# DriverPass System Design

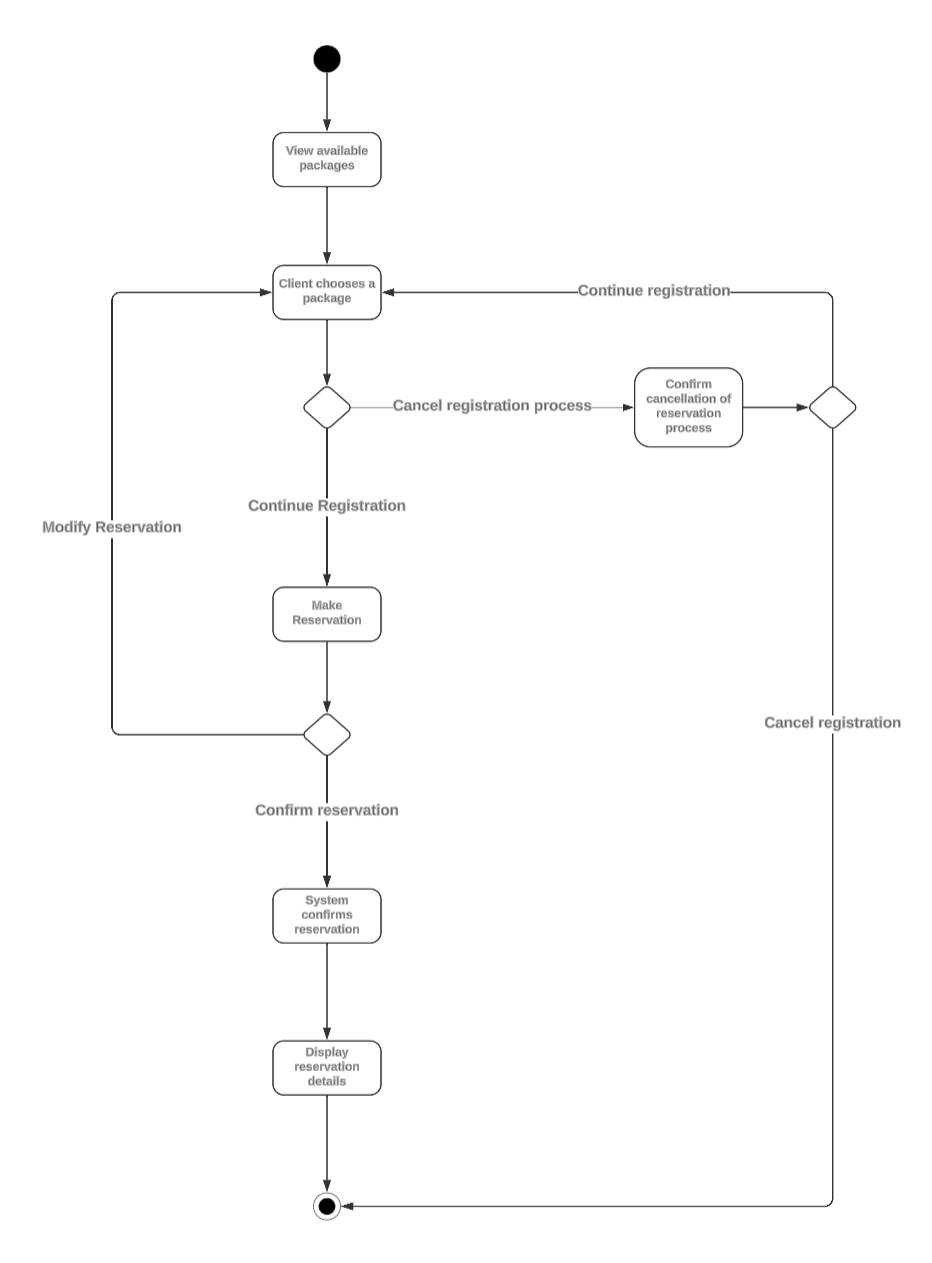
## UML Diagrams

### *A diagram of a diagram Description automatically generated*UML Use Case Diagram

### UML Activity Diagrams

A diagram of a computer program

Description automatically generated



### UML Sequence Diagram

*A screenshot of a computer screen

Description automatically generated*

### UML Class Diagram

*A diagram of a computer flow

Description automatically generated with medium confidence*

## Technical Requirements

**Hardware Requirements:**

**Server:** A high-performance server with a modern multi-core processor, at least 16GB of RAM, and sufficient storage for the database and application files.

**Client Devices:** The system should be accessible from various client devices such as desktop computers, laptops, tablets, and smartphones. This means the system should be designed to be responsive and work well on different screen sizes and operating systems.

**Software Requirements:**

**Operating System:** The server could run on a Unix-based operating system which are widely used for web servers due to their stability and security features.

**Backend Framework:** A server-side framework to handle requests and manage the application logic will be needed. This could be something like Django or Flask for Python or Express.js for Node.js

**Frontend Framework:** For creating the user interface, a frontend framework or library like React, Angular, or Vue.js could be used. Of these React seems to be most widely used.

**Tools:**

**Integrated Development Environment (IDE):** Tools like Visual Studio Code, PyCharm, or IntelliJ IDEA can be used in the development process.

**Version Control System:** Git, in conjunction with GitHub, can be used for version control.

**Project Management Tools:** Jira could be used to manage tasks, track progress, and collaborate more effectively.

**Infrastructure:**

**Cloud-Based Hosting:** Given that the system is expected to be cloud-based, then we should consider cloud service providers like Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure for hosting and scaling operations as needed.

**Security:** Infrastructure should be put in place to ensure the security of the system. This should include setting upfirewalls, using HTTPS with a proper SSL certificate for secure communication, regularly updating and patching the system, and following best practices for secure coding. The cloud service provider typically provides several security features and services, but we should also have additional measures within our system.

**APIs:**

**DMV API:** The system needs to have access to APIs for external systems like the DMV for updates on rules and policies.