31269: Business Requirements Modelling

Week 10 Lecture - Object Oriented Models with UML - State and Event Modelling

- ✓ References
  - ✓ Object Oriented Systems Analysis and Design Using UML, 4th Edition (Chapter 11) **by Bennnett, McRobb and Farmer**

Object Oriented Modelling

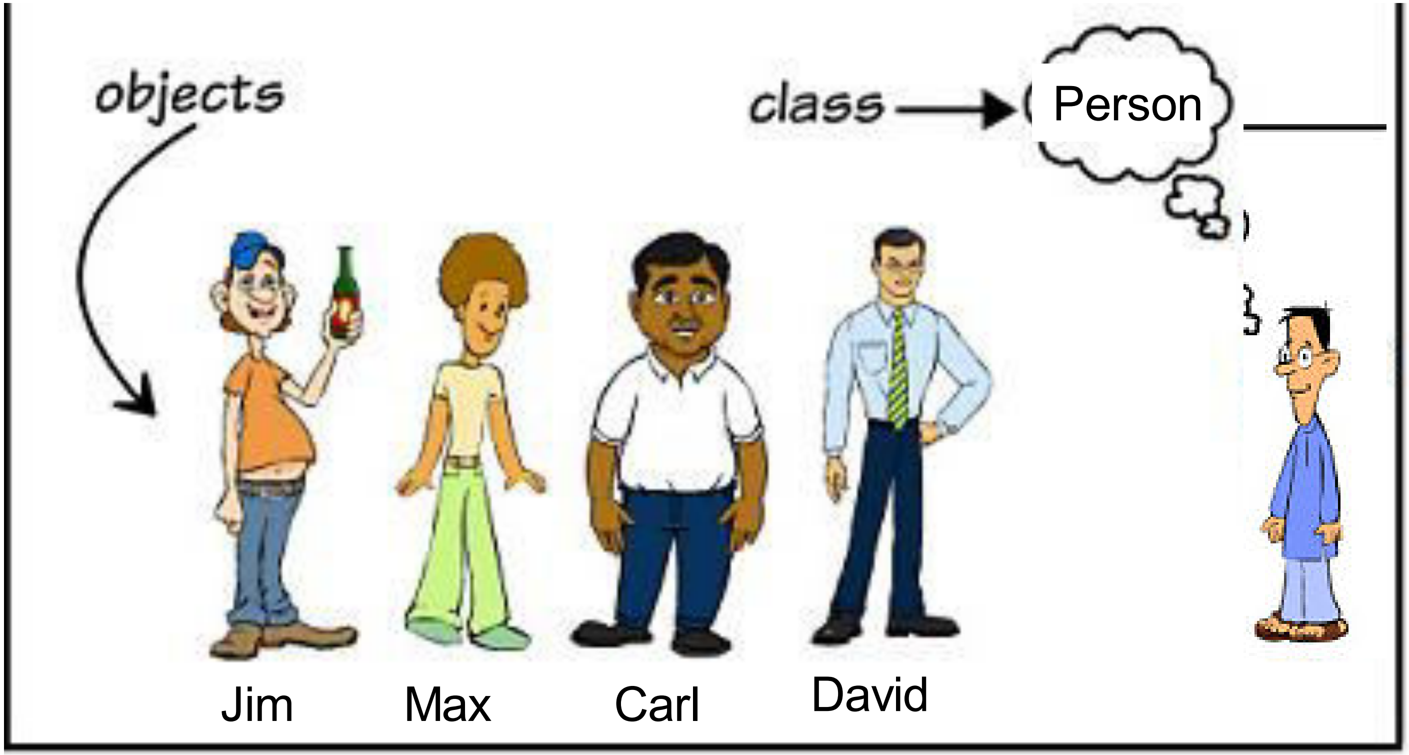
- Last Week
  - Interaction Modelling (Sequence Diagrams)
  - Message passing between objects
  - Synchronous and Asynchronous messages

- This Week
  - State and Event Modelling (State Transition Diagrams)
  - Show different states of an object.

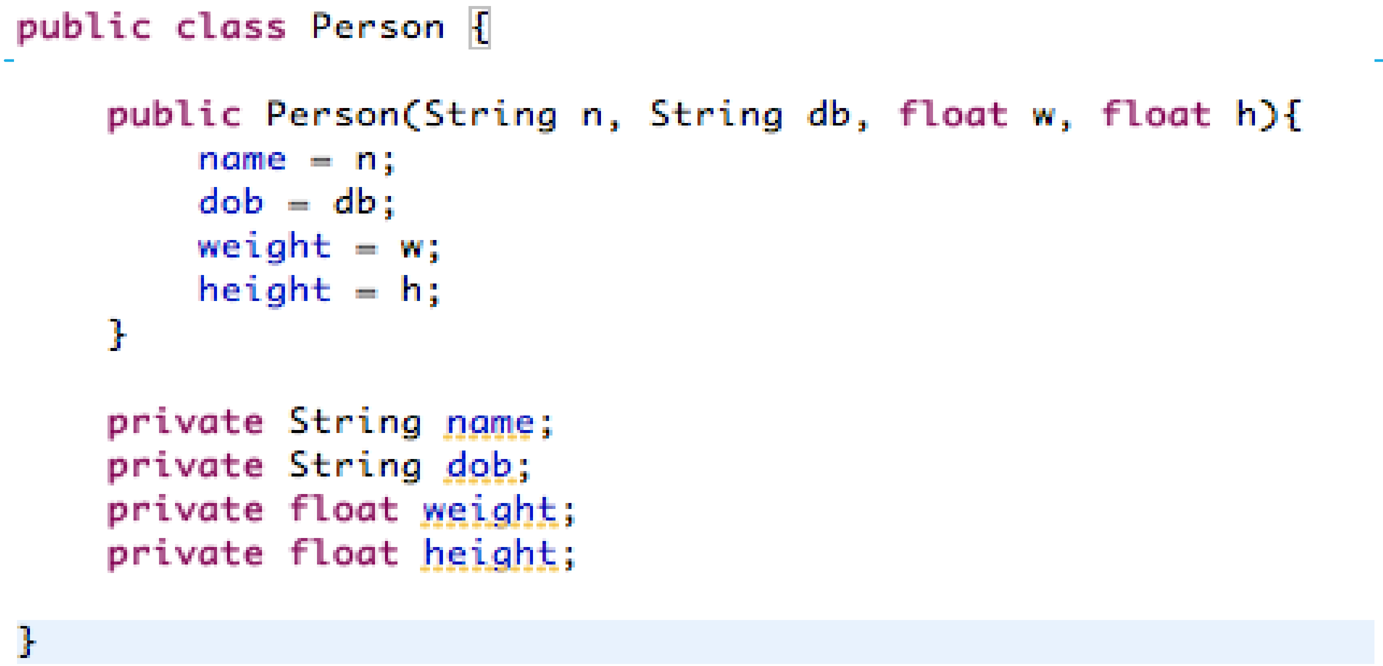
Classes and Objects

- A Class is a definition of Objects.
- A Class is a template or a blueprint for creating Objects.
- An Object is an instance of a Class. One to many objects can be created from a class.

