1. Discuss which UML models are most applicable at different stages of the Software Development Life Cycle

Unified Modeling Language (UML) provides standardized modeling techniques to design a system before implementation which support the various stages of the Software Development Life Cycle (SDLC) (Geeksforgeeks, 2024). Each UML model serves a different purpose and is most effective at specific stages of the SDLC:

Requirement Analysis:

Use Case Diagrams: These are useful for capturing the system's functional requirements by visualizing the interactions between the users (actors) and the system itself. Use Case diagram visualizes what the software or the system is supposed to do from a user's perspective,

System Design:

Class Diagrams: define the system's static structure by showing its classes, attributes, methods, and relationships among them. They show all of the components (classes, attributes, and methods) and how they interact with each other in an object-oriented system.

Sequence Diagrams depict the sequence of messages exchanged between objects to perform a function. They are valuable in understanding object interactions overtime during the design and implementation phases.

Activity Diagrams: These represent the flow of activities or or processes in a system. They help model the logic of complex methods and actions, often showing parallel behaviors (Geeksforgeeks, 2024).

Implementation:

State Machine Diagrams: describe the states an object goes through in response to events, showing how an object’s state changes over time. They are handy for modeling the behavior of state-dependent systems, for example an order processing system. If an activity diagram depicts the sequence of actions and activities that can happen in a system, state diagram represents how object’s state changes in relation to those activities.

Deployment Diagrams: These depict the physical deployment of artifacts on nodes. During the deployment phase, they can be useful to visualise how the software will be deployed on hardware components.

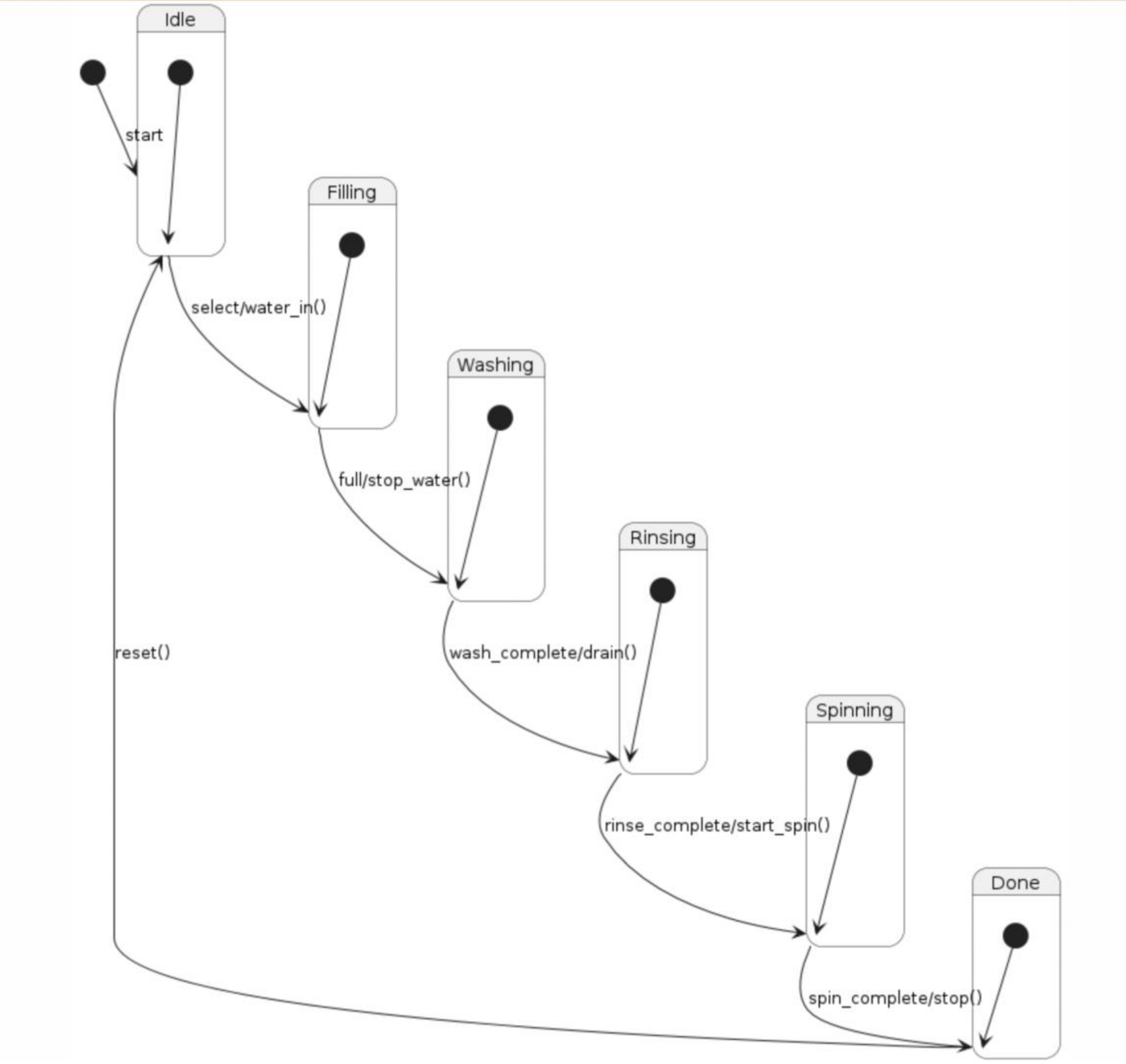
Maintenance:

Component Diagrams: show how the system is divided into the dependencies among these components. They help define the system's modular structure. This is especially useful in enterprise systems with many components that can be reused and combined with each other in various ways.

(Rumbaugh, et al., 2004)

2. Making reference to ‘The Unified Modeling Language Reference Manual Second Edition’, use the State Machine Diagram in Figure 3-7 to design a similar model for a washing machine

Based on the State Machine Diagram in Figure 3-7 from "The Unified Modeling Language Reference Manual Second Edition", a similar model for a washing machine was designed



Machine states:

Idle State: The initial state before starting the washing machine.

Filling State: Once the washing machine is selected, it fills the water.

Washing State: After the water is filled, the washing process begins.

Rinsing State: After washing, the machine drains the water and rinses.

Spinning State: After rinsing, the machine starts to spin and remove excess water.

Done State: The final state where the washing cycle is complete.

References:

Geeksforgeeks (2024) Is UML part of SDLC?. Available from: https://www.geeksforgeeks.org/is-uml-part-of-sdlc/ [Accessed 2 June 2024].

Geeksforgeeks (2024 Differences between Sequence Diagram and Activity Diagram. Available from: geeksforgeeks.org/difference-between-sequence-diagram-and-activity-diagram/ [Accessed 2 June 2024].

Rumbaugh, J., Jacobson, I. & Booch, G. (2004) The Unified Modeling Language Reference Manual. 2nd ed. Addison-Wesley.