# CS 255 System Design Document Template

This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead the goal is to complete each section based on what your client’s needs are. Remove this note when you are finished, and replace all bracketed text with the relevant information.

## UML Diagrams

### UML Use Case Diagram

*[A diagram of a driver pass system

Description automatically generated*

*]*

### UML Activity Diagrams

*[*

*A screenshot of a diagram

Description automatically generated*

*A diagram of a company

Description automatically generated with medium confidence]*

### UML Sequence Diagram

*[*

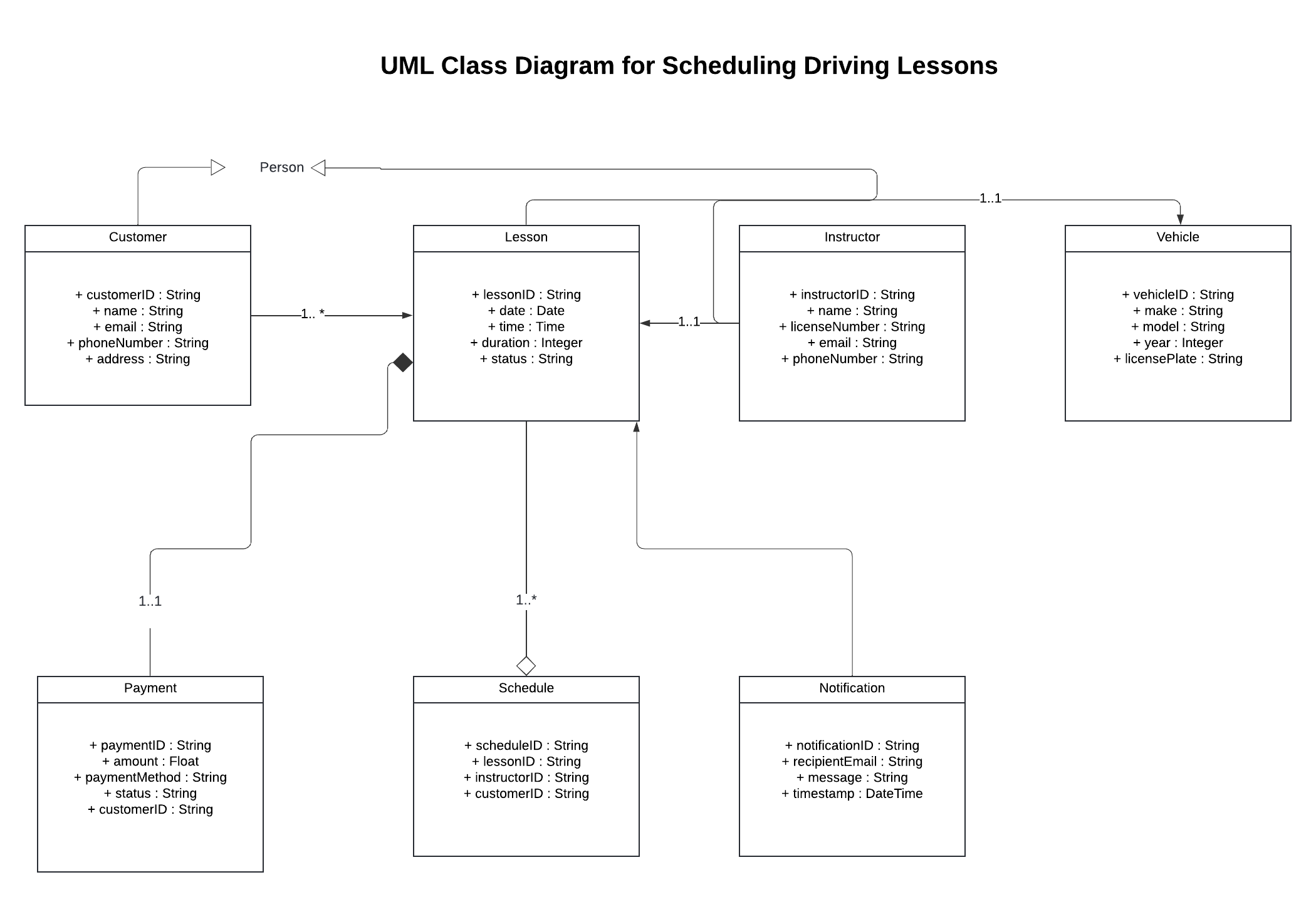
*A black background with white dots and lines

Description automatically generated*

*]*

### UML Class Diagram

*[*

**

*]*

## Technical Requirements

*[ To guarantee performance, scalability, and security, the driving lesson scheduling system needs hardware, software, tools, and infrastructure. A web server and a database server with multi-core CPUs, 16 GB of RAM, SSD storage, and fast network connections are needed for the system's hardware. Modern processors, 4 GB of RAM, and a steady internet connection are requirements for client devices, which might be desktops, laptops, or mobile phones.*

*The program consists of a frontend developed with React, Angular, or Vue.js, a backend framework like Python (Django/Flask) or Node.js, and a server-side operating system like Linux or Windows Server. Database administration will be handled by MySQL or PostgreSQL, and hosting, storage, and scaling will be provided by cloud platforms such as AWS, Azure, or Google Cloud. Client access will be supported by contemporary browsers like Chrome and Firefox.*

*Development will be aided by programs like Lucidchart for UML diagrams, Postman for API testing, GitHub for version management, and Visual Studio Code. Permissions will be managed by Role-Based Access Control (RBAC), and data will be secured by HTTPS, SSL/TLS, and encryption. Payments and notifications will be managed by APIs such as Stripe, SendGrid, and Twilio. Reliability and scalability will be guaranteed by load balancing, cloud backups, and monitoring tools, resulting in a safe and effective driving lesson scheduling system.]*