		Hope Foundation's Finolex Academy of Management and Technology, Ratnagiri Department of Information Technology	
Subject name: Software Design Lab			Subject Code: ITL601
Class	TE IT	Semester –VI (CBCGS)	Academic year: 2018-19
Name of Student	Kazi Jawwad A Rahim		QUIZ Score : 06
Roll No	27	Assignment/Experiment No.	08
Title: Designing of State chart Diagram.			

1. Lab objectives applicable: LOB4.

2. Lab outcomes applicable: LO4

3. Learning Objectives:

1. To learn about the different components of state chart diagram.
2. To understand how it can be used to represent the dynamic nature of information system.
3. To identify the events causing transitions from one state to another.

4. Practical applications of the assignment/experiment: Diagrams are used in order to give an insight for the activities involved in the functionality of a system.

5. Prerequisites:

1. SRS

6. Hardware Requirements:

Windows operating system (Windows 7 or higher)

7. Software Requirements:

UML designing tool such as IBM Rational Rose/StarUML

8. Quiz Questions (if any): (Online Exam will be taken separately batch-wise, attach the certificate/ Marks obtained)

1. What does a state transition diagram describe?
2. What is the purpose behind using state transition diagram?

9. Experiment/Assignment Evaluation:

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A <u>or</u> any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
Date of performance (DOP)		Total marks obtained	10

Signature of the faculty

10. Theory:

Statechart Diagrams

In case of Object-Oriented Analysis and Design, a system is often abstracted by one or more classes with some well-defined behavior and states. A statechart diagram is a pictorial representation of such a system, with all its states, and different events that lead transition from one state to another.

To illustrate this, consider a computer. Some possible states that it could have are: running, shutdown, hibernate. A transition from running state to shutdown state occurs when user presses the "Power off" switch, or clicks on the "Shut down" button as displayed by the OS. Here, clicking on the shutdown button, or pressing the power off switch act as external events causing the transition.

Statechart diagrams are normally drawn to model the behavior of a complex system. For simple systems this is optional.

Building Blocks of a Statechart Diagram

- **State**

A state is any "distinct" stage that an object (system) passes through in its lifetime. An object remains in a given state for finite time until "something" happens, which makes it to move to another state. All such states can be broadly categorized into following three types:

Initial: The state in which an object remains when created

Final: The state from which an object does not move to any other state [optional]

Intermediate: Any state, which is neither initial, nor final

As shown in figure-01, an initial state is represented by a circle filled with black. An intermediate state is depicted by a rectangle with rounded corners. A final state is represented by a unfilled circle with an inner black-filled circle

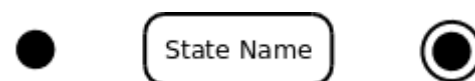


Figure-01: Representation of initial, intermediate, and final states of a state chart diagram. Intermediate states usually have two compartments, separated by a horizontal line, called the name compartment and internal transitions compartment.

They are described below:

Name compartment: Contains the name of the state, which is a short, simple, descriptive string

Internal transitions compartment: Contains a list of internal activities performed as long as the system is in this state

The internal activities are indicated using the following syntax: action-label / action-expression. Action labels could be any condition indicator. There are, however, four special action labels:

Entry: Indicates activity performed when the system enters this state

Exit: Indicates activity performed when the system exits this state

Do: indicate any activity that is performed while the system remain in this state or until the action expression results in a completed computation

Include: Indicates invocation of a sub-machine

Any other action label identify the event (internal transition) as a result of which the corresponding action is triggered. Internal transition is almost similar to self transition, except that the former doesn't result in execution of entry and exit actions. That is, system doesn't exit or re-enter that state. Figure-02 shows the syntax for representing a typical (intermediate) state.

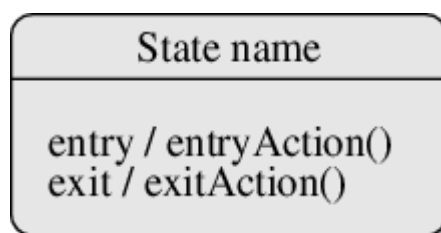


Figure-02: A typical state in a statechart diagram States could again be either simple or composite (a state containing other states). Here, however, we will deal only with simple states.

- **Transition**

Transition is movement from one state to another state in response to an external stimulus (or any internal event). A transition is represented by a solid arrow from the current state to the next state. It is labeled by: event [guard-condition]/[action-expression], where

Event is the what is causing the concerned transition (mandatory) -- Written in past tense

Guard-condition is (are) precondition(s), which must be true for the transition to happen [optional]

Action-expression indicate action(s) to be performed as a result of the transition [optional]

It may be noted that if a transition is triggered with one or more guard-condition(s), which evaluate to false, the system will continue to stay in the present state. Also, not all transitions do result in a state change. For example, if a queue is full, any further attempt to append will fail until the delete method is invoked at least once. Thus, state of the queue doesn't change in this duration.

- **Action**

As mentioned in actions represents behavior of the system. While the system is performing any action for the current event, it doesn't accept or process any new event. The order in which different actions are executed, is given below:

Exit actions of the present state

Actions specified for the transition

Entry actions of the next state

Figure-03 shows a typical statechart diagram with all its syntaxes.

