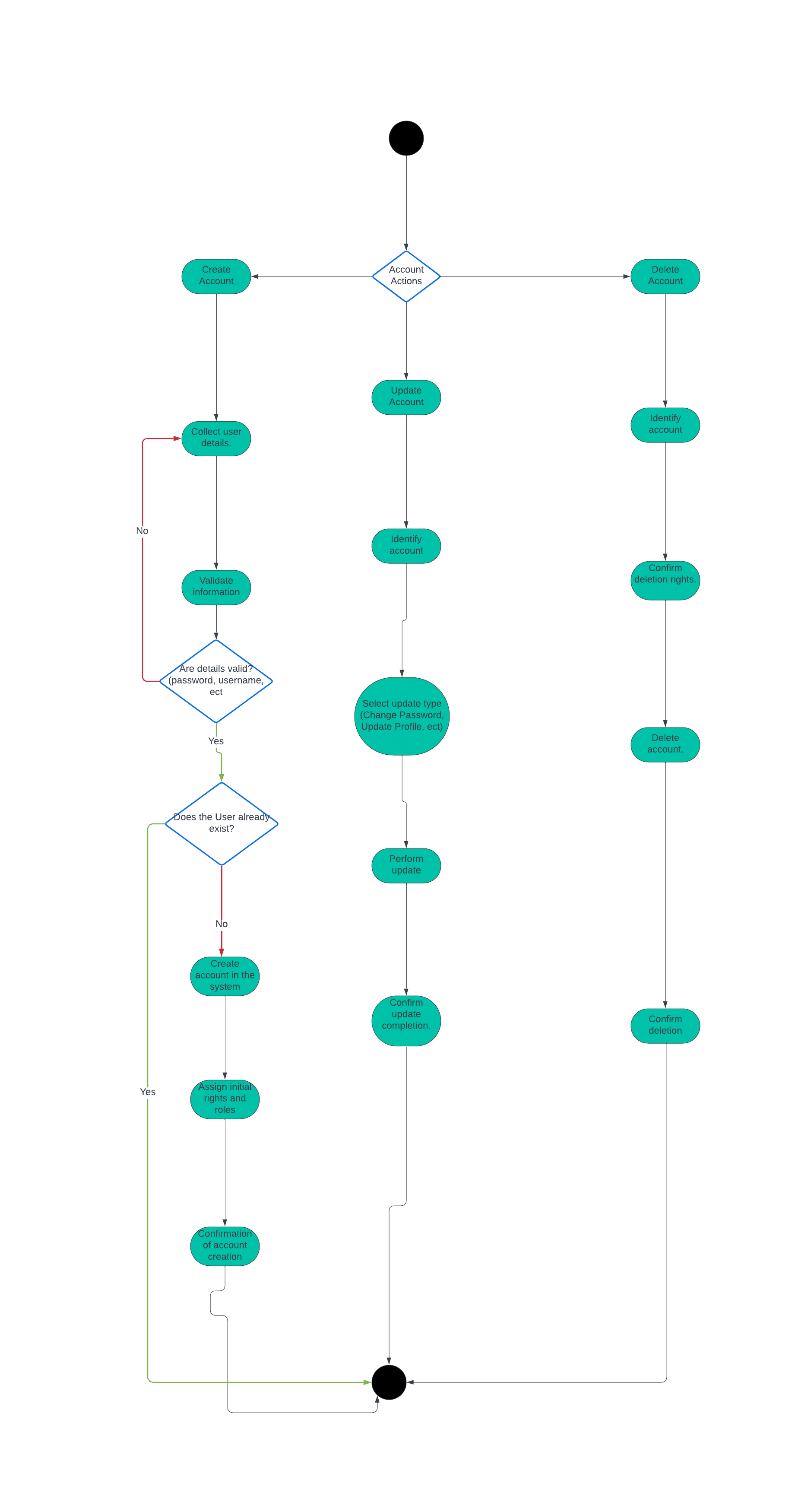
# CS 255 System Design Document

## UML Diagrams

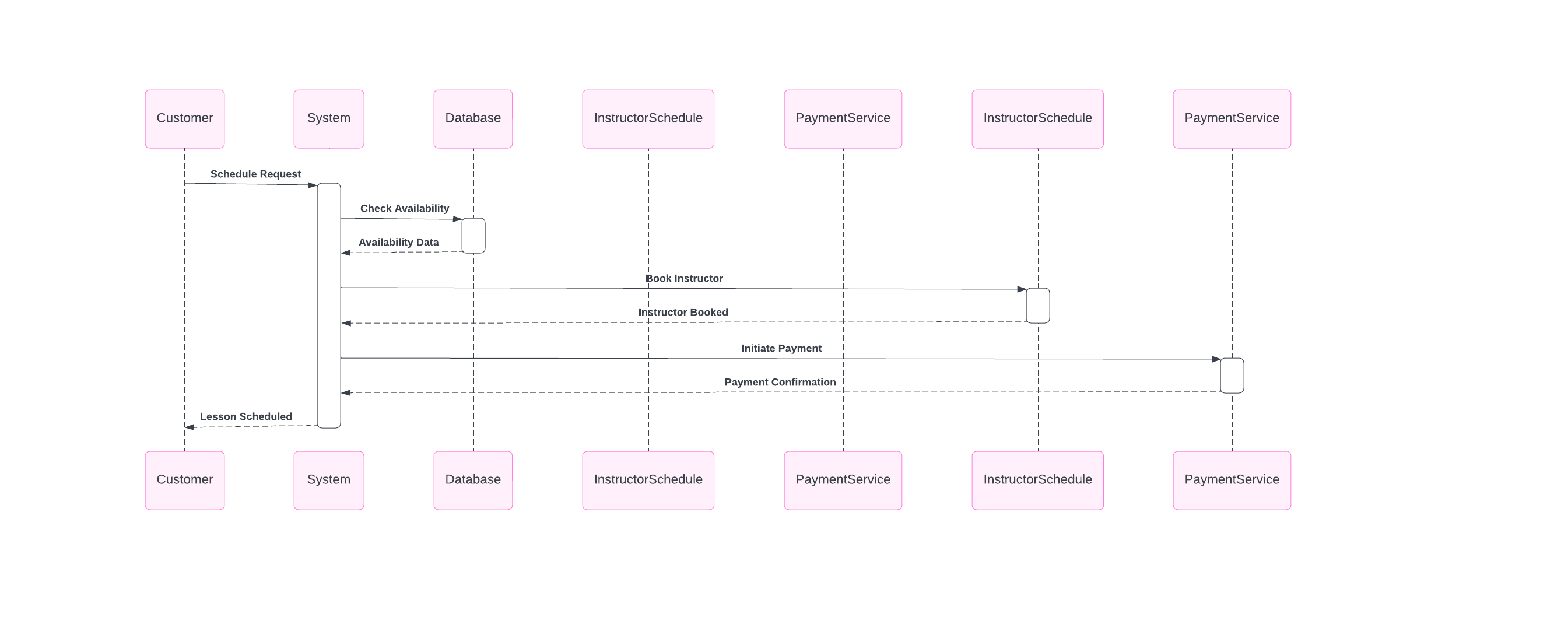
### UML Use Case Diagram



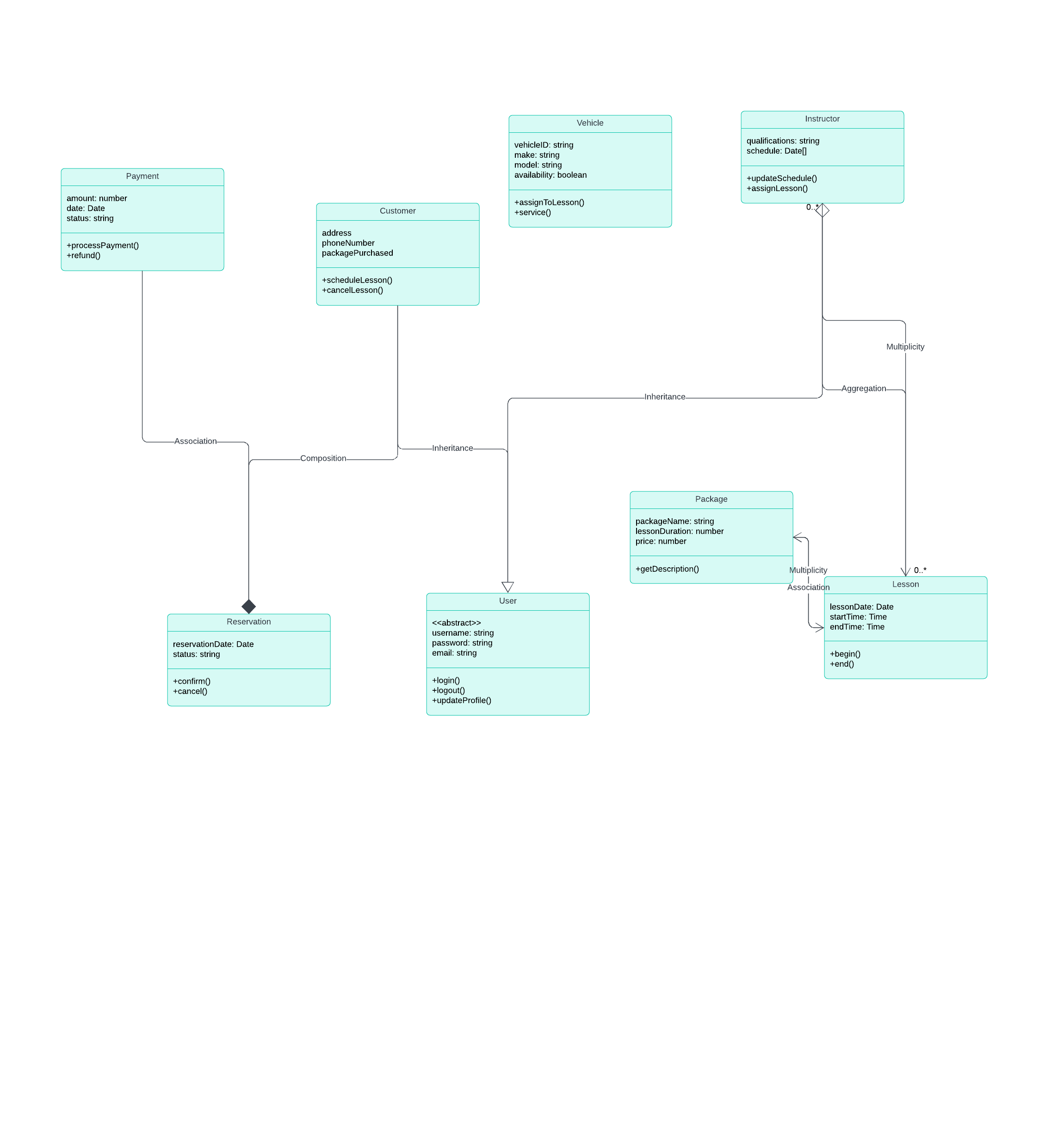
### UML Activity Diagrams Making a Reservation and Managing User Accounts

### UML Sequence Diagram

*Schedule Driving Lesson*  


### UML Class Diagram



## Technical Requirements

Hardware Requirements:

Server Specifications:

Type: Dedicated or cloud-based servers depending on deployment strategy

CPU: Multi-core processor for handling simultaneous requests

RAM: Sufficient memory for supporting concurrent users (specific size depending on expected load)

Storage: High-speed SSDs for database storage with enough space to handle user data and backups

Network: Gigabit Ethernet for internal connectivity, with high-speed external access

Client Devices:

Compatibility with modern web browsers on desktops and mobile devices