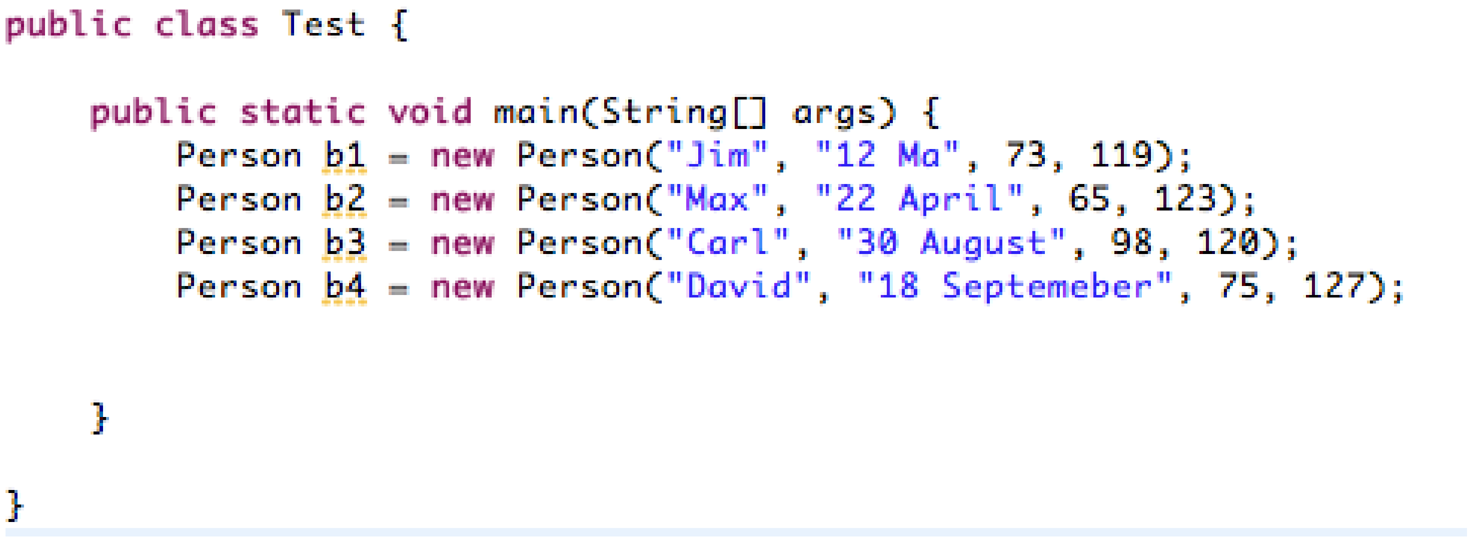
Classes and Objects



Classes and Objects



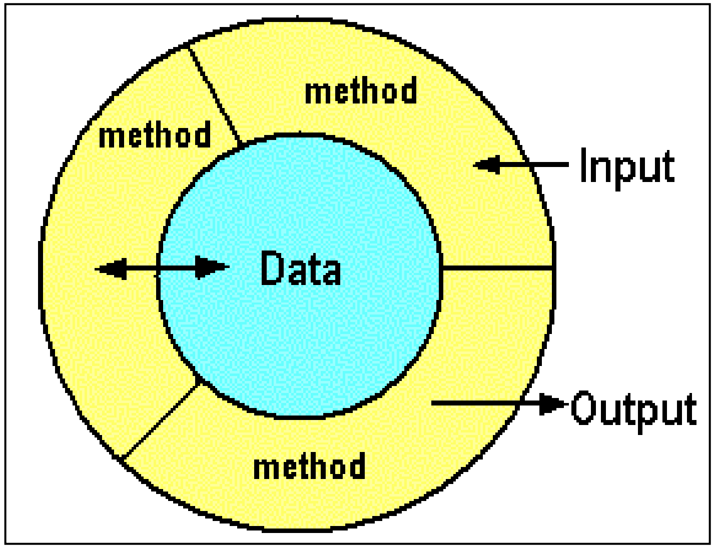
```
public class Person {

    public Person(String n, String db, float w, float h){
        name = n;
        dob = db;
        weight = w;
        height = h;
    }

    private String name;
    private String dob;
    private float weight;
    private float height;

}
```

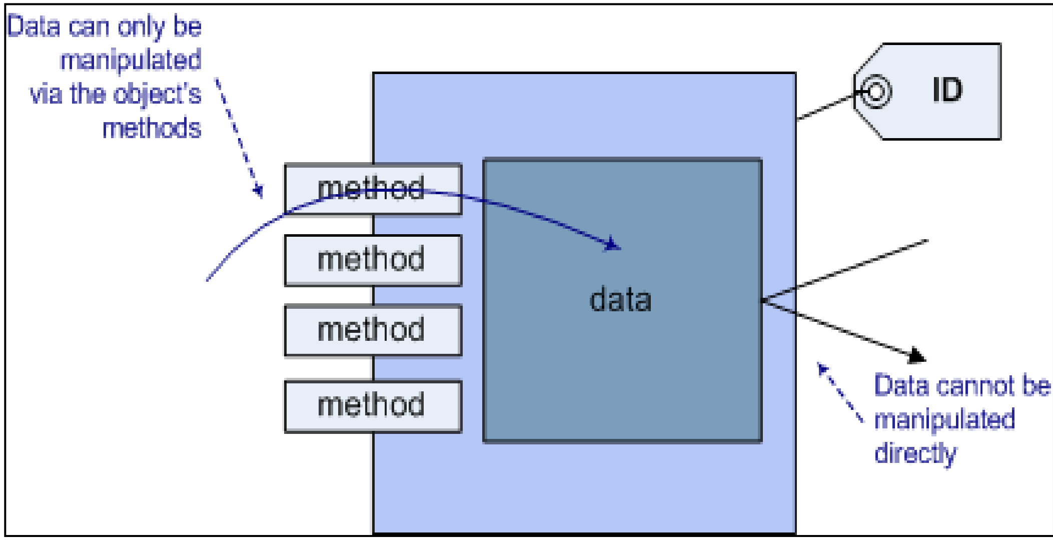


```
public class Test {

    public static void main(String[] args) {
        Person b1 = new Person("Jim", "12 Ma", 73, 119);
        Person b2 = new Person("Max", "22 April", 65, 123);
        Person b3 = new Person("Carl", "30 August", 98, 120);
        Person b4 = new Person("David", "18 Septemeber", 75, 127);


    }

}
```

