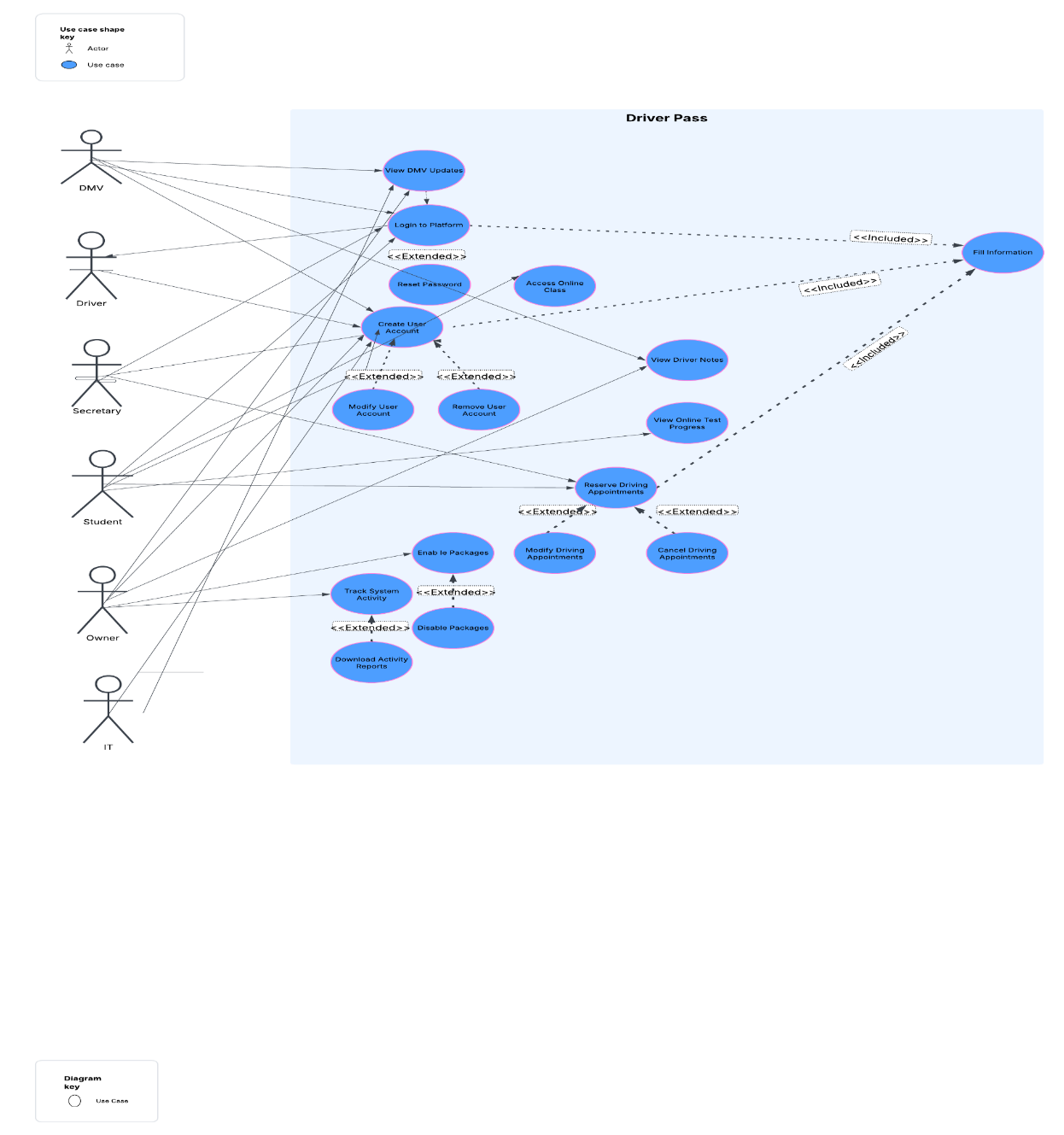
# CS 255 System Design Document Template