Mobile devices should have GPS capability for location services if the system will utilize real-time location tracking

Software Requirements:

Operating System:

Server: A stable server OS like Linux (Ubuntu, CentOS) or Windows Server

Client: Cross-platform support (Windows, macOS, Linux for desktops; iOS, Android for mobile devices)

Database Management System:

Relational Database: Use an RDBMS like PostgreSQL, MySQL, or a cloud-based solution like Amazon RDS if using cloud services

Ensure support for ACID properties to handle transactions securely

Backend Framework:

A robust backend framework like Node.js

Must support RESTful API development for mobile and web application communication

Frontend Development:

Use modern JavaScript frameworks like React.js or Angular for responsive design

HTML5 and CSS3 for markup and styling

Tools and Utilities:

Integrated Development Environment (IDE):

Visual Studio Code

Version Control:

Git with repository hosting on GitHub

Project Management and Collaboration:

Tools like JIRAor Asana for task management

Slack or Microsoft Teams for communication

Continuous Integration/Continuous Deployment (CI/CD):

CloudFormation (AWS), or GitHub Actions for automating the deployment pipeline

Cloud Services:

AWS, Google Cloud, or Azure for hosting, databases, storage, and other services if opting for cloud infrastructure

Content Delivery Network (CDN):

Use of a CDN like Cloudflare or AWS CloudFront for faster content delivery

Security:

SSL/TLS certificates for secure HTTP requests

Firewalls, anti-malware, and intrusion detection systems for server security

Backup and Disaster Recovery:

Regular automated backups

Strategies in place for disaster recovery and data redundancy

Compliance and Regulations:

Data Protection:

Compliance with GDPR, CCPA, or other relevant data protection regulations

Encryption of sensitive data both at rest and in transit

Payment Processing Compliance:

PCI DSS compliance if handling credit card transactions directly