**BUS RESERVATION SYSTEM**

**EXPERIMENT-5**

**To Perform UML Diagrams Using Argo**

# 5.1 Introduction to UML

Unified Modelling Language (UML) is a standardized modelling language used in software engineering for visualizing, specifying, constructing, and documenting the artifacts of a software system. It provides a common language for developers, analysts, designers, and stakeholders to communicate and understand system requirements, design decisions, and system behaviour.

Here's a brief introduction to some key concepts in UML:

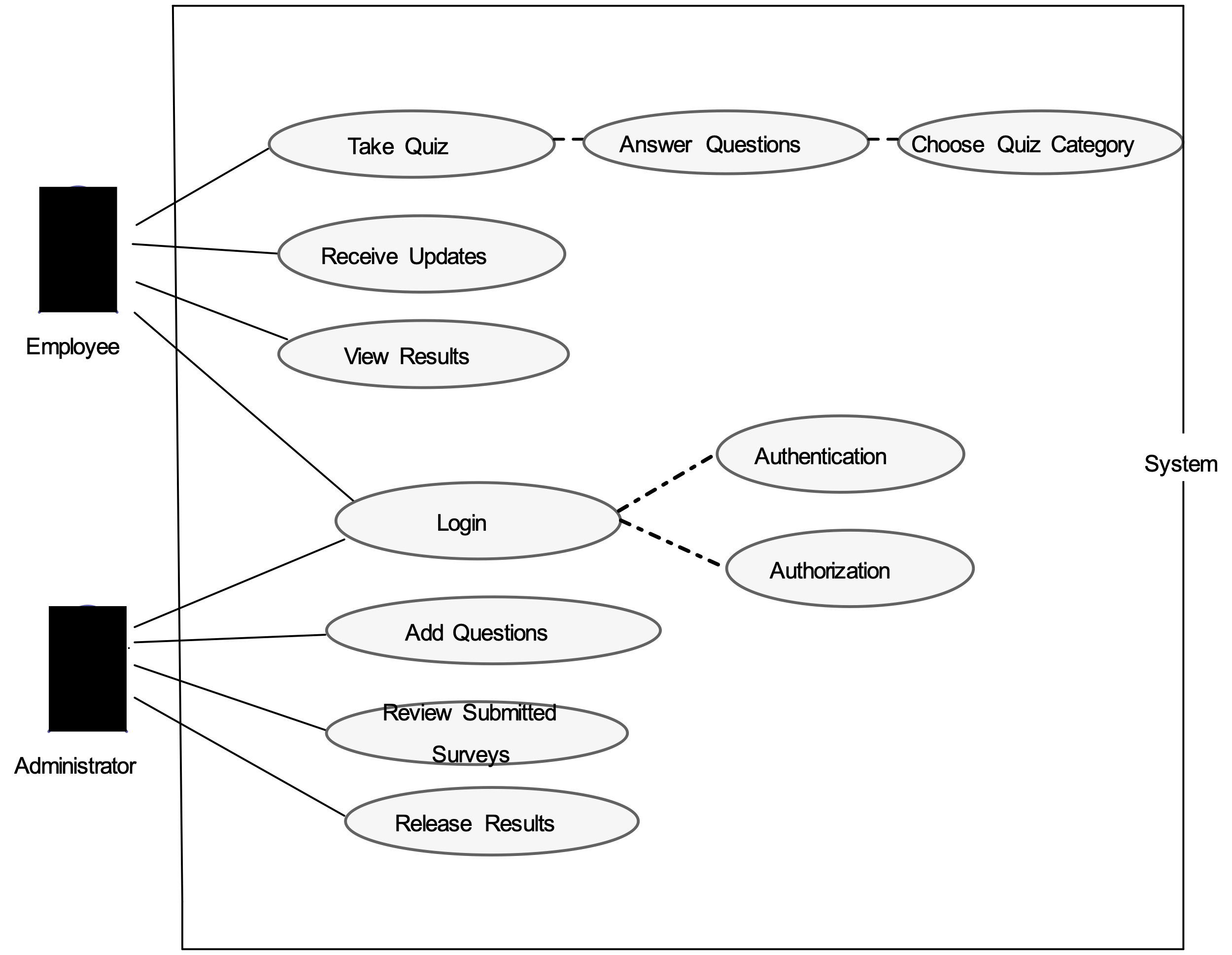
1. **Diagrams**: UML offers several types of diagrams to represent different aspects of a system. Some common types include:
   * **Class Diagram**: Represents the structure of a system by showing the classes, their attributes, methods, and relationships.
   * **Use Case Diagram**: Depicts the functional requirements of a system by illustrating the interactions between actors (users or external systems) and the system itself.
   * **Sequence Diagram**: Shows how objects interact over time in a particular scenario or use case.
   * **Activity Diagram**: Represents workflows or business processes, showing the flow of activities and decision points.
   * **State Diagram**: Describes the various states that an object can have and how it transitions between those states in response to events.
2. **Elements**: UML diagrams consist of various elements such as classes, interfaces, objects, associations, dependencies, and more. Each element has specific symbols and notation to represent it visually.
3. **Relationships**: UML allows you to depict relationships between elements. Some common relationships include:
   * **Association**: Represents a structural relationship between two or more classes, indicating that objects of one class are connected to objects of another class.
   * **Generalization/Inheritance**: Depicts an "is-a" relationship between classes, where one class (subclass) inherits attributes and behaviour from another class (superclass).
   * **Dependency**: Indicates that one element (e.g., a class) relies on another element (e.g., a class or interface) in some way, such as method invocation or parameter passing.
4. **Behavioural Modelling**: UML not only models the structure of a system but also its behaviour.

