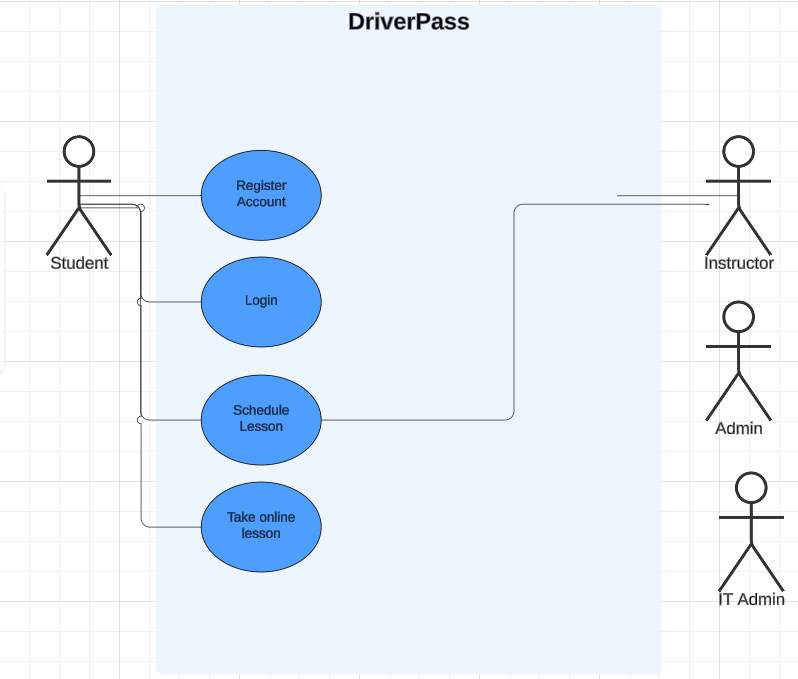
# CS 255 System Design Document Template