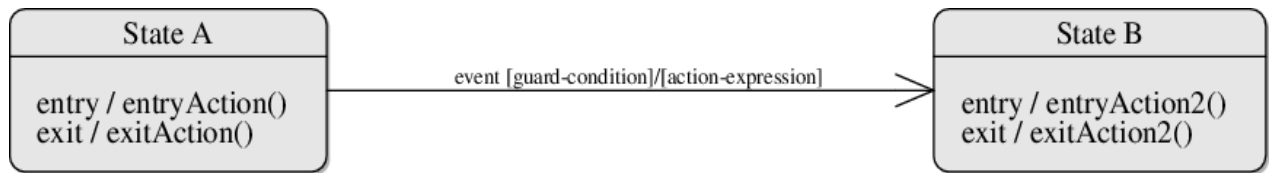


Figure-03: A statechart diagram showing transition from state A to B

Guidelines for drawing Statechart Diagrams

Following steps could be followed, as suggested to draw a statechart diagram:

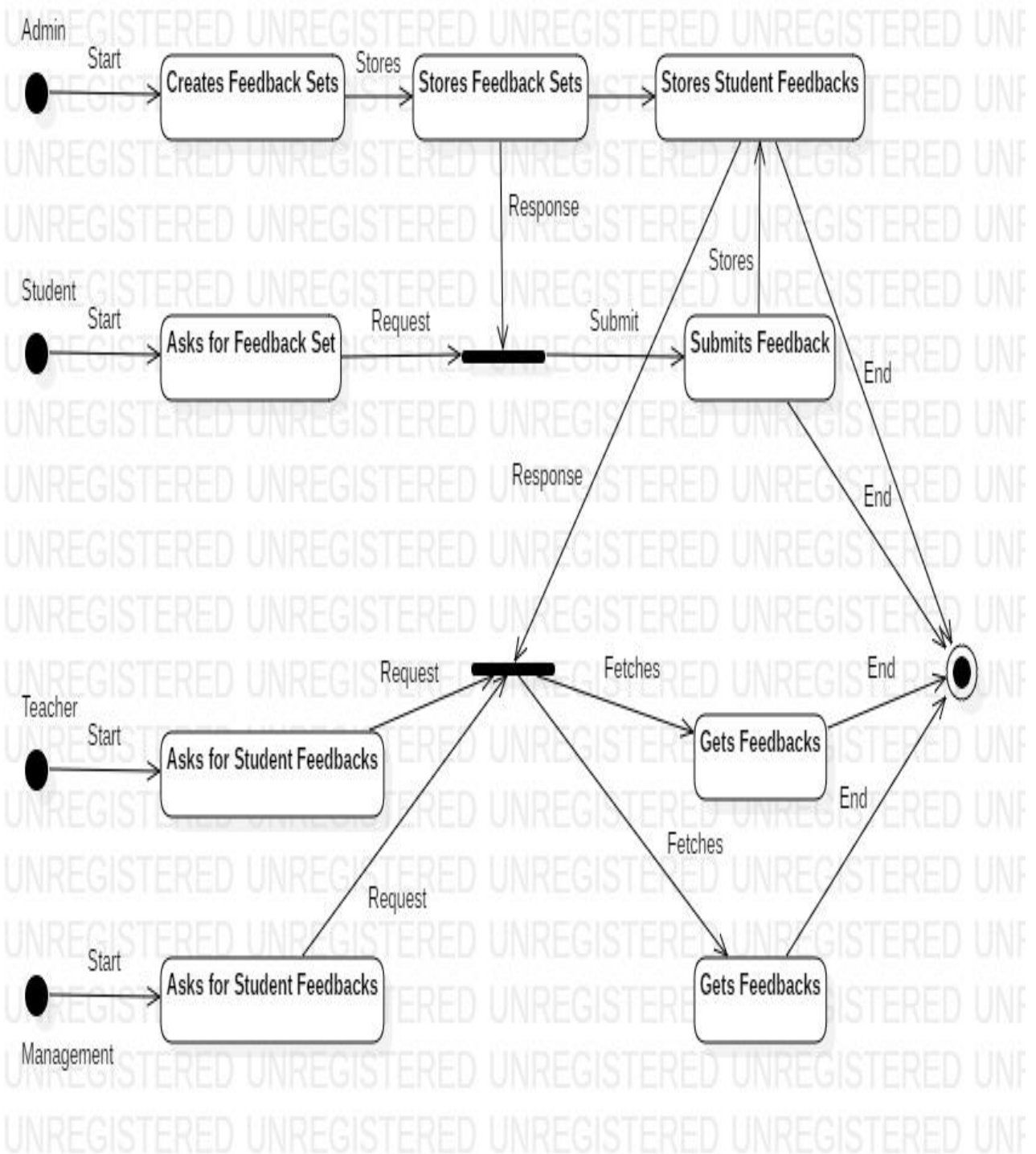
For the system to developed, identify the distinct states that it passes through

Identify the events (and any precondition) that cause the state transitions. Often these would be the methods of a class as identified in a class diagram.

Identify what activities are performed while the system remains in a given state

11. Source code / Diagrams:

Online Feedback System



References:

- [1]. <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>
- [2]. <https://www.tutorialspoint.com/uml/>
- [3]. <http://vlabs.iitkgp.ernet.in/se/6/theory/>