This includes capturing how objects interact with each other, the sequence of those interactions, and the changes in the state of objects over time.

1. **Tool Support**: Various tools are available to create and manipulate UML diagrams, ranging from simple drawing tools to sophisticated CASE (Computer-Aided Software Engineering) tools that offer features like code generation, reverse engineering, and model validation.

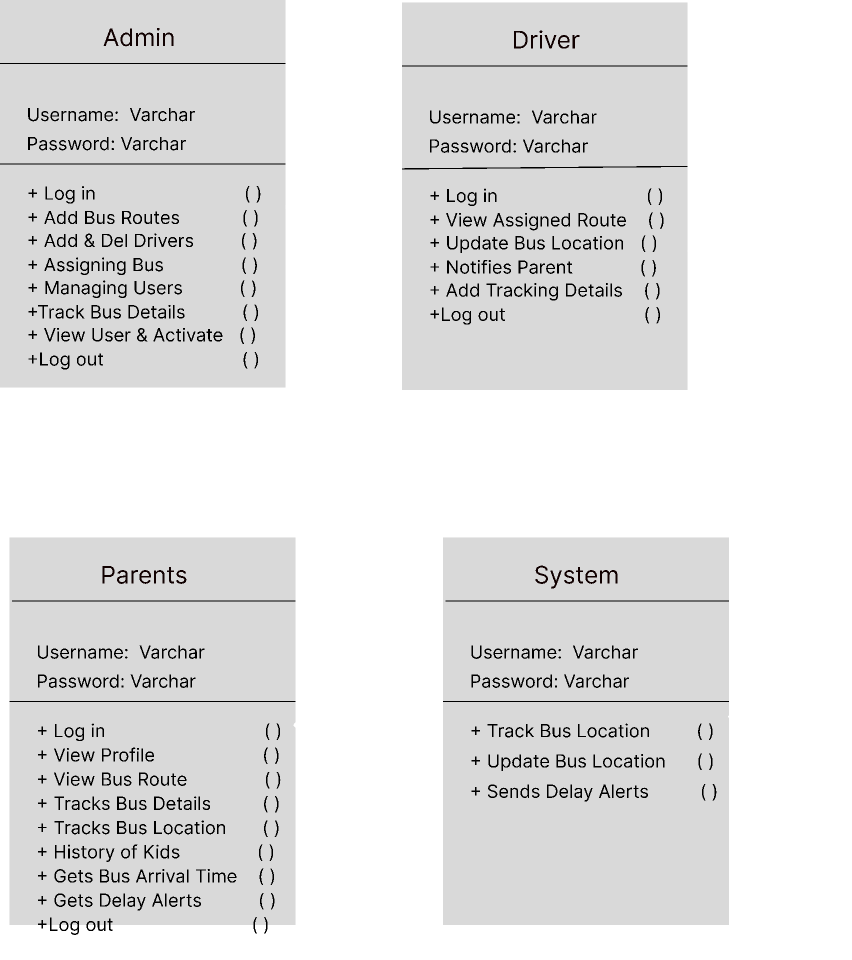
# Types of UML Diagram

1. Use Case Diagram
2. Class Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. State Diagram

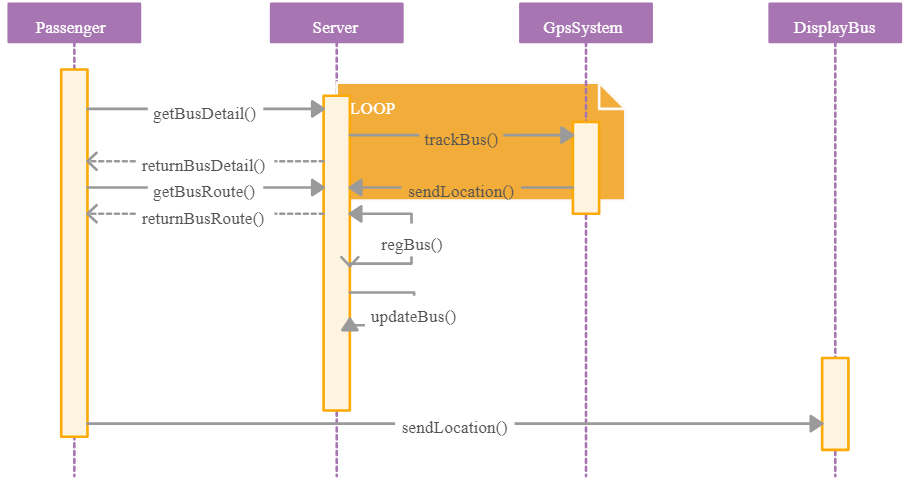


# 1.) Use Case Diagram

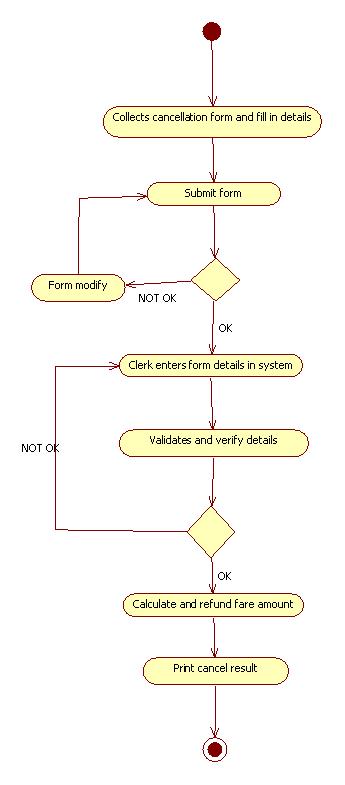
**2. Class Diagram**



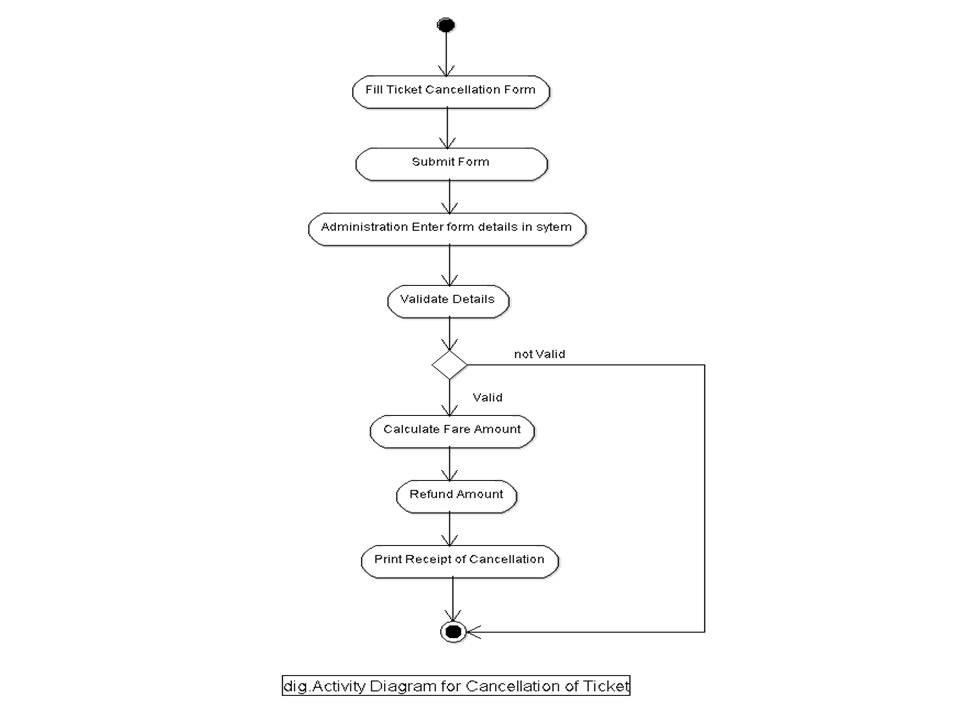
# 3.) Sequence Diagram



# 4) Collaboration diagrams



1. **State Diagram**

****

# 5.2) INTRODUCE ARGO UML TOOL

Argo UML is an open-source UML modelling tool designed to support all standard UML diagrams. It provides software designers and developers with a platform to create, visualize, and analyse their software systems using UML notation. Here's an introduction to Argo UML:

