# CS 255 System Design Document Template

DriverPass has asked for a new system design for their new web-based system to help them manage and deliver driving education to new students. The following are several diagrams that together help illustrate the system as a whole and demonstrate how this solution meets the client’s expectations.

## UML Diagrams

### UML Use Case Diagram

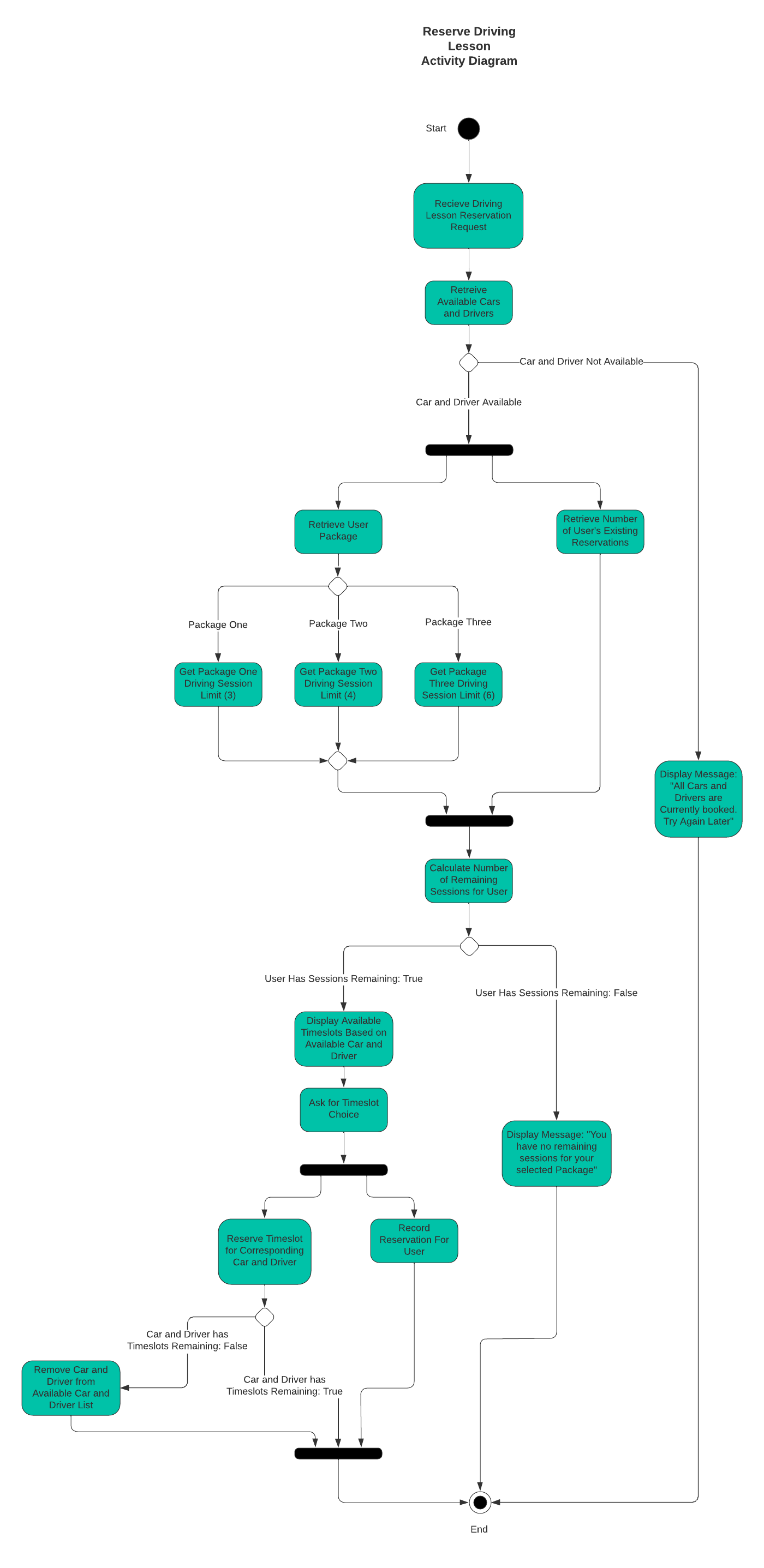
### A diagram of a diagram Description automatically generatedThis diagram illustrates the different roles that will be using the new system and what system functions are available to those roles.

### UML Activity Diagrams

Below are two diagrams depicting the flow and logic of when a user resets their password and when they reserve a driving lesson. These diagrams make it easier to follow the steps that need to happen to fulfill the user’s request. In addition, they also help to visualize the decision branches that must take place.

*A diagram of a program

Description automatically generated*

**

### UML Sequence Diagram

The Following Sequence Diagram shows the order of interactions between the user, the application client, the application server, and the application database when the user is resetting their password. This diagram helps to illustrate the sequence of events and how much involvement each application component is required.

A screenshot of a computer

Description automatically generated

### UML Class Diagram

A screenshot of a computer screen

Description automatically generated

## Technical Requirements

The system will have several technical requirements to support the functionality outlined in the above diagrams. To be able to function and take any actions, the system will need hardware to house the system and carry out certain tasks. The system will also need certain software to tell the hardware what to do and support the development process. There may also be some technical requirements that do not fall under either hardware or software. The following is a breakdown of the technical requirements required for this system.

Hardware:

* Cloud provider – to make supporting the infrastructure easier on the client, we will need a cloud provider to handle the creation and maintenance of resources we will need for our web-application.
* Virtual Machine (VM) – Provided by the Cloud provider, used to house the components our application needs.
  + We will need one or more VMs to host our web server that will service our clients' requests.
  + We will need one or more VMs to house our Database(s).

Software:

* Database – Used to store different kinds of information the system needs, such as customer data. We may have several if we want to insulate the different kinds of data for security purposes.
* Front-end framework – Provides consistency and best practices to develop the front end as well as tools to help development.
  + React, Angular, Bootstrap, etc.
* Software Languages for the Frontend
  + JavaScript
  + HTML
  + CSS
* Software Languages for the server
  + Partially based on developers’ expertise
  + Could be Java, C++, C#, Python etc.
* Operating system for the application server
  + Windows might be a good choice since the client does not have many IT resources.
* Firewall
  + Could be a hardware requirement, but not as likely since the infrastructure is cloud-based.
  + Could use a firewall native to the cloud being utilized.
* Load balancer
  + Could be a hardware requirement, but not as likely since the infrastructure is cloud-based.

Other:

* Authorization to interface with DMV systems – So that we can keep rules and policies up to date for online tests and classes.