Practical - 01.

Subject - Object oriented modelling and design.
Assignment title - Draw state machine diagrams representing system states and transitions (whatsapp example).
Semester / Year - Fourth Year VII Sem.
Instructor - Saiprasad Bhise.
Submission Date: 15/09/2025.

1] OBJECTIVE:

- Understand and apply UML standards for modeling system behaviour using state machine diagrams.
- Capture dynamic states, events and transitions of system entities.
- Communicate lifecycle and state-dependent behaviour clearly to stakeholders.

2] Problem statement -

- Draw one or more state machine diagrams to represent the state changes of a chosen systems key entities.
- Complement the diagrams with detailed descriptions of states, transitions, events and actions.

Example system:
Whatsapp message lifecycle.

3] Introduction to state machine diagram modelling -

3. State machine diagrams (also called state charts) show the lifecycle of an object through states, transitions triggered by events and actions performed. They model reactive behaviour by focusing on how an object changes state in response to external and internal events. State machines help clarify complex dynamic behaviour and lifestyle management in software systems.

4] Theory and Best practices -

UML Elements -

- State: Represents a condition or situation during the lifestyle of an object (rounded rectangle).
- Initial state - Starting point of the state machine (filled black circle).
- Final state - End of lifecycle (circle with dot inside).
- Transition - Arrow between states triggered by an event optionally with a guard condition and action (eg. [condition] / action).
- Event - External or internal trigger causing a state change.
- Action - Activity performed on entering, exiting or during a state.

- Composite state :- A state containing nested substates.

- Naming notation -
  - State names should be nouns or noun phases (eg message sent).
  - Transitions labeled with triggering events, optional guards, and actions.
  - Use clear and minimalistic layout for readability.

5] Assignment Workflow -

1. System definition and boundary - describe the entity to be modeled (eg, whatsapp message object lifestyle)
2. Identify states - List all relevant states the entity can be in during its life style.
3. Identify events and transitions - determine events that triggers transitions and actions performed.
4. Draw state machine diagram - model, states, transitions, events and actions using UML notations.
5. Doument State descriptions - Provide detailed templates for at least two key states including entry, exit actions and associated events.
6. Stakeholder Validation (Hypothetical) - Explain how reviews and feedback would ensure the correctness and completeness of the state model.

6] Deliverables -
- Title page.
- Introduction.
- Clearly labelled diagrams.
- Optional notes.
- detailed state templates.

7] Evaluation Criteria -
- Correct and clear use of UML state machine notation.
- Completeness of states, events, transitions and actions.
- Clarity of state descriptions and lifecycle flow.
- Presentation quality and formatting.
- Timely submission.

8] Recommended tools -
- UML tools - Draw IO, Microsoft Visio, Lucidchart.
- Collaborative tools - Miro, confluence.

9] System description -

Whatsapp message lifecycle involves multiple states from composition to delivery and read confirmation.
This assignment models the message object state transitions from creation through sending delivery and reading.

10] Assignment requirements -

- System boundary - define entity lifecycle scope
- States - Identify meaningful States (eg created, sent, delivered, read).
- Events - Include events triggering transitions (eg. send, deliver, read reciept)
- Transition - Correctly model events, guards, actions.
- Diagram - Standard UML notation, readable layout.
- Templates - Detailed description of states and transitions for at least two key states.
- Stakeholder validation - reflection or example feedback to verify design.
- Documentation - Professional formatting and clarity.

11] Sample states machine diagram overview : WhatsApp Message lifecycle.

States -

- Drafting
- Ready to send.
- Sent.
- Delivered
- read.
- Failed.

Transitions and events -

- Compose → [User finishes message] → ready to send.
- Ready to send → [send event] → Sent.
- Sent → [Delivery receipt] → delivered.
- delivered → [read receipt] → read.
- Sent → [Send feilure] → Failed.

12] State template structure -

State Name: Sent.
- Description - Message has been sent from the client but not yet delivered.
- Entry Action - Store timestamp of sending.
- Exit action - Await delivery receipt.
- Events triggering transitions -
  - Delivery receipt received - Transition to delivered.
  - Send failure - Transition to failed.
- Notes - Message status displayed as sent in UI.

13] Stakeholder Validation -
- Conduct walkthrough with product owners and developers.
- Validate that all realistic message states and transitions are covered.
- Adjust for missing events or error handling scenarios.

14] Conclusion -

State machine diagrams provide a precise view of an object's lifecycle and dynamic behaviour. Modelling the whatsapp message lifecycle clarifies system responsiveness and status management, critical for real time messaging apps. Mastery of state modelling supports better design, testing and implementation of reactive systems.