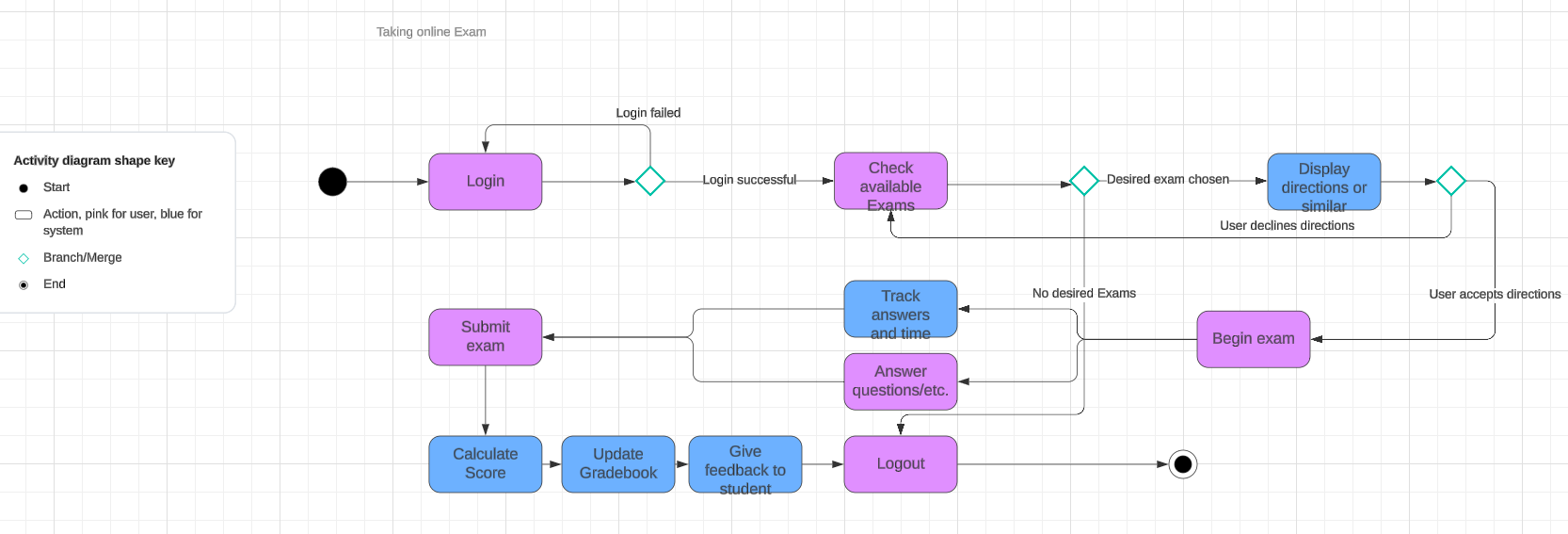
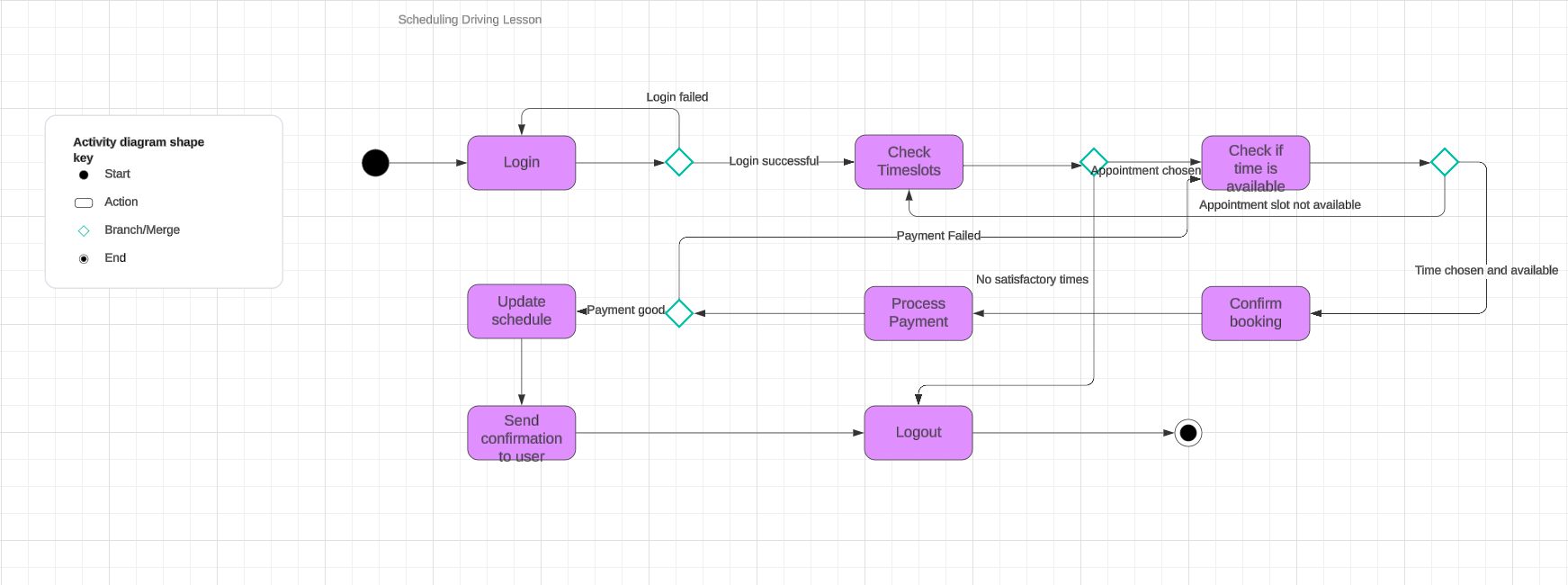
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length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