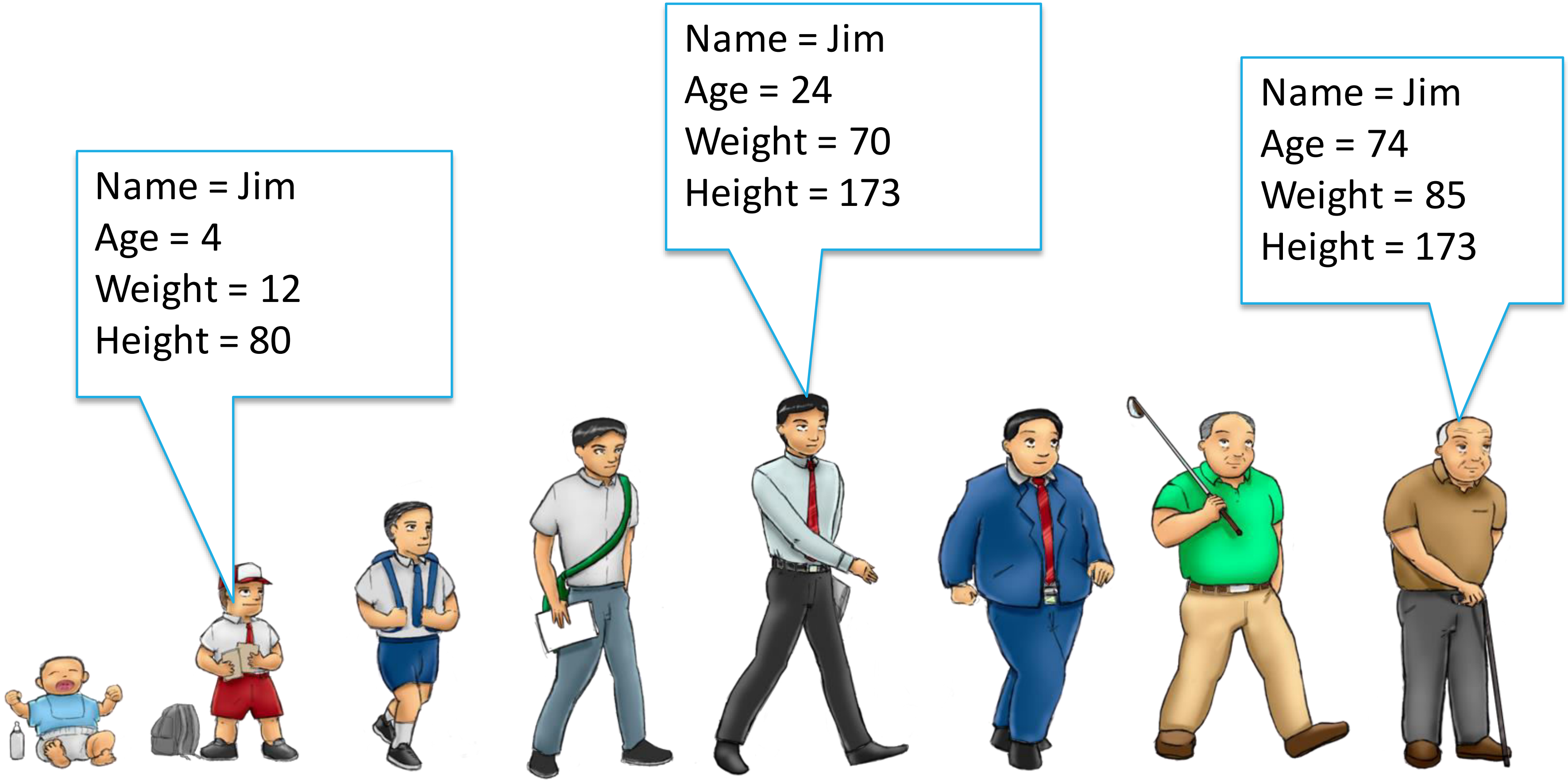
Objects have…

- Properties/Data/**State** that is defined by
  - Attributes of the class
- **Behavior** that is defined by
  - Methods of the class




Data can only be manipulated via the object's methods
ID
method
method
method
method
data
Data cannot be manipulated directly

Objects have “States”



Name = Jim
Age = 4
Weight = 12
Height = 80

Name = Jim
Age = 24
Weight = 70
Height = 173

Name = Jim
Age = 74
Weight = 85
Height = 173

What has changed in these objects?

State and Event Modelling

- A state transition diagram shows the various states of a **single object**.
  - There must be a separate state transition diagram for each object/class in your class diagram.
  - It helps analysts, designers and developers **understand the behaviour of the objects** in the system. They won't have to guess about **what the object is supposed to do**.
  - State-transition diagrams are very useful for describing the **behavior of individual objects over the full set of use cases** that affect those objects.
  - Create State Transition Diagrams when the business logic for a particular flow is very complex, or needs greater understanding by the developer to be coded perfectly.

Why State and Event Modelling?

- To track an object’s lifecycle.
- To provide status of an object at a given point in time.
- To better understand the several states that an object goes through.

State and Event Modelling

- Show an object’s **states,** and the **events** that cause them to transition between states.
- Movement from one state to another is called **transition,** and is triggered by an event. When its triggering event occurs a transition is said to fire.
- An event happens at a specific time and place.
  - Events cause a change of state for an object as the transition “fires”

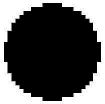
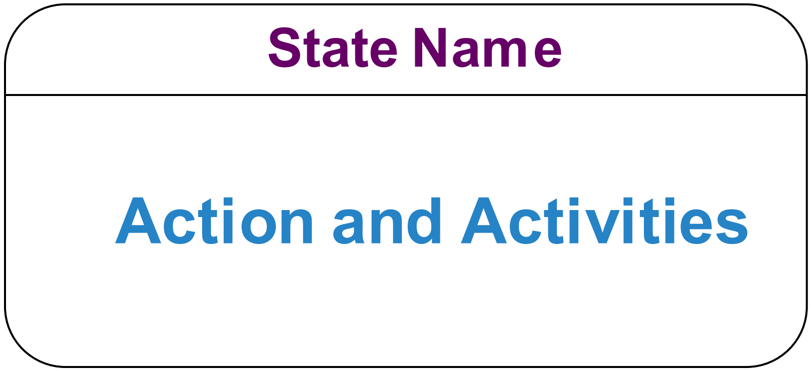
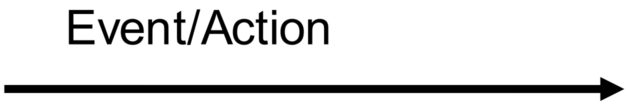
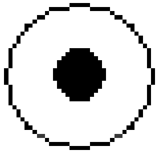
State and Event Modelling

- Objects change their ***state*** in response to events (time and non-time events).
  - e.g. When you *press a switch/button*, a Light object changes its state from ***off*** to ***on***.
  - e.g. The *changeCourse()* method changes the state of the Student object.
  - e.g. The enrol*()* method changes the state of the Student object from being ‘prospective’ to ‘enrolled’.

- Each time an object changes state, some of its attributes must change.

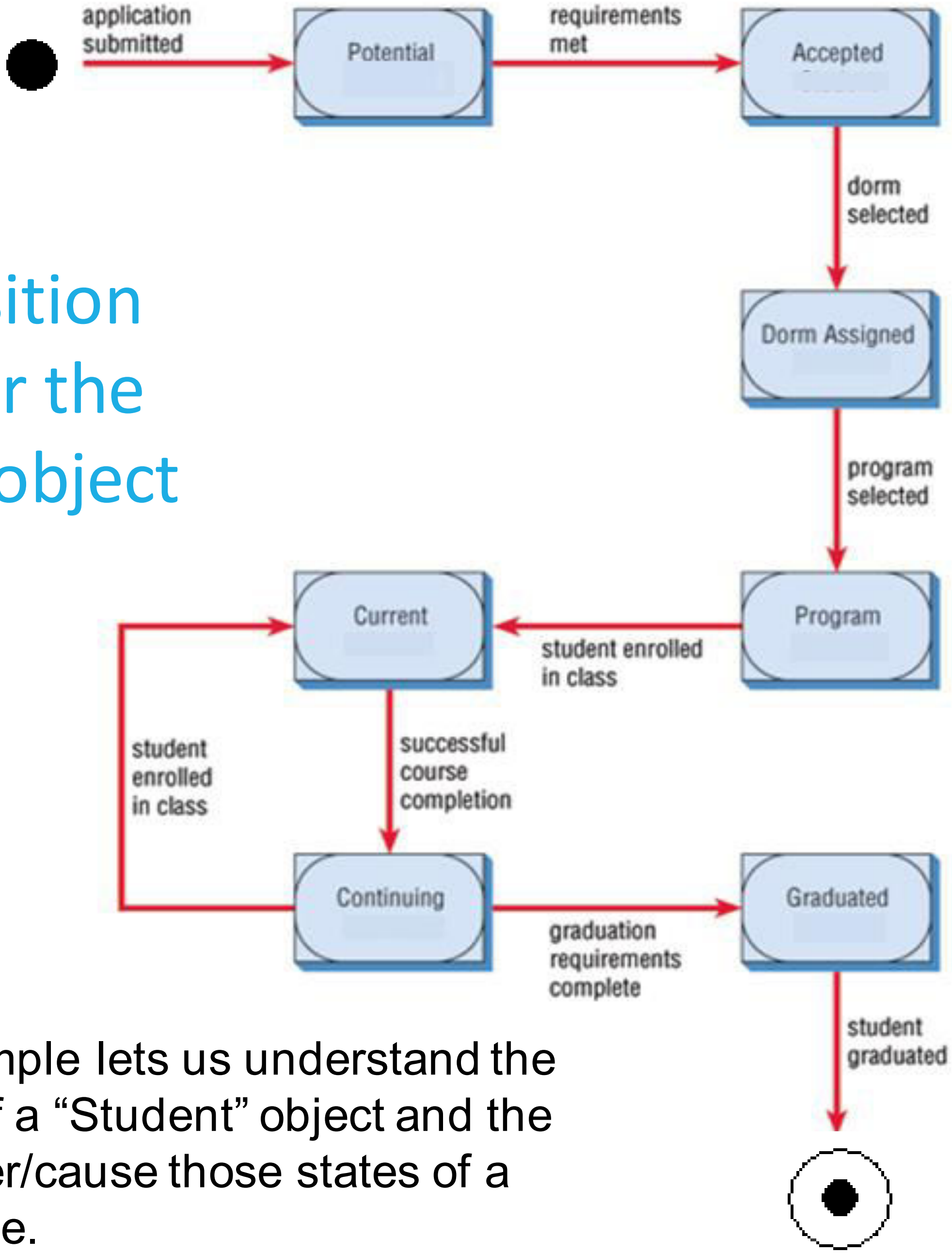
UML Syntax for State Transition

ref http://www.developer.com/design/article.php/2238131

	**Initial State:** This shows the starting point or first activity of the flow. Denoted by a solid circle.
 State Name Action and Activities	**State:** Represents the state of an object at an instant of time; one for each state of the Object we are describing. Denoted by a rectangle with rounded corners and compartments
 Event/Action	**Transition:** An arrow indicating the Object is to transition from one state to the other. The actual trigger event and action causing the transition are written besides the arrow, separated by a slash.
	**Final State:** The end of the state diagram is shown by a bull's eye symbol.

