**UML diagrams: used in the design of flowcharts of the software application**

A UML (Unified Modelling Language) diagram visually represents a system or software application using standard notation. It is used to design, analyze, and communicate the structure and behavior of a software system or application. UML diagrams are a popular modeling tool software developer, system analysts, and designers use.

There are several types of UML diagrams, each serving a different purpose. Some of the most common types of UML diagrams include:

Class diagram: This shows the structure of a system, including classes, attributes, methods, and relationships between objects.

Use case diagram: Illustrates the interactions between the user and the system and identifies the use cases the system will support.

Sequence diagram: Shows the interactions between objects and the order in which they occur, typically used to model the behavior of a single use case.

Activity diagram: Illustrates the flow of activities within a system, including branching, parallelism, and synchronization.

State diagram: Depicts the lifecycle of an object, including the different states that it can be in and the events that cause it to transition from one state to another.

Component diagram: Depicts the physical components of a system and how they interact with each other.

Deployment diagram: Illustrates the deployment of system components on physical hardware.

Overall, UML diagrams are essential for software development and system analysis, allowing developers and analysts to model and communicate complex systems standardized and efficiently.

**UML diagram symbols and their use.**

start SymbolStart symbol: Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point.

activity SymbolActivity symbol: Indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram.

connector SymbolConnector symbol: Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity; once the step is completed, the flow continues with the outgoing arrow.

joint SymbolJoint symbol/ Synchronization bar: Combines two concurrent activities and re-introduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line.

fork SymbolFork symbol: Splits a single activity flow into two concurrent activities. Symbolized with multiple arrowed lines from a join.

decision SymbolDecision symbol: Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or merging of various flows with the symbol acting as a frame or container.

note SymbolNote symbol: Allows the diagram creators or collaborators to communicate additional messages that don't fit within the diagram itself. Leave notes for added clarity and specification.

send signal SymbolSend signal symbol: Indicates that a signal is being sent to a receiving activity.

receive signal SymbolReceive signal symbol: Demonstrates the acceptance of an event. After the event is received, the flow that comes from this action is completed.

shallow history pseudostate symbolShallow history pseudostate symbol: Represents a transition that invokes the last active state.

option loop symbolOption loop symbol: Allows the creator to model a repetitive sequence within the option loop symbol.

flow final symbolFlow final symbol: Represents the end of a specific process flow. This symbol shouldn’t represent the end of all flows in an activity; in that instance, you would use the end symbol. The flow final symbol should be placed at the end of a process in a single activity flow.

condition textCondition text: Placed next to a decision marker to let you know under what condition an activity flow should split off in that direction.

end symbolEnd symbol: Marks the end state of an activity and represents the completion of all flows of a process.

Create notes

User

Create notes

Save notes

Retrieve

Edit

**Diagram 1:** Case scenario of the Application.

**Diagram 2:** Flowchart

Start

Create notes

Save notes?

Save

Discard

No

Yes

Input notes

Database

Retrieve notes

Read/edit notes