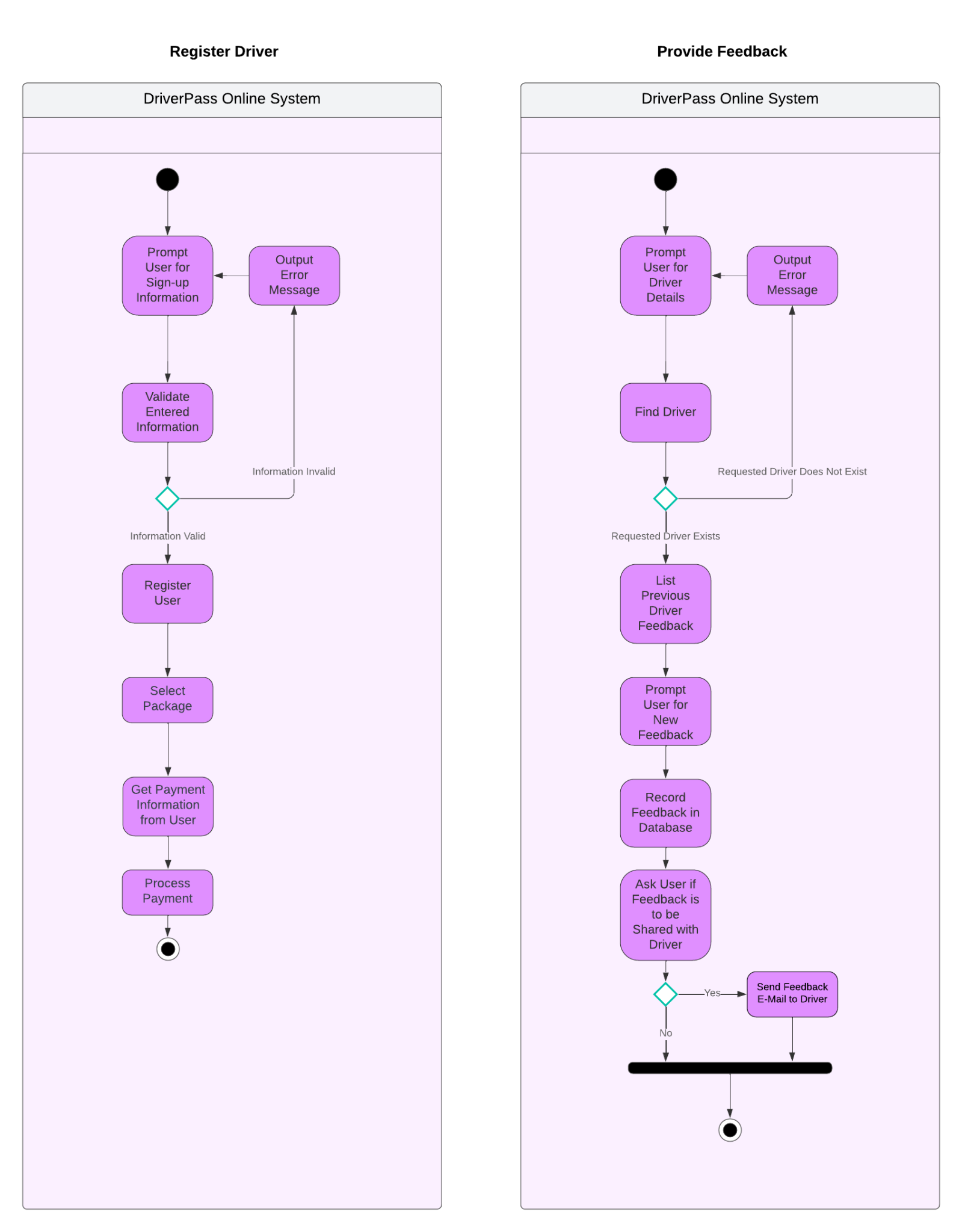
# CS 255 System Design Document

**Zachariah Spencer**

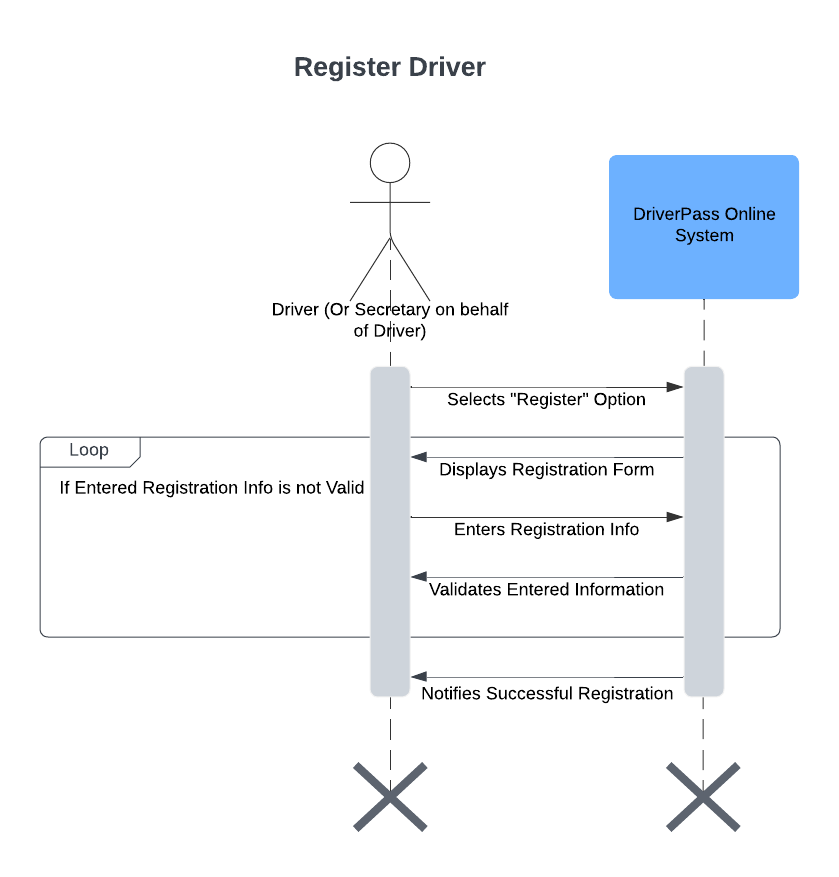
## UML Diagrams

### UML Use Case Diagram

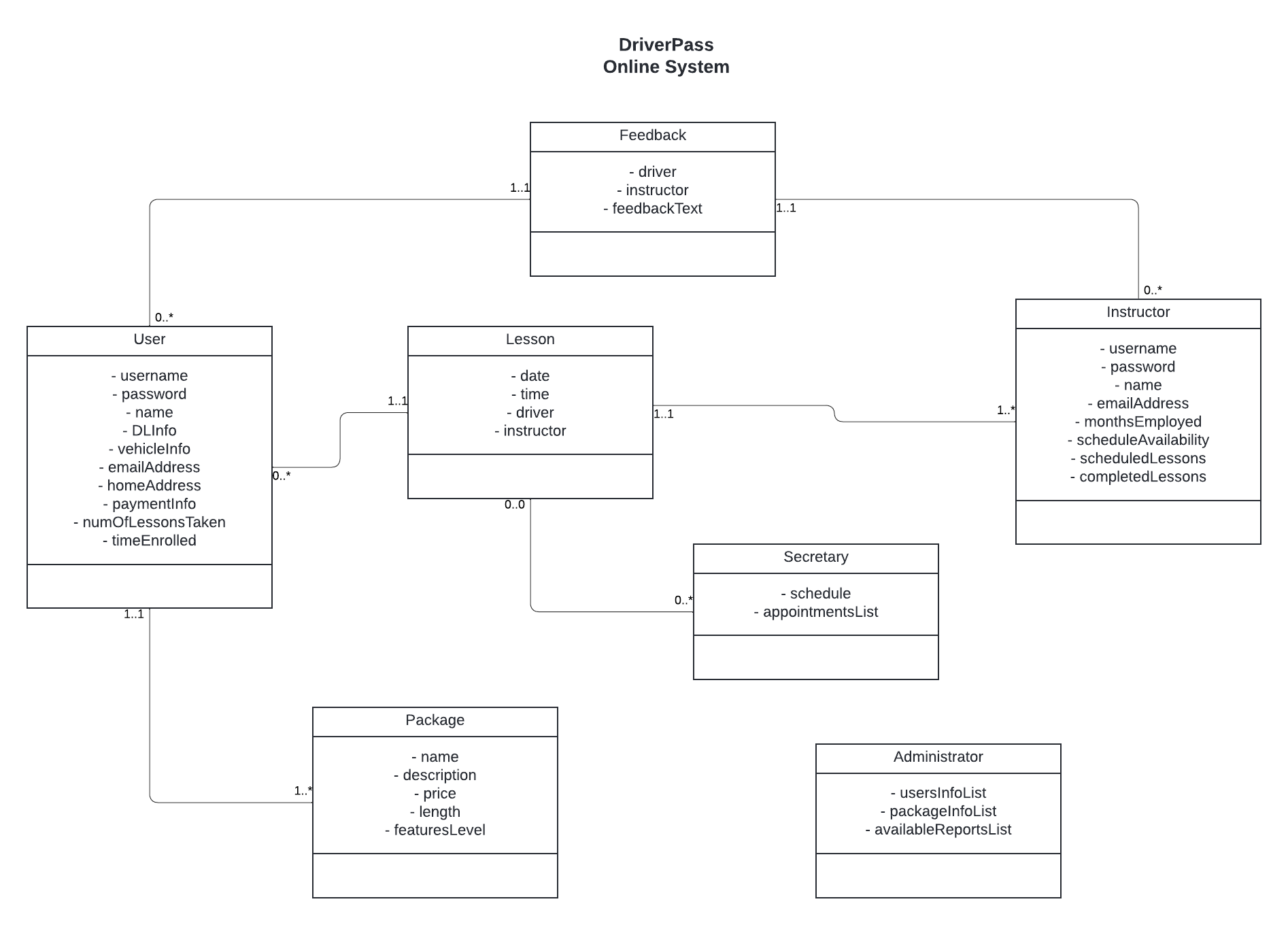
### UML Activity Diagrams

**

### UML Sequence Diagram

**

### UML Class Diagram

**

## Technical Requirements

* The system must must use cloud-based infrastructure to manage persistent data.
* The system must be run via a server off of a Linux system.
* The system must present user interface elements via HTML, CSS, and JavaScript.
* The system must use a NoSQL database to organize persistent data.
* The system must use a single-page web application architecture such as Angular to ensure responsive UI elements and seamless user access.
* The system must be developed utilizing Object Oriented Programming methodologies to ensure it is both scalable and relatively easy to modify and update.
* The system must make use of an OAuth API for user authentication and robust security processes.
* The system must include a variety of administrative tools that can only be seen and accessed when logged in with credentials that have administrator-level permissions. These tools must include a means to notify administrators of system issues or errors.
* The system must integrate a payment system API such as Square or Stripe to handle payment processing of packages.
* The system must either implement a scheduling system manually or utilize a scheduling/availability API such as Timekit.io or OnSched.