## UML Diagrams

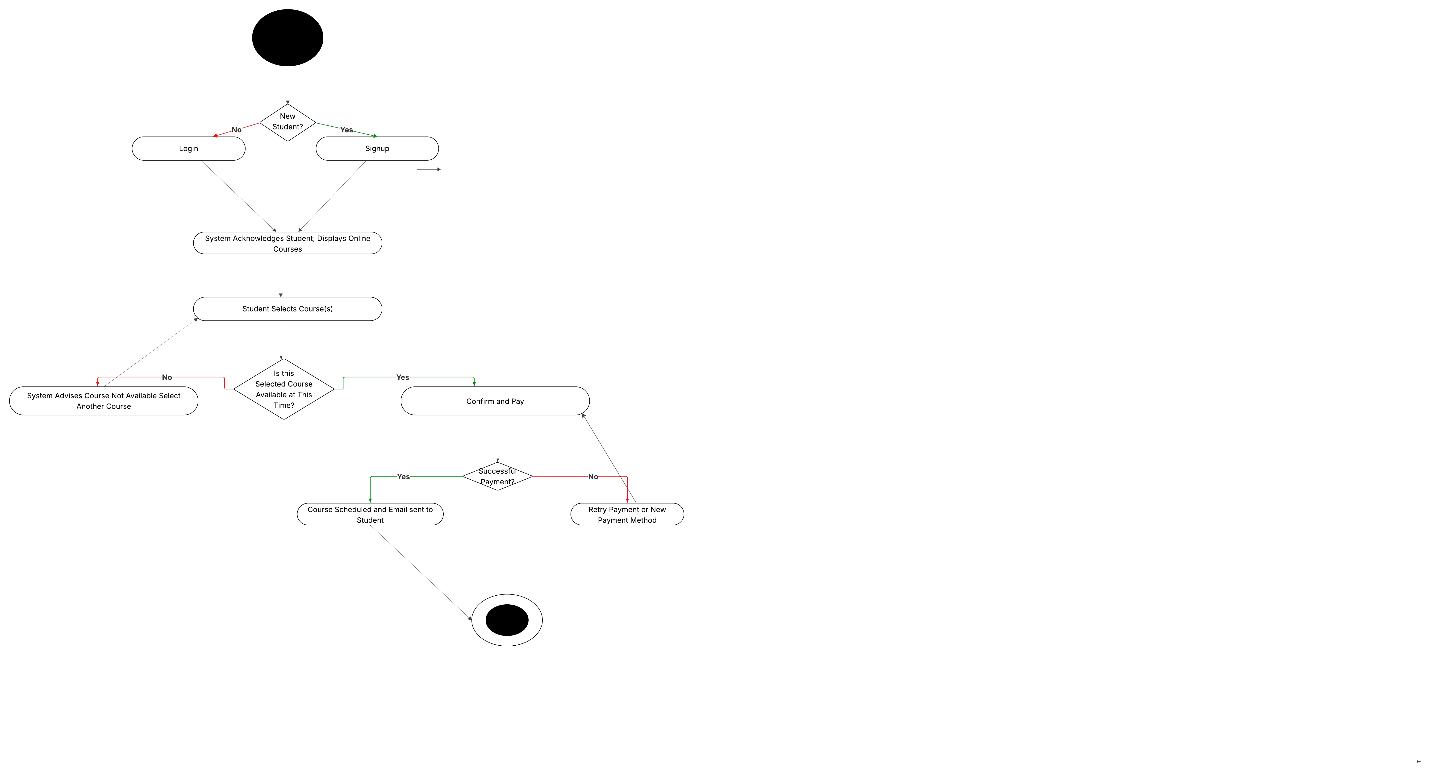
### UML Use Case Diagram



### UML Activity Diagrams

A black and white image of a black dot and a red dot

AI-generated content may be incorrect.

