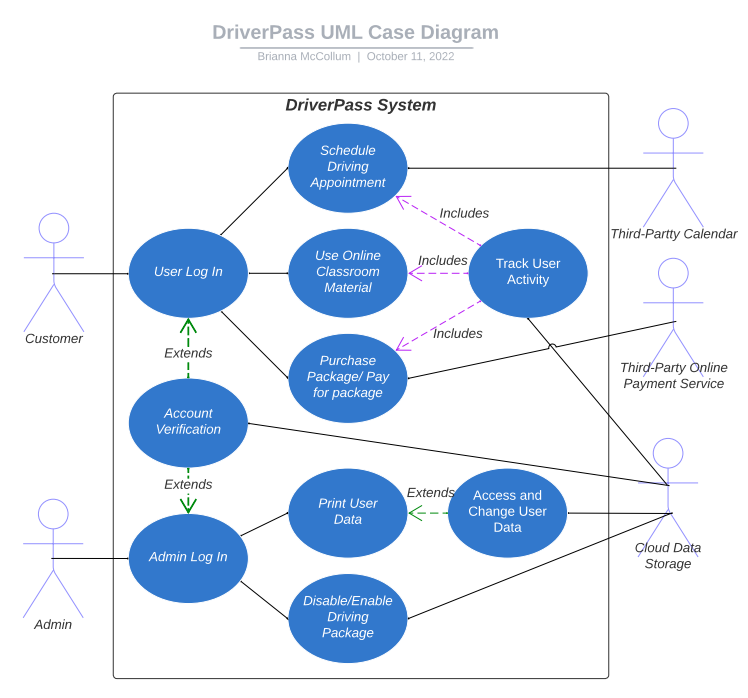
# CS 255 System Design Document Brianna McCollum

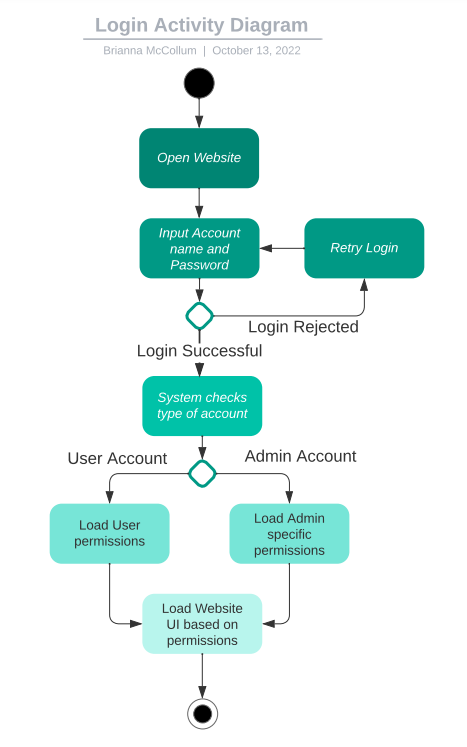
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length 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

