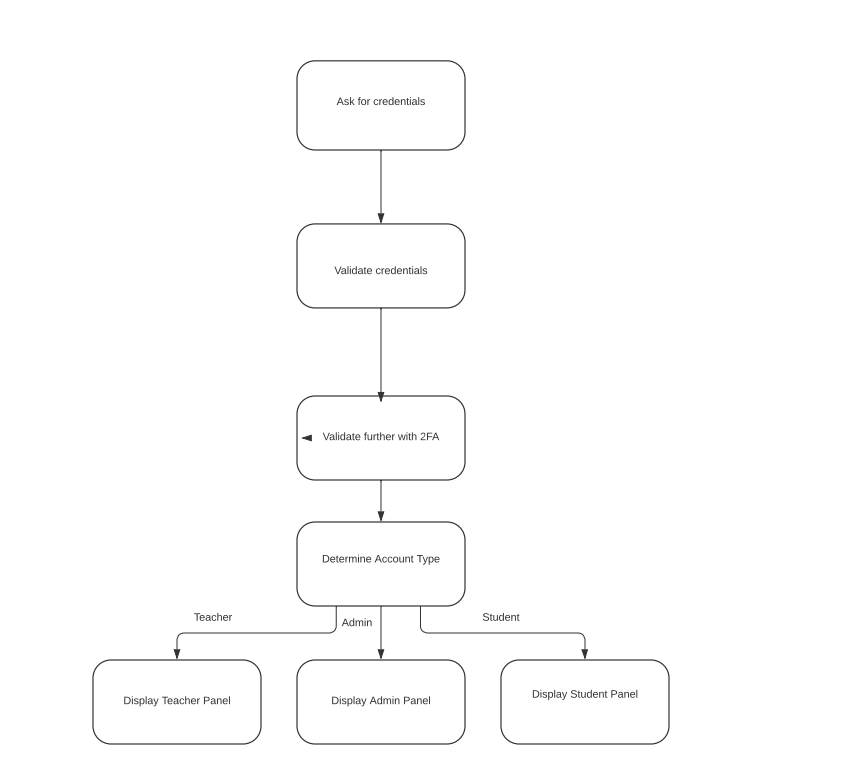
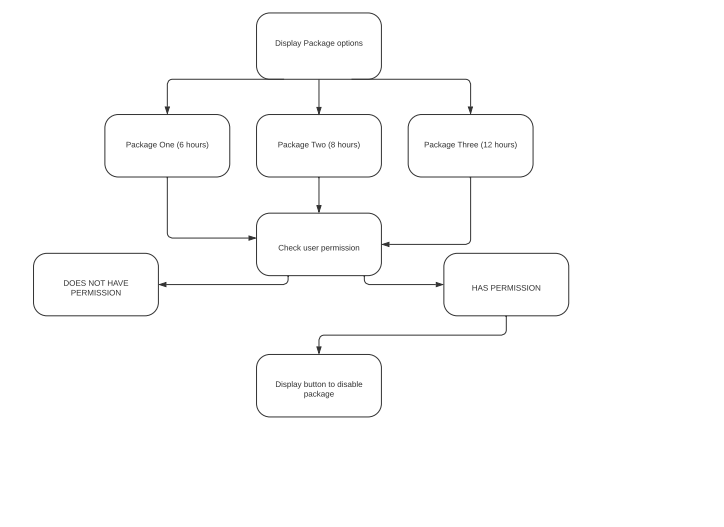
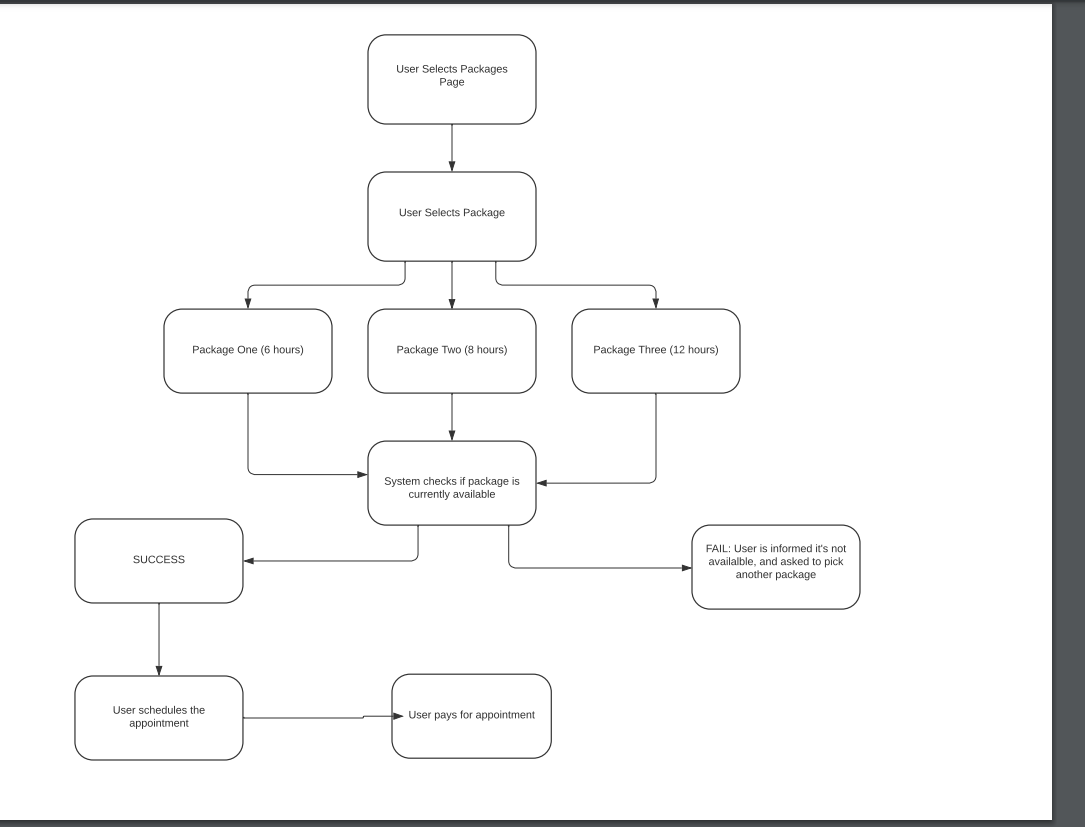
# CS 255 System Design Document Template