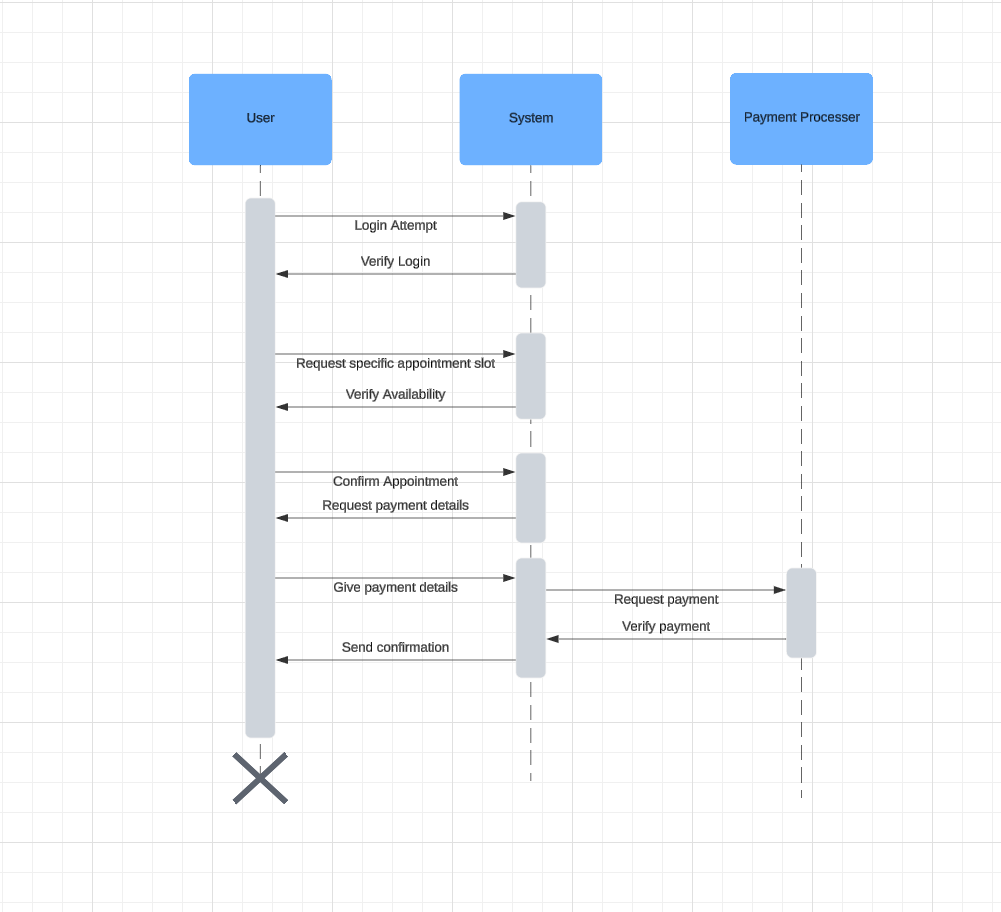
### UML Use Case Diagram

**

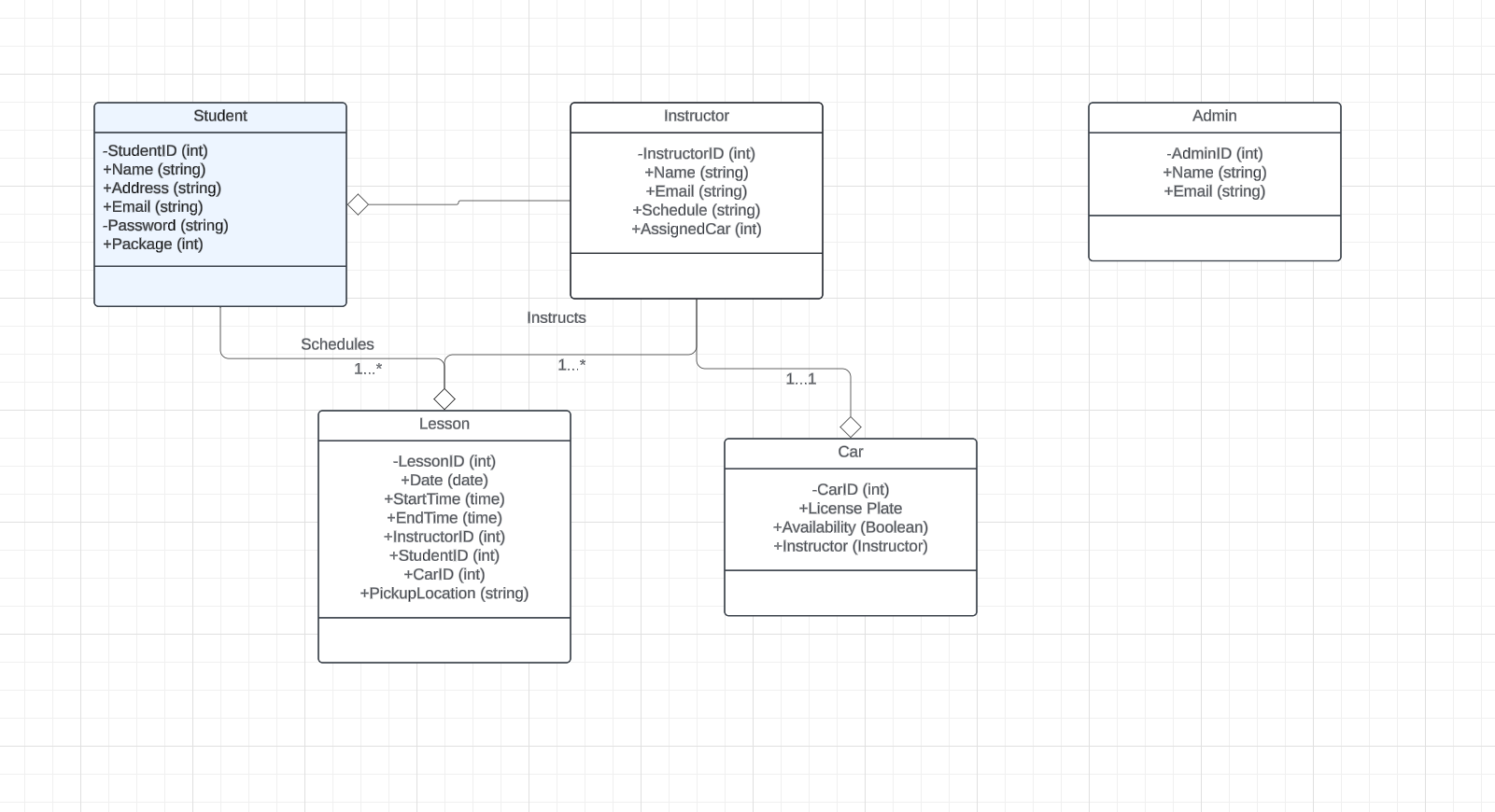
### UML Activity Diagrams

**

### UML Sequence Diagram

**

### UML Class Diagram

**

## Technical Requirements

For hardware requirements, the system requires servers to operate, cloud-based probably being the most optimal, that are able to handle multiple concurrent user logins of students, instructors, and administrators, able to store the users’ information, lesson schedules, and lesson progress, are able to host the students’ practice exams, and whatever training materials DriverPass implements.

For software requirements, the server should probably run on either Linux or Windows Server OS, depending on the staff’s preference. Mac Server OS may also be an option, but the fact that I do not know off the top of my head if that exists probably points to that being an inferior option. There should also be cross-platform support for Windows, Linux, macOS, iOS and Android, so users can access the system from both mobile and laptops or desktops. Some relational database for storing the previously mentioned data would also be required, and considering this system is unlikely to get to the extreme end of scalability, a cloud-based database would likely be sufficient again. Finally, a secure login mechanism is needed, probably including some kind of multi-factor authentication, and password hashing.