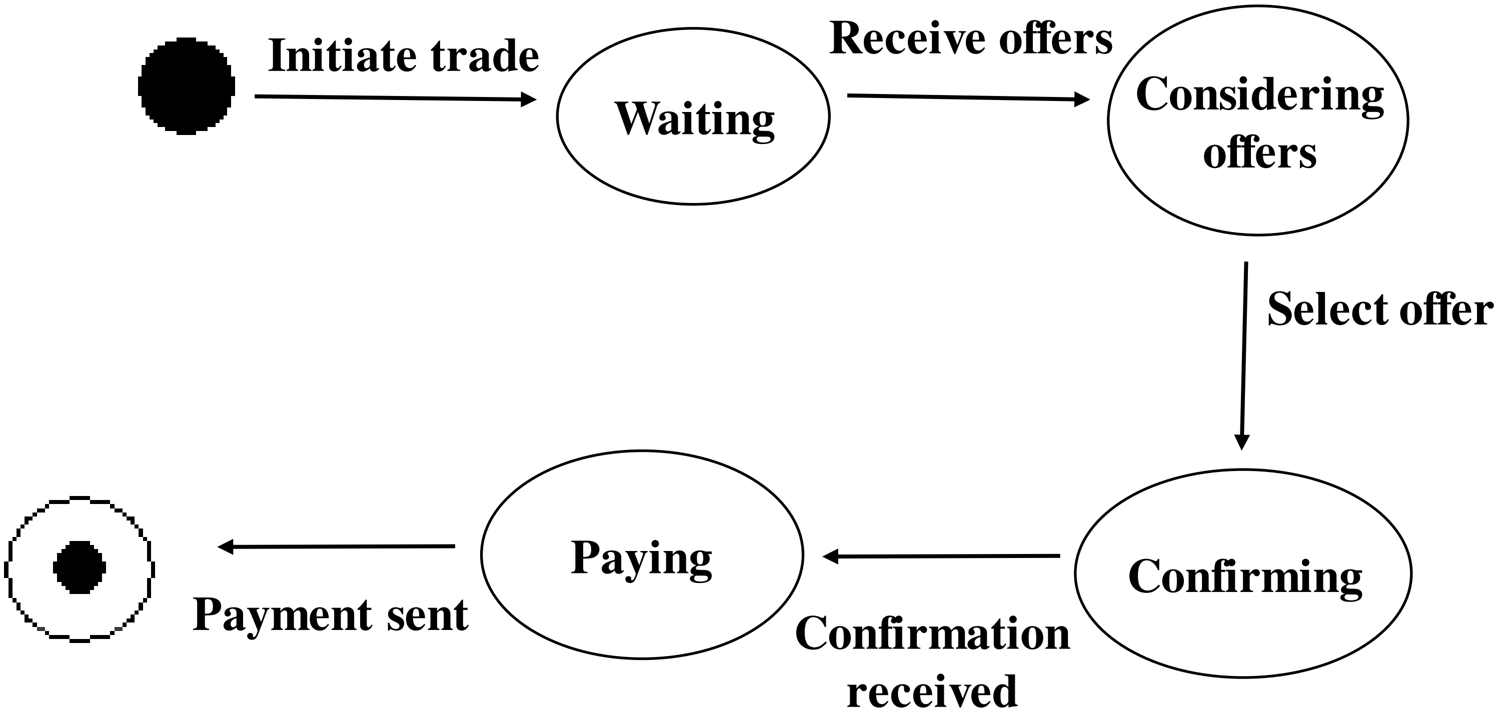
Example 1: State Transition Diagram for the "Student" object

A statechart diagram showing how a student progresses from a potential student to a graduated student.

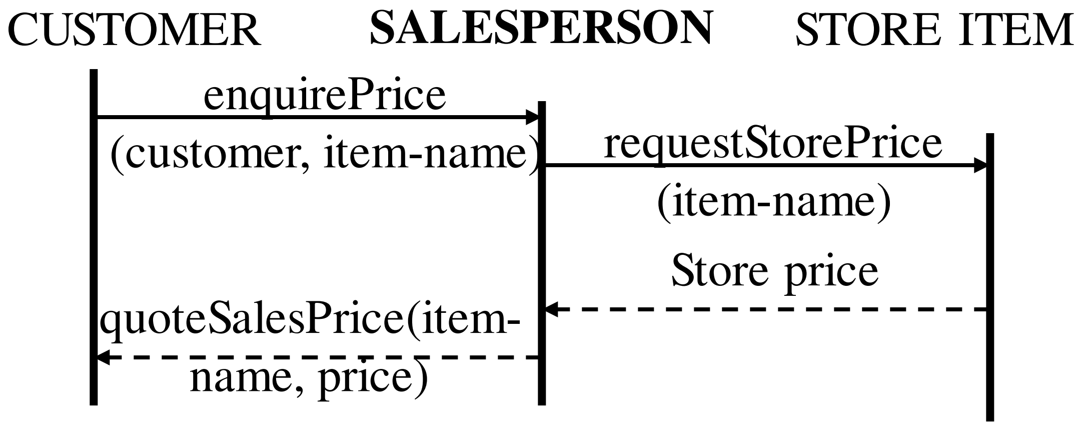


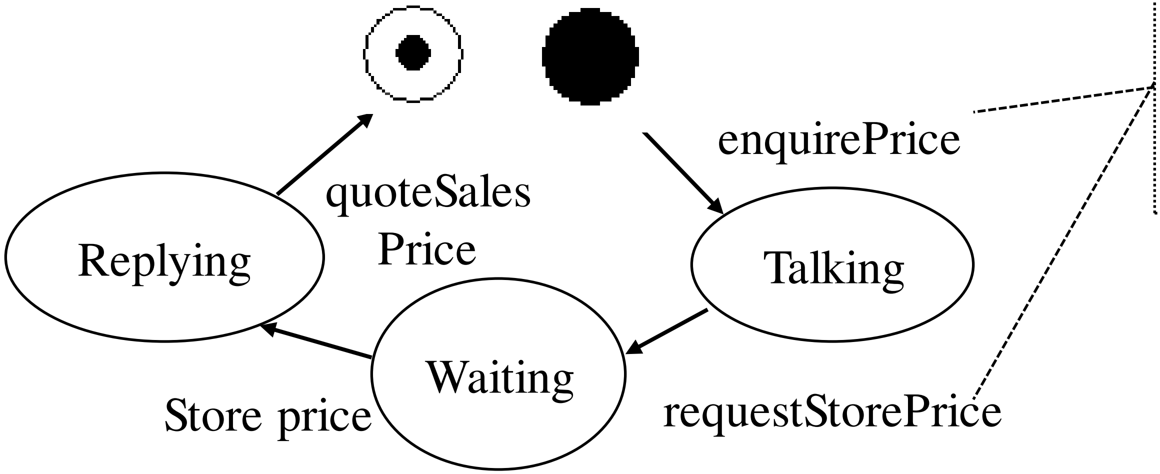
- The above example lets us understand the different states of a "Student" object and the events that trigger/cause those states of a Student to change.

