# 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 diagram

AI-generated content may be incorrect.UML Activity DiagramA diagram of a flowchart

AI-generated content may be incorrect.A group of squares with black dots

AI-generated content may be incorrect.

### UML Sequence Diagram

A diagram of a company

AI-generated content may be incorrect.

### UML Class Diagram

A diagram of a flowchart

AI-generated content may be incorrect.

## Technical Requirements

**System Architecture**

The DriverPass system will use a three-tier architecture: a web-based presentation layer, a REST API application layer, and a relational data layer. It will be deployed on a cloud platform using containerized services behind a load balancer for high availability and scalability.

**Infrastructure**

* **Compute & Hosting:** Managed Kubernetes or an autoscaling app platform with at least two production pods.
* **Storage:** A relational database (PostgreSQL/MySQL) for core data; cloud object storage for files; and Redis for caching.
* **Networking:** Private subnets, security groups, and load balancing to isolate internal traffic.
* **Environments:** Separate development, testing, and production setups with distinct credentials.

**Application Stack**

* **Backend:** RESTful API built with Java Spring Boot, .NET, or Node.js.
* **Frontend:** Responsive web app using React or Vue.
* **Background Jobs:** Queued workers for notifications, scoring, and reporting.
* **Config Management:** Centralized secrets and configuration service.

**Integrations**

* **Payment Processor:** Secure integration with Stripe, Square, or Shopify using tokenized payments.
* **Messaging:** SendGrid, SES, or Twilio for email/SMS notifications.
* **Optional Services:** Mapping API for location validation and DMV data import for up-to-date content.

**Security and Access**

* **Authentication:** Role-based access (Student, Instructor, Admin). Passwords stored using bcrypt or Argon2.
* **Encryption:** All data in transit via TLS 1.2+ and sensitive fields encrypted at rest.
* **Auditing:** Logins, payments, and system changes recorded.
* **Compliance:** Meets PCI SAQ A and privacy standards.

**Data and Backup**

Core tables include Student, Account, Instructor, Lesson, Vehicle, Reservation, Payment, and TestAttempt. Daily database backups support point-in-time recovery. Logs are retained 90-365 days, and financial data for seven years.

**Performance and Monitoring**

The system targets 99.9% uptime, API latency under 500 ms, and responsive page loads under three seconds. Centralized logging and alerts will track performance and error trends.

**Development and Quality Assurance**

* **Testing:** Unit and integration tests with at least 80% coverage.
* **CI/CD:** Automated builds, static scans, and blue-green deployments.
* **Environments:** Staging environment mirrors production for QA.

**Client and Administrative Tools**

The system supports current browsers on desktops and mobile devices. An admin dashboard will manage users, packages, lessons, and reports, while instructors and students access tailored portals for lessons and testing.