For infrastructure, a cloud-based solution is, again, probably most effective for actually hosting the system. Most cloud-based systems come with at least somewhat effective load-balancing for the system, and, although this isn’t an application that is likely to have an extreme amount of traffic, it’s better to have that load-balancing and not need it, than the other way around. Pretty much all cloud hosts are also going to have the monitoring and logging tools that administrators will pretty much require.

For nonfunctional requirements, the system should technically perform well enough that even at its peak times, the response time for any actions the user takes should be less than two seconds. The system should also be as reliable as possible, and most cloud systems will have at least the tools in place to allow some sort of failover, and for database reliability, depending on the total filesize literally just buying a second drive for, like, $50 on Amazon and replicating the database would probably suffice. For very large scale operations this wouldn’t really be optimal, even if all the data fit on a single drive, but I get the impression that the DriverPass office burning down would be so great a hindrance that their data being lost or damaged by the fire would be a negligible concern. But, as with the rest of these requirements, the cloud is also a very good solution, as cloud-based storage is fairly cheap and very reliable, so even that’s not that great a concern. The system should also be maintainable, so code should be kept modular and as simple as possible, and the system should have robust logs and performance metrics. The system should also be scalable; even if it’s not likely that DriverPass is going to go to Fortune500 scale, there isn’t much of a downside to it technically being possible.