Example2: State Transition Diagram for the “Buyer” object



Example3: State Transition Diagram for the“**SalesPerson**” object

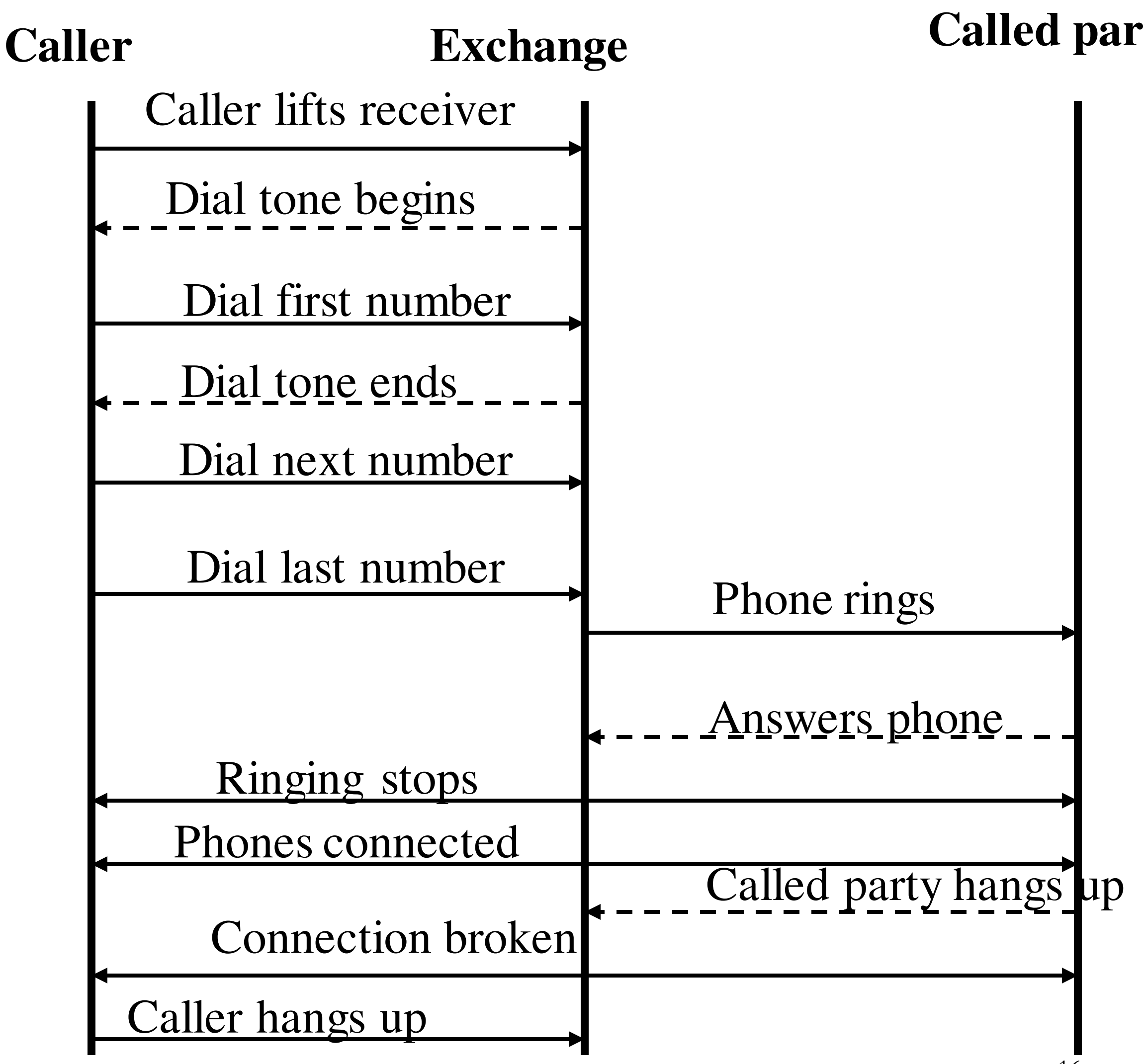




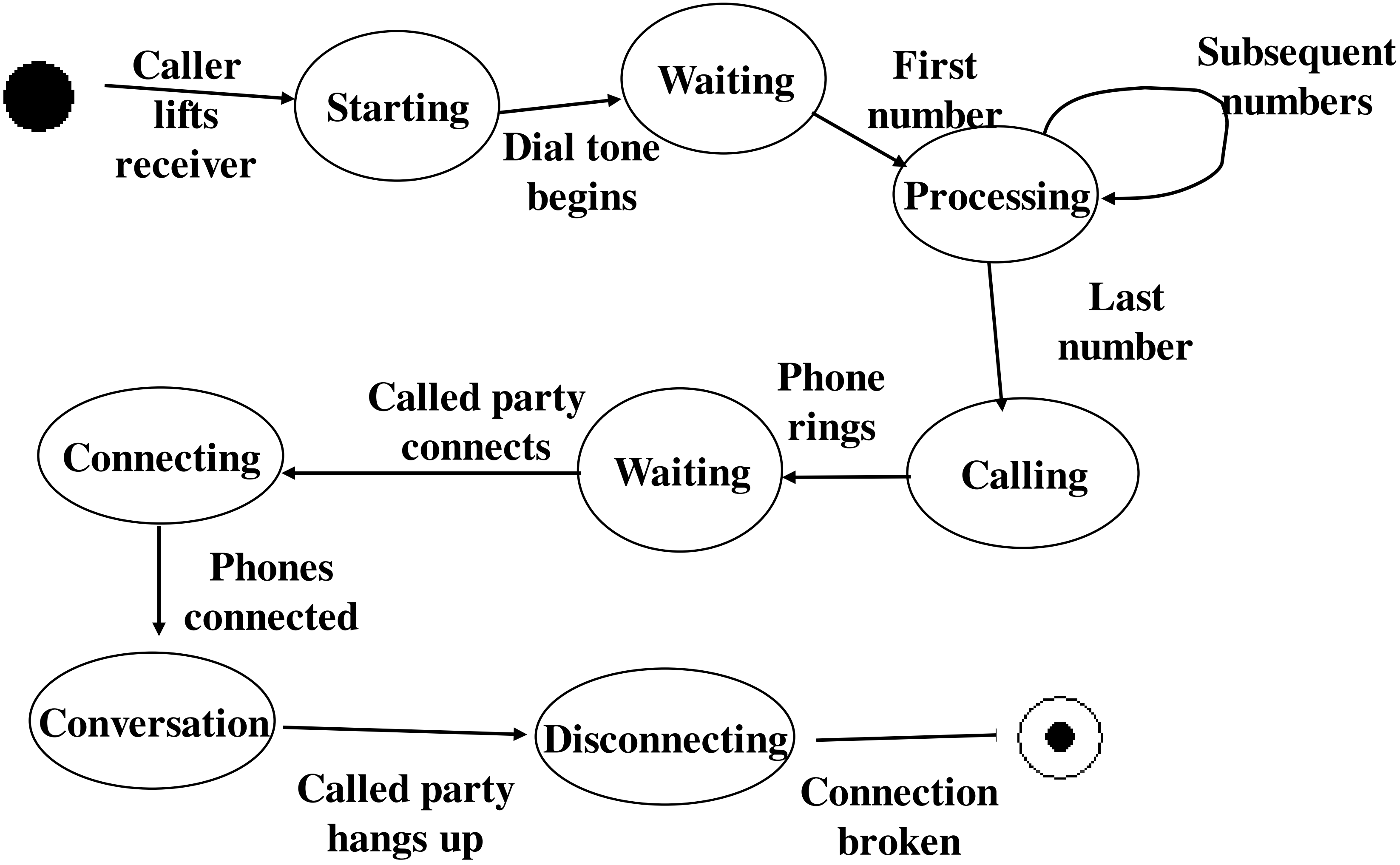
Methods are shown that change state of the SalesPerson

- The customer requests the price and availability of the item from the Sales Person.
- The Sales Person calls the store requesting a store-price.
- The store provides the store price of the item.
- Sales Person computes a commission and quotes a sales price to the customer.

