# CS 255 System Design Document Mike Brown

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

AI-generated content may be incorrect.*

### UML Activity Diagrams

A diagram of a program

AI-generated content may be incorrect.A diagram of a program

AI-generated content may be incorrect.

### UML Sequence Diagram

*A diagram of a student login

AI-generated content may be incorrect.*

### UML Class Diagram

A diagram of a data flow

AI-generated content may be incorrect.

## Technical Requirements

DriverPass will be a browser-based application so students, instructors, and staff can sign in, book lessons, and take practice tests without installing anything special. The system needs to be dependable and straightforward to use so nontechnical people can get tasks done quickly (Digital.gov, n.d.).

**Hosting & Hardware**

The app can run on either a cloud server or a local machine, depending on the budget. A practical minimum is a quad-core CPU, 16GB of RAM, and ~1TB of SSD storage. That supports the web server, background jobs, and a relational SQL database for accounts, schedules, and payments.

**Software & Security**

We’ll build on a modern web framework (for example, an ASP.NET Core–style stack) and store data in an SQL database. All browser traffic must use HTTPS with TLS to protect user data in transit. Passwords will be hashed, and the system will enforce account lockouts on repeated failures to reduce account abuse (NIST, 2023).

**Development & Operations**

Development tools: Visual Studio or VS Code with GitHub for source control. Diagrams and documentation will come from visual charting software, and test changes in a staging environment before release. Use cloud hosting (Azure or AWS) for easy scaling, plus regular backups and basic monitoring to keep the service available.

***References***

*Abrahams, M. (2018). Hit the mark: Make complex ideas understandable. Stanford Graduate School of Business. https://www.gsb.stanford.edu/insights/hit-mark-make-complex-ideas-understandable*

*Ashman, M. (2018). Introduction to professional communications. BCcampus. https://pressbooks.bccampus.ca/professionalcomms/*

*Bohl, F. (2018). How to present design work to non-designers. UX Collective. https://uxdesign.cc/how-to-present-design-work-to-non-designers-60c39b1a2d3f*

*Digital.gov. (n.d.). Plain language guidelines. U.S. General Services Administration. https://digital.gov/guides/plain-language*

*National Institute of Standards and Technology. (2023). Guidelines for the selection, configuration, and use of Transport Layer Security (TLS) implementations (NIST Special Publication 800-52 Rev. 2). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-52r2*