1. **Open Source:** Argo UML is an open-source project, which means it's freely available to download, use, modify, and distribute. This makes it accessible to a wide range of users, including individual developers, small teams, and larger organizations.
2. **Platform Independence:** Argo UML is built using Java, making t. It can run on various operating systems, including Windows, macOS, and Linux, as long as Java is installed.
3. **Support for UML Standards:** Argo UML supports all standard UML diagrams defined by the

Object Management Group (OMG), including Class Diagrams, Use Case Diagrams, Sequence Diagrams, Activity Diagrams, State Diagrams, and more. This comprehensive support allows users to model different aspects of their software systems effectively.

1. **User-Friendly Interface:** Argo UML features a user-friendly interface that allows users to create and manipulate UML diagrams easily. It provides intuitive tools for adding elements, defining relationships, and arranging components within diagrams. The interface aims to streamline the modelling process and enhance productivity.
2. **Extensibility:** Argo UML supports extensions through modules and plugins, allowing users to extend its functionality according to their specific needs. This extensibility enables the integration of additional features, such as code generation, model validation, and support for domain-specific languages.
3. **Collaboration:** Argo UML facilitates collaboration among team members by supporting features like version control integration (e.g., Git), allowing multiple users to work on the same models concurrently. This promotes teamwork and enhances communication within development teams.
4. **Documentation and Code Generation:** Argo UML enables users to generate documentation from their UML models, helping to keep project stakeholders informed about system design and requirements. Additionally, it supports code generation, allowing developers to generate source code from UML diagrams, thus bridging the gap between design and implementation