Sequence Diagram For a Telephone Call

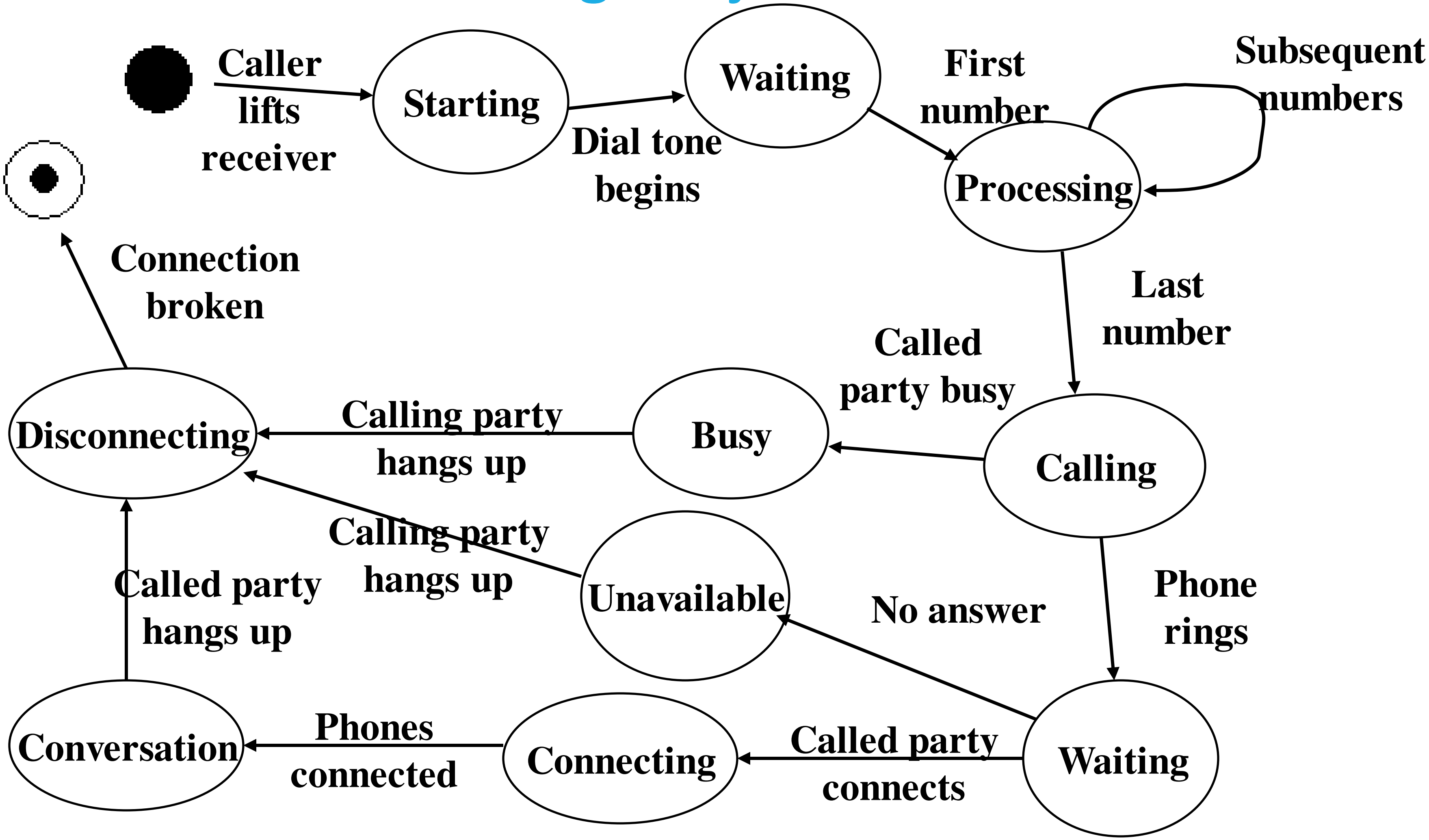


Example4: State Transition Diagram for “**Exchange**” **object**



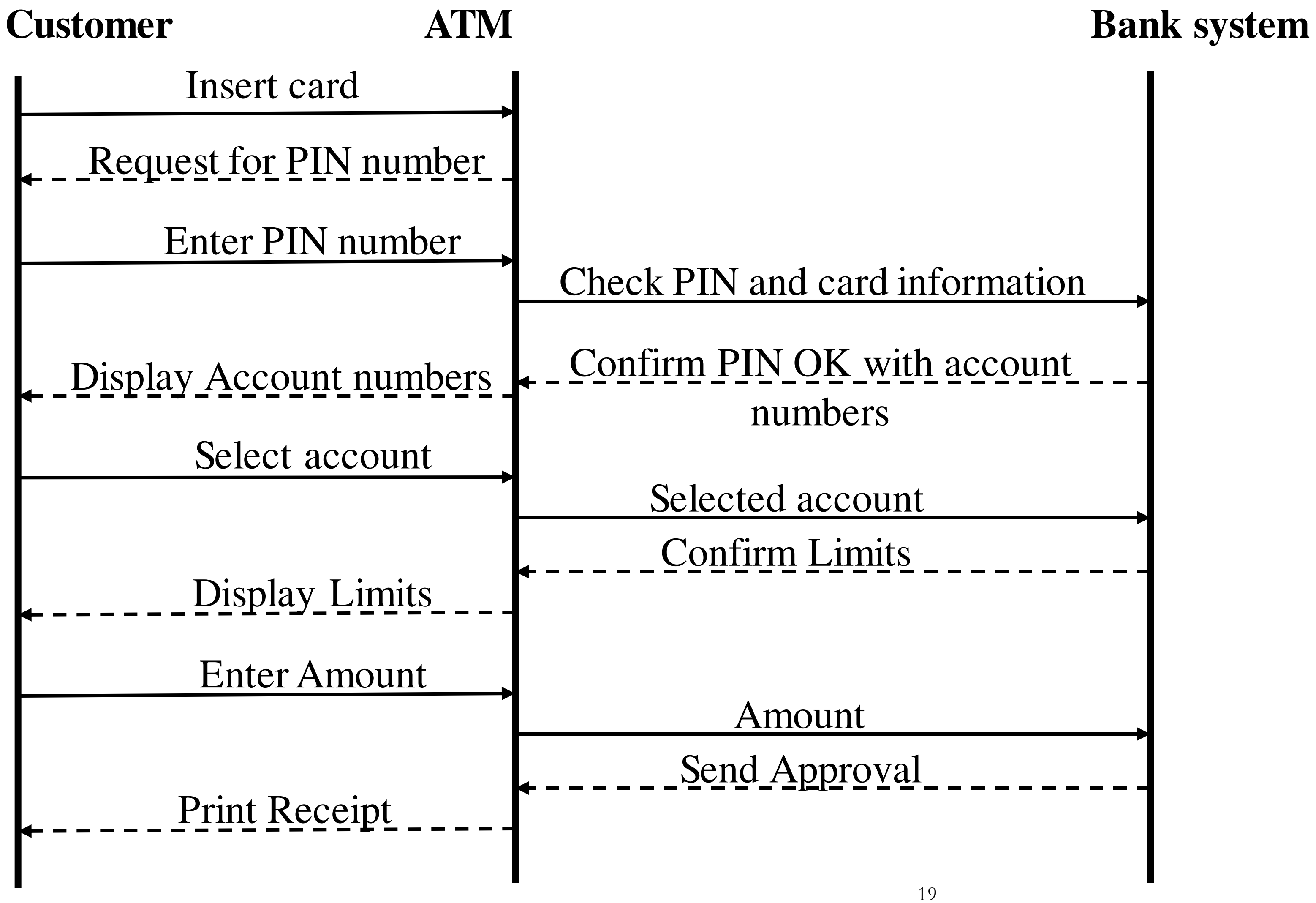
- The above example lets us understand the different states of an Exchange object and the events that trigger/cause those states of an Exchange object to change.