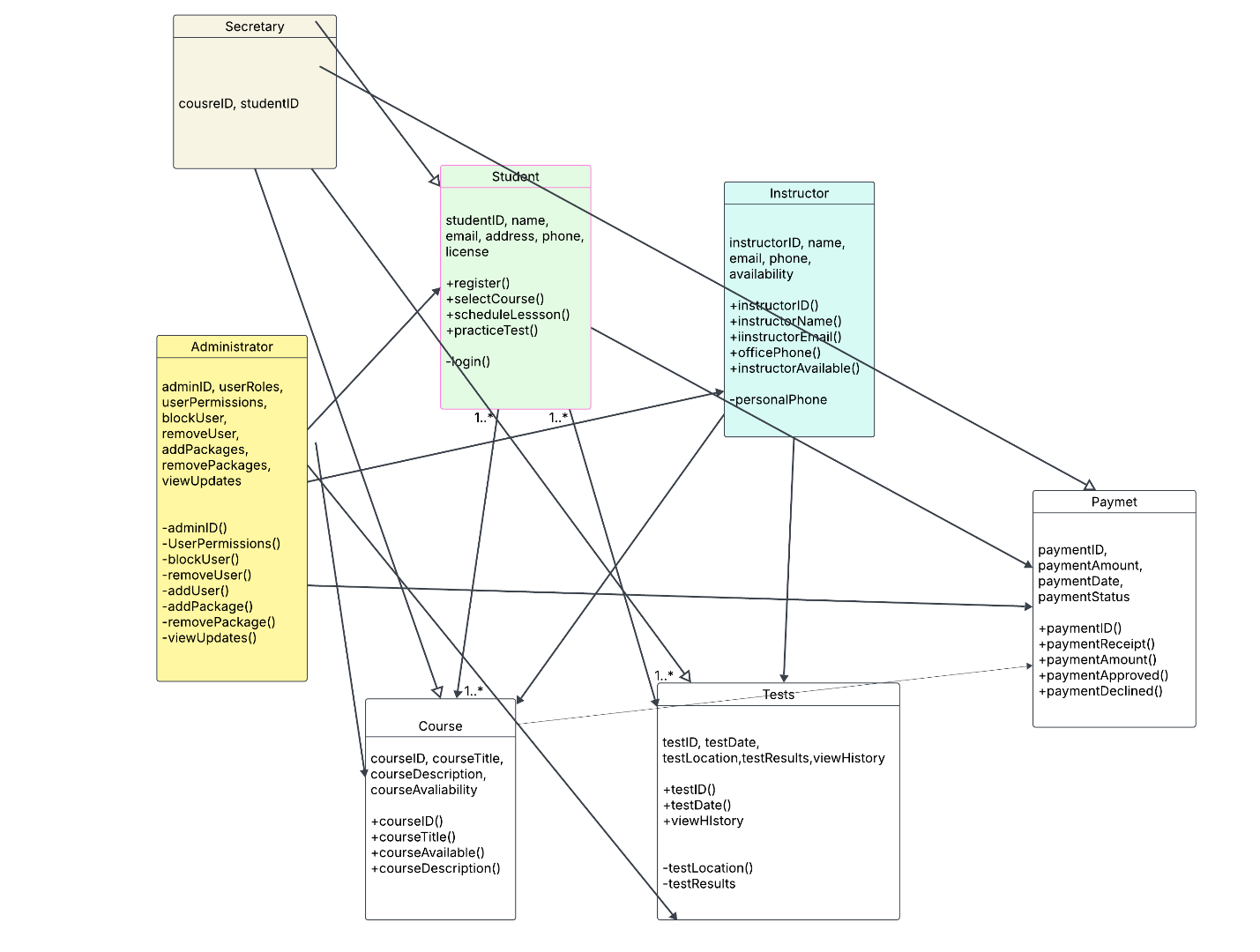


### UML Sequence Diagram

A diagram of a software application

AI-generated content may be incorrect.

### UML Class Diagram



## Technical Requirements

**Security** – Protecting personal data from intrusions

**Reliability** – The system needs to maintain a reliable connection in order for students to access their information and the system and not lose any progress.

**Scalability –** It is important that the system support an always increasing userbase and the ability to easily upgrade modules and components.

**Compatibility** – It is crucial for the application to work with different operating systems and platforms. These include but not limited to desktops, mobile phones, android OS, Apple IOS.

**Availability** – An application of this magnitude must be able to be accessed 99 to 100% of the time by users. Downtime should be minimized.

#### **Performance Requirements**

* Apple IOS, Android OS, Windows. Web Browsers such as Edge, Safari, Chrome.