## UML Diagrams

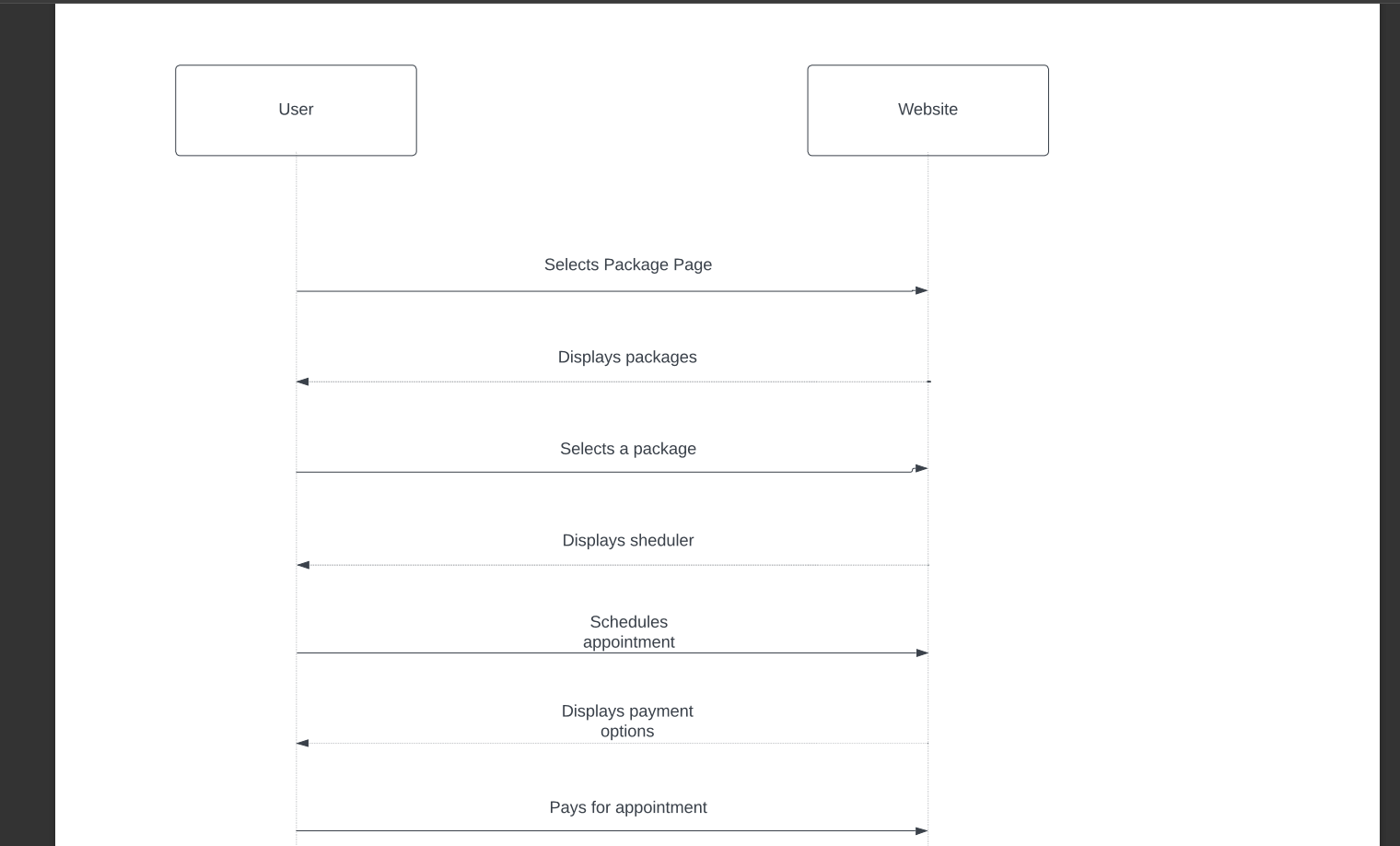
### UML Use Case Diagram

**

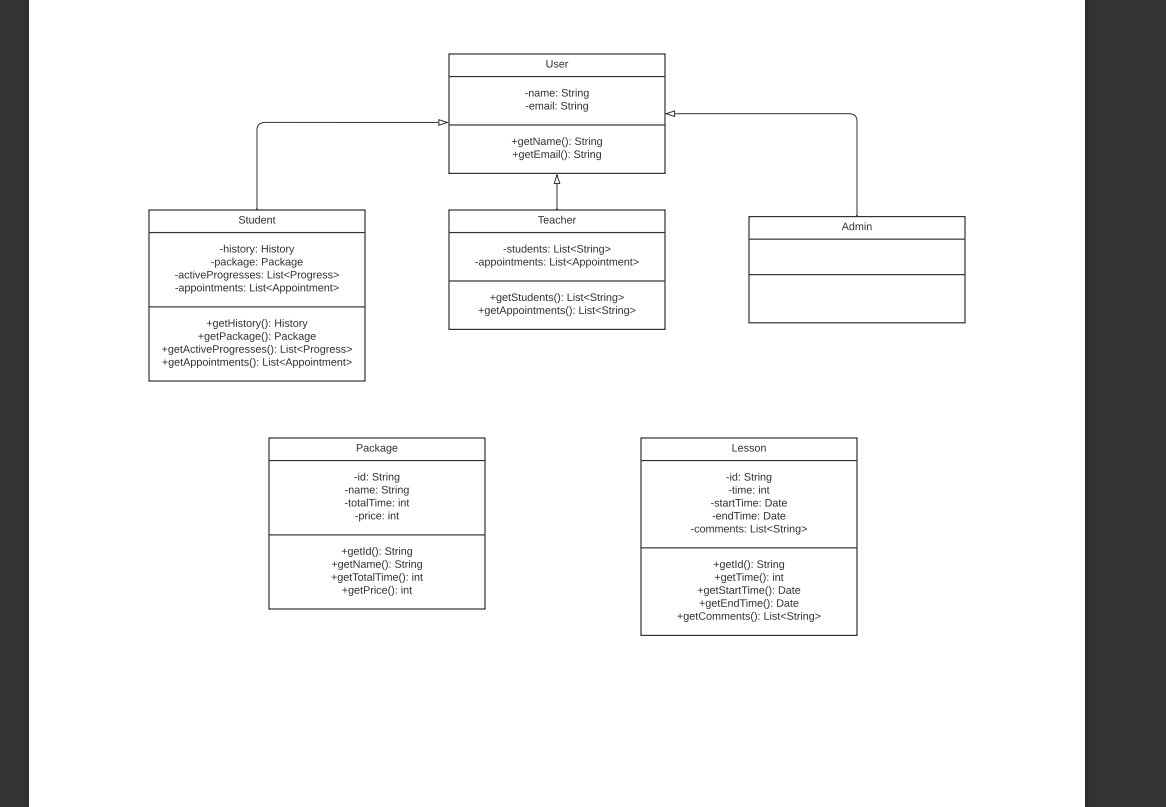
### UML Activity Diagrams



### UML Sequence Diagram

**

### UML Class Diagram

**

## *Technical Requirements*

Hardware:

* I think for hardware the best solution is to not worry about it and use the cloud. Having to deal with dedicated machines is bad for a few reasons. One, if it’s on site, that’s an additional IT person that would have to manage the physical machine. Two, even if it’s off site through a company like OVH, bare bone machines can’t handle scaling, and don’t have any bells and whistles, everything must be set up from scratch, and that requires a highly experienced system admin. The cloud offers scalability, and ease of use for the developers, and it’s cheaper.

Software & Tools:

* Software and tools basically go together. Speaking from a pure developmental standpoint, the software needed is strictly dependent on the tech stack being used. Here’s the tech stack I think this system needs. JavaScript, HTML, CSS, Firebase, and likely some sort of API handler for transactions, as doing that from scratch is too much work. There are likely some libraries I’m missing from this stack, but for the most part I think that’s really all that’s necessary here.

JavaScript handles the frontend/and backend by working with Firebase. HTML and CSS are pure front-end tools.