State Transition Diagram including alternate scenario for Exchange object



Sequence Diagram for ATM Machine

This slide to be read at home by students



Customer

ATM

Bank system

Insert card

Request for PIN number

Enter PIN number

Check PIN and card information

Confirm PIN OK with account numbers

Display Account numbers

Select account

Selected account

Confirm Limits

Display Limits

Enter Amount

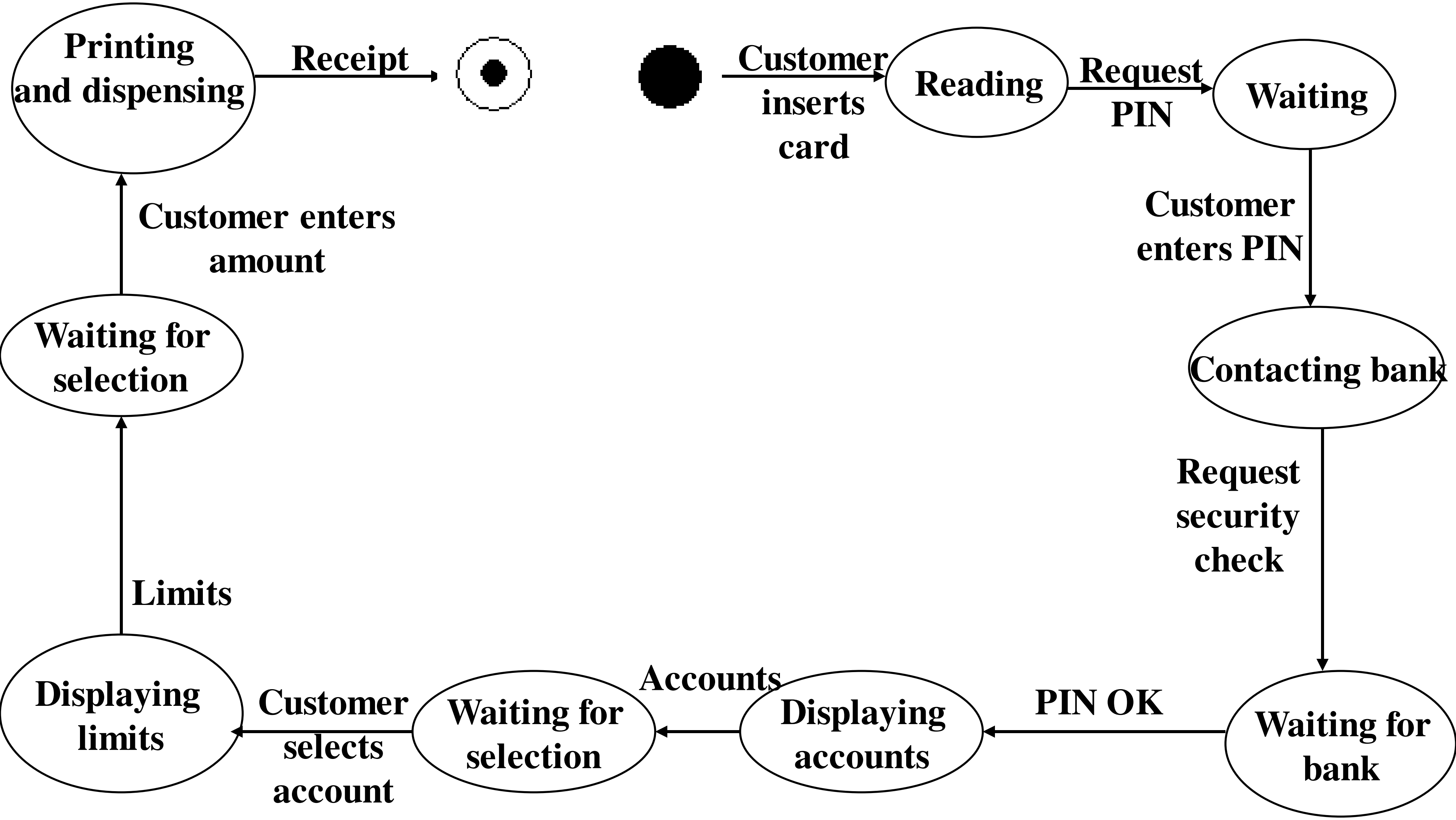
Amount

Send Approval

Print Receipt

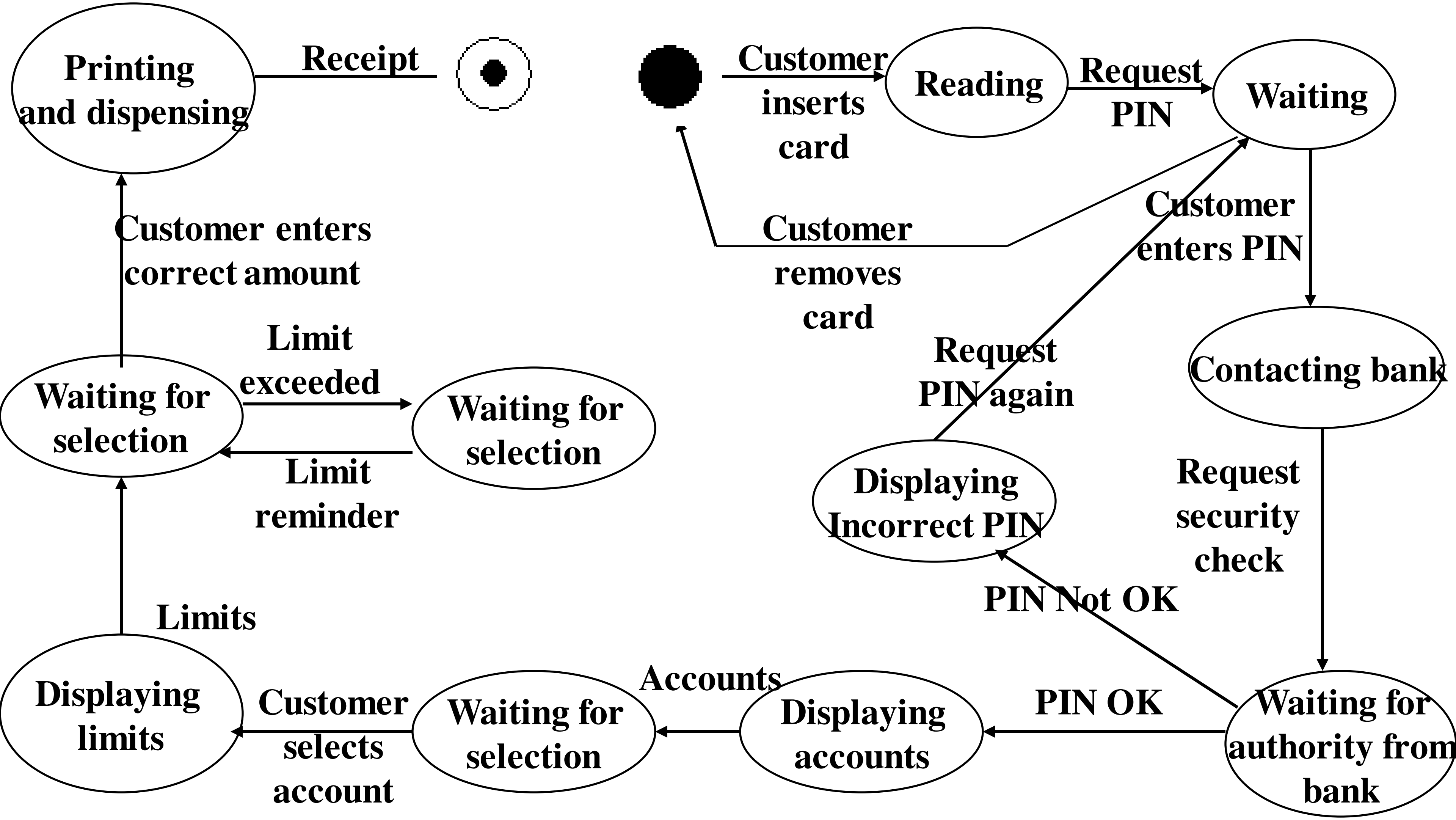
Example5: State Transition Diagram for **ATM machine** Object

