# CS 255 System Design Document

# Christopher Bryan

# August 17, 2025

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

*[In Module Six, you were asked to complete a use case diagram based on your system design. If you would like to make any adjustments to your diagram, please do so. Please insert your use case diagram 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 software system

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 process

AI-generated content may be incorrect.*

*A diagram of a software process

AI-generated content may be incorrect.*

### UML Sequence Diagram

*[You were asked to create a sequence diagram based on* ***one*** *of the use cases you chose. Please insert your sequence diagram 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 software application

AI-generated content may be incorrect.*

### UML Class Diagram

*[You were asked to create a class diagram based on the different classes and attributes needed for your system design. You are* ***not*** *required to include methods, but you may if you wish. Please insert your class diagram here. Check to make sure that you included appropriate components and symbols and that your design meets the client’s requirements.]*

*A diagram of a company

AI-generated content may be incorrect.*

## Technical Requirements

*[Based on the diagrams you have created, describe the technical requirements of your system. These requirements should address the required hardware, software, tools, and infrastructure necessary for your system design.]*

### Hardware / Hosting

* **Cloud-hosted** web application capable of scaling with application and database tiers.
* Redundant storage and **automated daily backups**.
* Bandwidth sized to support **at least 100 concurrent users** under normal load.

### Software Stack

* **Frontend:** HTML5 and JavaScript; capable and responsive UI for desktop/tablet/mobile.
* **Backend:** Server-side framework (e.g., Java, Python, or PHP) providing REST endpoints for auth, scheduling, payments, reports.
* **Database: Relational DB** with MySQL for users, reservations, vehicles, instructors, packages, payments, and test results.
* **Reporting/Export:** Server-side PDF/Excel generation for activity and performance reports.

### Tools

* **Modeling:** Lucidchart for UML (use case, activity, sequence, class).
* **CI/CD:** Automated build and deployment to staging and production.
* **Issue tracking:** Any standard tracker (Azure Boards).

### Infrastructure & Security

* **HTTPS (TLS)** for all web traffic with HSTS enabled.
* **Authentication** with credential verification; **account lockout** on brute-force attempts; secure password reset options (email).
* **RBAC** (role-based access control) for Student, Instructor, Secretary, Admin/IT.
* **Data protection:** Encrypt sensitive data at rest (DB/backup encryption).
* **Audit logging** for login attempts, reservation create/modify/cancel, password resets.

### Integrations

* **DMV content updates** (policies/test questions) via scheduled job or API pull with admin notification on a regular basis.
* Optional: Calendar integration for reminders (Google Calendar).

### Performance & Scalability

* **Page interactions with less than 2 seconds** under normal load; horizontal scaling for app nodes.
* Reservation workflow prevents double-booking by checks on driver and vehicle availability.

### Availability & Recovery

* **Uptime target:** 99.5%+ during business hours.
* **Backups:** Weekly full + Daily incremental.
* **Disaster recovery:** Cross-region backup storage; RTO/RPO defined by business.

### Monitoring & Logging

* Centralized logs for auth events, reservation state changes, and integration jobs.
* Application metrics, DB health, and alerting on error rates.

### Environments & Deployment

* **Environments:** Dev → Staging → Prod with data seeding for test accounts.
* **Maintenance windows:** Off-peak updates for system and DMV content.