**Speed -** Ideally, a page should load in under 2 seconds. Preferably 1 second. A response to users that a page is loading should appear within 100ms according to MDN Web Docs (n.d.).

**Updates –** It’s important for updates to the application be done regularly. The DMV interface needs to maintain updated information for the learning and testing materials to be accurate.

#### **Platform Constraints**

* The system should be able to run on Windows, macOS, Linux, and mobile devices such as Android and iOS. Since it will be a web-based application, the application will be built using a cloud-based server, which will support scalability. The backend will need APIs to connect to external interfaces such as the DMV; also, web-based databases, such as Supabase, should be considered.

#### **Accuracy and Precision**

* Users will be required to enter a unique email address, and the input will be case sensitive. Two-factor authentication is also a good way to provide a more secure user connection.

**Administrators will be notified of a problem:**

* If the 2FA fails several times
* Failure to connect to the DMV interface
* If the system fails or crashes
* If there is an attempted breach or actual breach

#### **Adaptability**

* Changes to the user can be made without changing the code. This code should already be implemented during the initial development. This can also be done by accessing the user database on the cloud server, and because this is a cloud-based application, cloud-based deployments can be utilized to assist with these smooth updates.
* Platform updates should be smooth and not require any excessive downtime.
* The IT admin will need top-level permissions such as admin cloud panel access.

#### **Security**

**User Login Requirements:**

User email and password

**Secure Connection:**

A secure connection between the client and server can be established by using HTTPS/TLS and JSON Web Token.

**Brute Force Hacking Attempt:**

If a brute-force hacking attempt occurs, the user account should be locked, and the

administrator should be notified.

**User Forgets Password:**

The user can either click on the “forgot password” button, which will notify the system, and

Administrator that the user needs to reset their password, or the user can fail to log in a certain

number of times, and the account will be locked until the user resets their password.

### Functional Requirements

* The system shall authenticate and validate user credentials
* The system shall allow users to sign up and enter their personal information
* The system shall allow users to select and sign up for different training packages
* The system shall allow users to cancel their training packages at any time
* The system shall allow users’ information to be entered by the secretary
* The system shall allow users to contact certain employees via online chat, email
* The system shall allow users to reset their passwords
* The system shall provide activity reports to track appointments
* The system shall track the progress of classes, tests, etc.
* The system shall notify the admin of DMV updates
* The system shall allow the admin to disable packages
* The system shall allow the admin to block those who are no longer employees.

### User Interface

* The interface needs to be responsive, easy to navigate, and work across different devices and platforms. Support secure user authentication and real-time updates and allow users to easily access the online learning and booking options.

**Types of Users:**

Students

Administrators

Instructors

Secretary

**Students:** I would expect most students to interact with the application via mobile, but I am sure some students will use laptops or desktops as well. Students will need to be able to sign up/login, schedule training lessons, access training and testing materials. Students should be able to contact faculty via chat or email.

**Administrators:** Administrators will need access via mobile and desktop. They will need to be able to disable packages, manage user access, receive update notifications from DMV, and block access if necessary.

**Instructors:**  Access should be on mobile and desktop. Instructors will need to check the scheduling, confirm scheduling, and be able to contact students. They should also be able to check and document student progress.

**Secretary: Access** will be mostly on a desktop/laptop. The secretary should have access to the scheduling of driving lessons and student progress.

### Assumptions

I am assuming that all students have internet access and access to an internet accessible device, and have the basic knowledge of how to navigate the application and web competently. There is no offline access; all tasks must be completed while connected to the internet.

### Limitations

* Cloud servers are a great way to provide security and storage, but the more you scale the application, the more expensive this can become.
* Accessing the application requires a stable internet connection
* Depending on the time frame specifications, it may not be possible to have all requested aspects of the application available at launch.

Reference:

MDN Web Docs. (2025, February 25). *Recommended web performance timings: How long is too long? Mozilla.* [https://developer.mozilla.org/en-US/docs/Web/Performance/Guides/How\_long\_is\_too\_long Timings: How long is too long? - Performance | MDN](https://developer.mozilla.org/en-US/docs/Web/Performance/Guides/How_long_is_too_long%20Timings:%20How%20long%20is%20too%20long?%20-%20Performance%20|%20MDN)