# DriverPass System Design

## UML Diagrams

### UML Use Case Diagram

A diagram of a system

Description automatically generated with low confidence

### UML Activity Diagrams

A picture containing text, diagram, sketch, drawing

Description automatically generated

### A picture containing text, diagram, sketch, drawing Description automatically generated

### UML Sequence Diagram

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

### UML Class Diagram

A picture containing text, diagram, plan, technical drawing

Description automatically generated

## Technical Requirements

**Hardware:**

Servers: This system is web-based and requires servers for users to connect to. A cloud-based server is recommended for ease of use and scaling.

User device: A user device will be necessary to connect to the system. Any device with a web browser and internet connectivity will suffice as there’s currently no plans for a mobile application.

**Software:**

Operating System: An operating system that is compatible with the server is required. A platform like Azure is a great choice as it includes all the baseline needs for the application.

Database Management: Since we are adding, editing, and deleting client information for the database, we need a program to manage the database. MySQL or MongoDB are great options for SQL and non-SQL database respectively. Cloud services will usually offer something for this as well. For example, Azure offers Azure SQL Database among many others to choose from.

Server Software: The web-based interface will also need software to handle HTTP requests and responses. Apache, Tomcat and many other software can be used for this.

Backend Framework: Any server-side programming language can be used. Web-based systems have been trending toward JavaScript, Python or Java as of recently, but it is not a requirement to choose one of those.

Front End: This can be custom built with technologies like HTML and CSS or React, Angular or Vue.js. This decision will become clearer as the backend language is chosen.

**Tools:**

IDE:An IDE is necessary for writing and debugging the code.I prefer Visual Studio as it has the capabilities to handle most languages.

Project Management: A tool for managing tasks, having meetings, discussions, etc. is essential for communication and keeping the team on task. I prefer Discord for this as it’s very customizable, allows screen sharing, video chats, pictures, and can create different message boards for different topics. This could also be done by using a collection of programs like Google calendar, Trello and Zoom.

Version Control: To have a team work on a project simultaneously as well as manage the version releases and patch notes, something like GitHub is required.

**Infrastructure:**

Cloud: A cloud-based infrastructure like Amazon Web Services, Google Cloud or Microsoft Azure offers the simplest, all-inclusive solution for hosting the application and offering other the services touched on above to save development time and money as well as make the system more manageable and scalable with a smaller in-house dev team.

Internet Connection: Although some data will be downloadable and available off-line, this system requires an internet connection for most of its functionality. Data modifications will only be available when online to prevent data redundancy or other conflicting data.