# 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

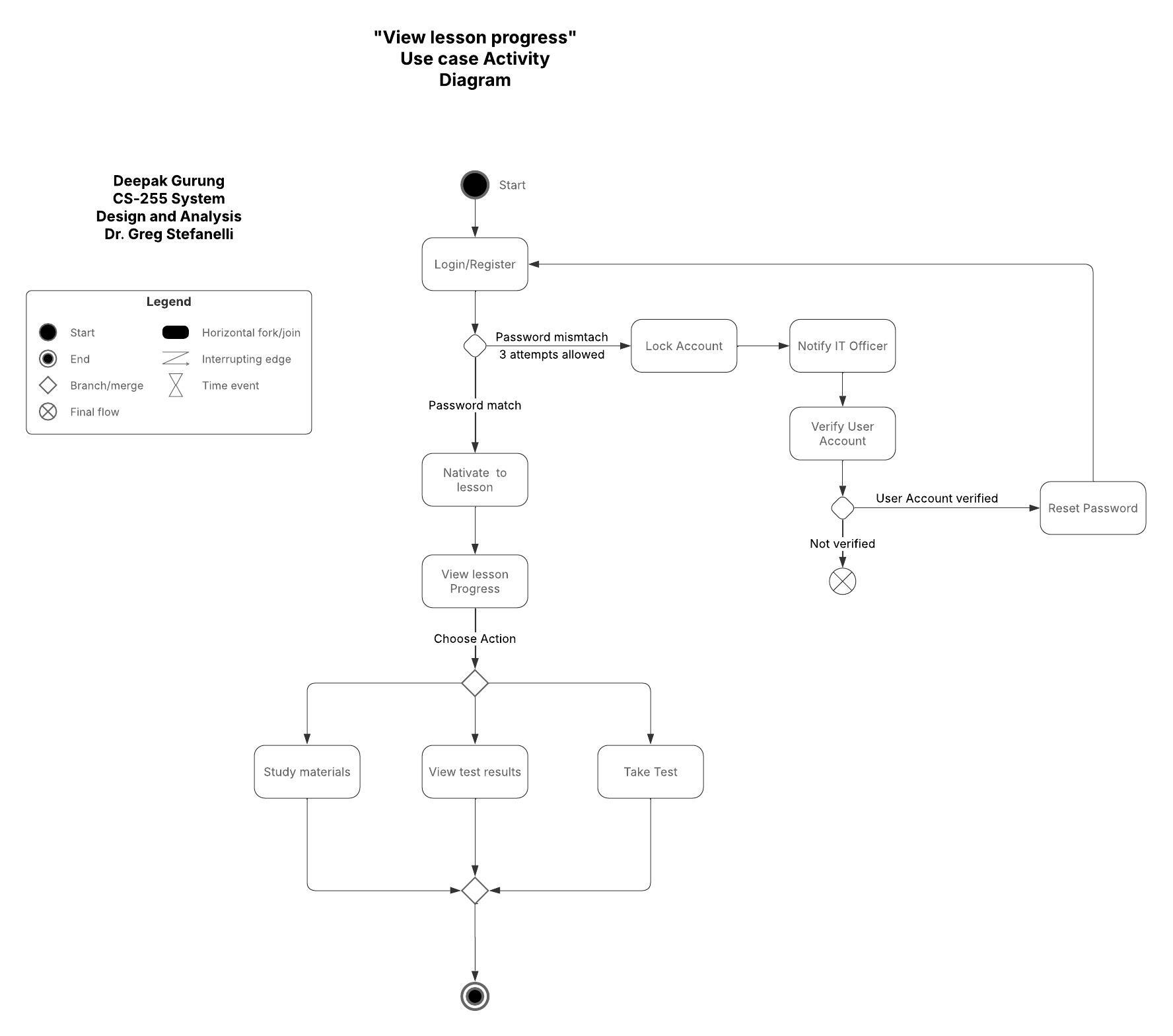
A diagram of a driver's process

AI-generated content may be incorrect.

### UML Activity Diagrams

A diagram of a process flow

AI-generated content may be incorrect.



### UML Sequence Diagram

A diagram of a diagram

AI-generated content may be incorrect.

### UML Class Diagram

A diagram of a company

AI-generated content may be incorrect.

## Technical Requirements

Based on the diagram I have created, the technical requirements, such as hardware, software, tools, and infrastructure are explored below:

* Hardware Requirements

1. Server: A Webserver is needed to host the application or software. A database server is also needed to store the information, such as bookings, payment methods, and customer profiles. And backup server to recover data if lost.
2. Client device: The user should be able to access the website from anywhere around the world via PCs, tablets, or smartphones.

* Software Requirements

1. Operating System: Any Windows server, such as Windows Server 2022, or Linux server, such as Ubuntu Server 22.04 LTS.
2. Web development: Node.js, Django (Python) for the backend (server side), and React or Angular for the frontend (client side).
3. Security: SSL/TLS for encrypting sensitive data, such as payment information and personal details. SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are protocols designed for safe and secure communications.

* Tools:

1. Developers: Developers will use IDEs (VS Code, PyCharm) or Visual Studio Code.
2. Testing ream will use PyTest (Python) for the backend. For the front end, we can use Jestand for angular, it comes with Angular Testing Framework.

* Infrastructure:

1. Cloud: The software should be hosted on cloud infrastructure, such as Azure, AWS, etc. It is important to host those platforms for scalability and availability. Scalability means, the system will be able to handle more loads and availability means, the system will run smoothly without much interruption, if hosted on the cloud.
2. Back and Recovery: Data is securely stored in the cloud and automatically backed up at scheduled intervals, which helps to remove manual backup.
3. APIs:
4. To process payments, Stripe/PayPal (payments).
5. The system will need a connection to the DMV through an API, which will automatically fetch the latest rules and updates from the DMV to keep the system up to date.

* User Requirements:

1. Users should be able to create accounts with **role-based access control.**
2. Admins can **add, remove, or modify user accounts** without backend access.
3. Admins receive **notifications for suspicious activity** (e.g., three failed login attempts, profile changes, system errors).
4. Customers can **schedule, cancel, or modify driving lessons** online.
5. The system should be able to offer **three training packages.**
6. The system should be able to **track lesson progress** (Not Taken, In Progress, Failed, Passed).
7. Drivers can **add comments** after each lesson.
8. The system **disables fully booked packages automatically**.