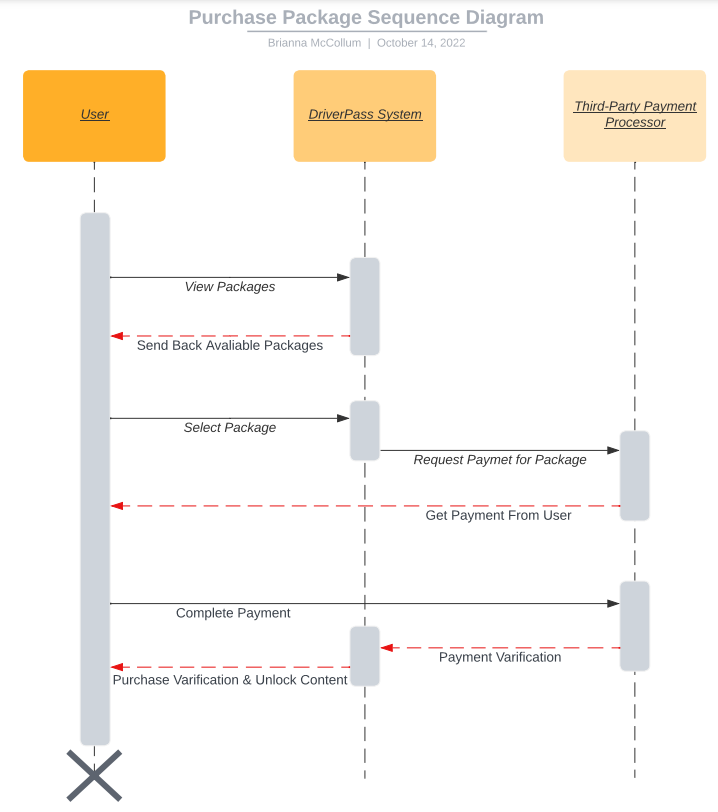
**

### UML Activity Diagrams

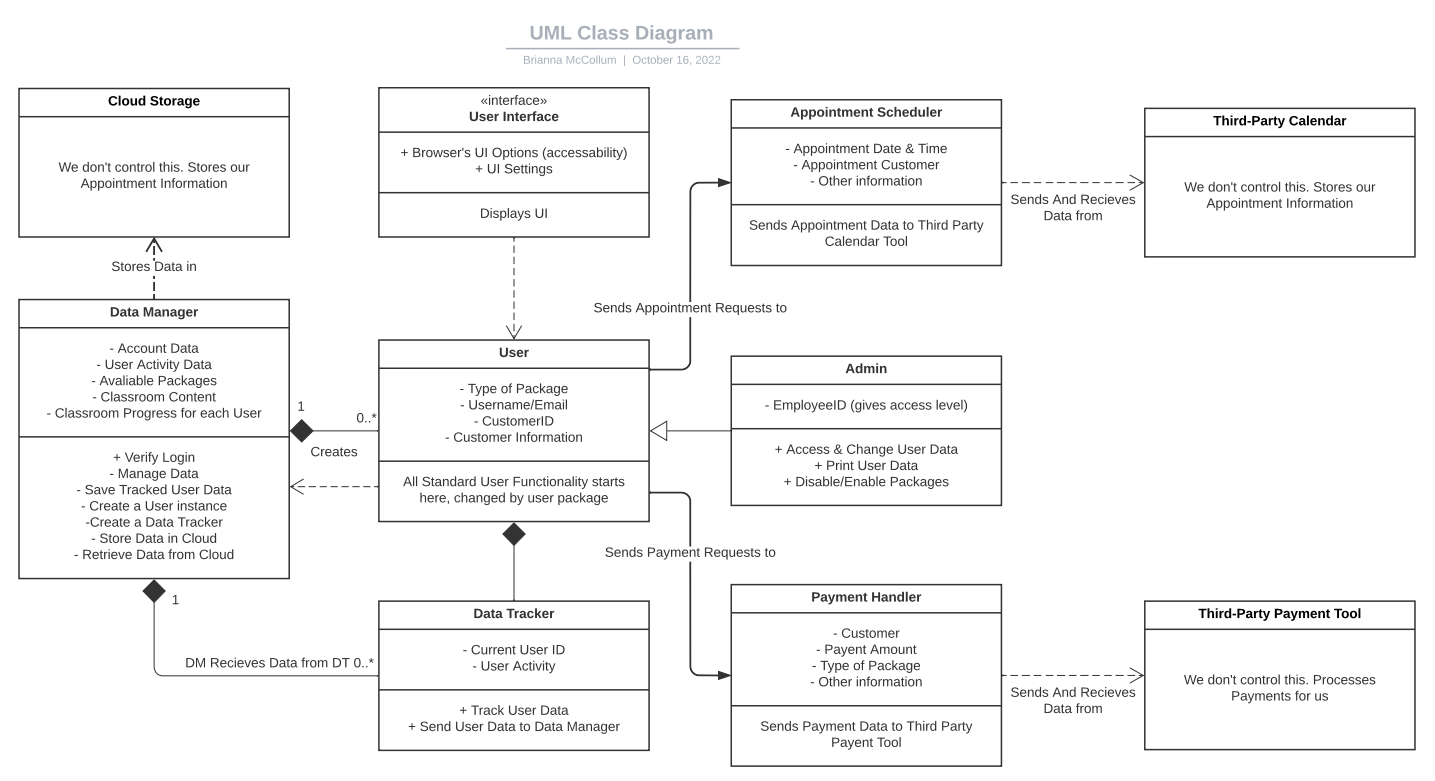
Diagram

Description automatically generated

### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

As far as hardware goes, the system we currently have laid out is relatively light on storage space. Instead of the servers needing to handle every single user’s information, non-active users have their info stored in the cloud. This means the server only needs to handle the active userbase, and considering most packages are just driving appointments this should ease the physical storage load on the system considerably. Probably the most intensive action the system will be taking is constantly tracking user activity, which means the servers will have to be prepared to handle 24/7 data recording of numerous users.

The third-party tools save a lot on custom software, serving as a simplified and more connected means to each end. The third-party cloud storage, calendar, and payment tool each have their own distinct uses that ease technical requirements, but it’s not entirely perfect. A lot of the system is out of our hands due to this reliance on third-party tools, meaning we must adjust the infrastructure around unforeseen and uncontrollable issues with the third-party tools. If the calendar or cloud goes down, and if the payment tool has security issues, etc., we must be ready to handle that.

Using Java as our main programming language means that the software will be incredibly flexible and powerful, but also will require outside compatibilities. Using website UI, encryption, and modernized internal objects to streamline functionality requires these dependencies, which is again not inherently a bad thing, but requires the system being prepared for compatibility issues and problems outside our control.