This slide to be read at home by students

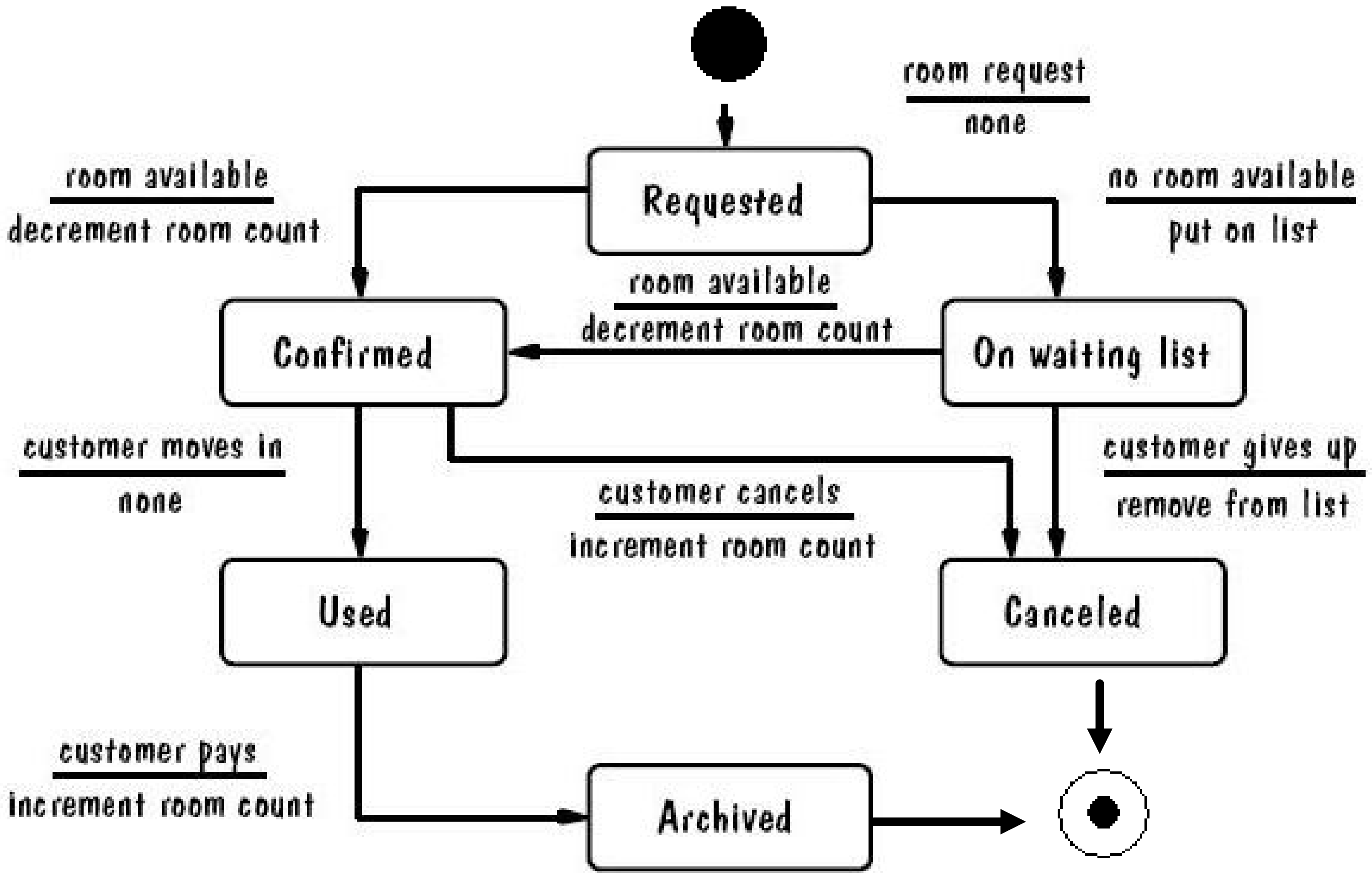


State Transition for ATM machine with alternatives

This slide to be read at home by students



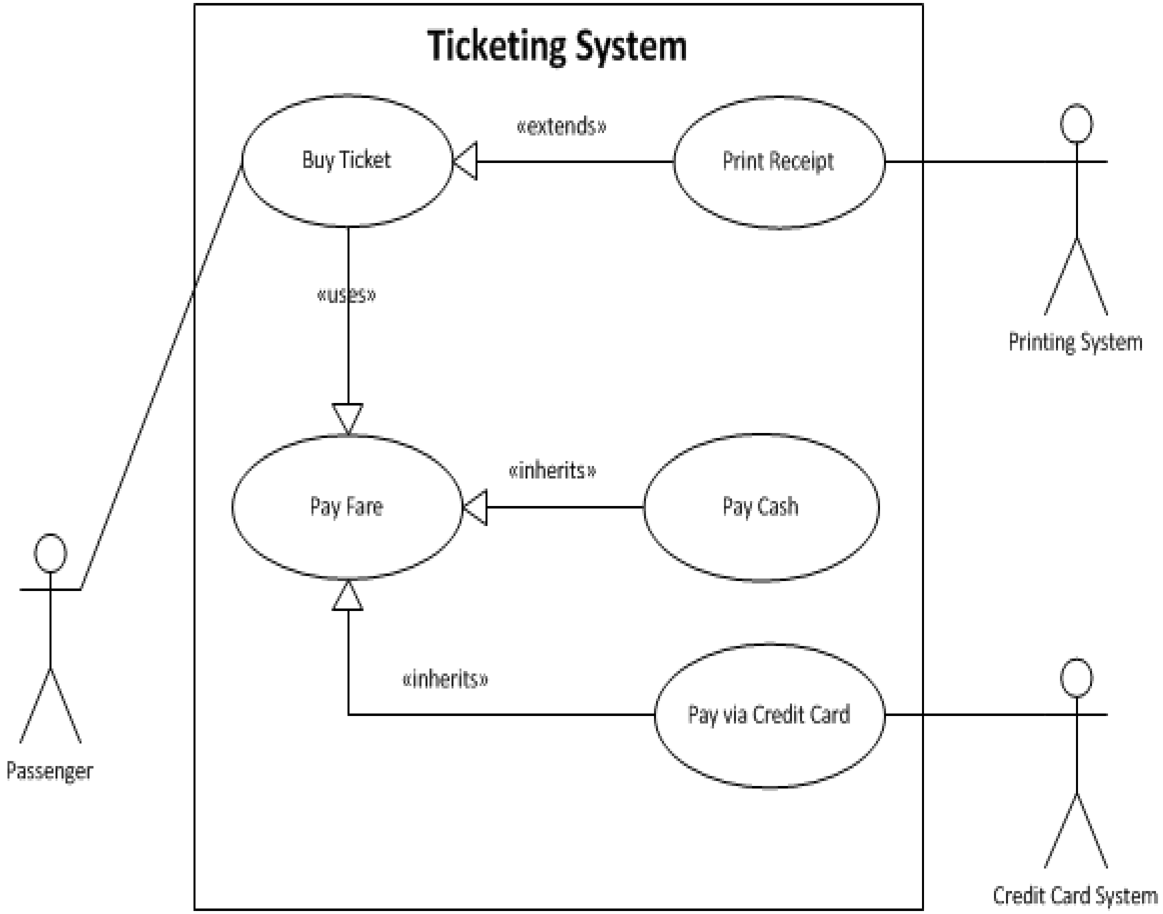
Example6: State Transition Diagram for "Reservation" object (for hotel room reservation)



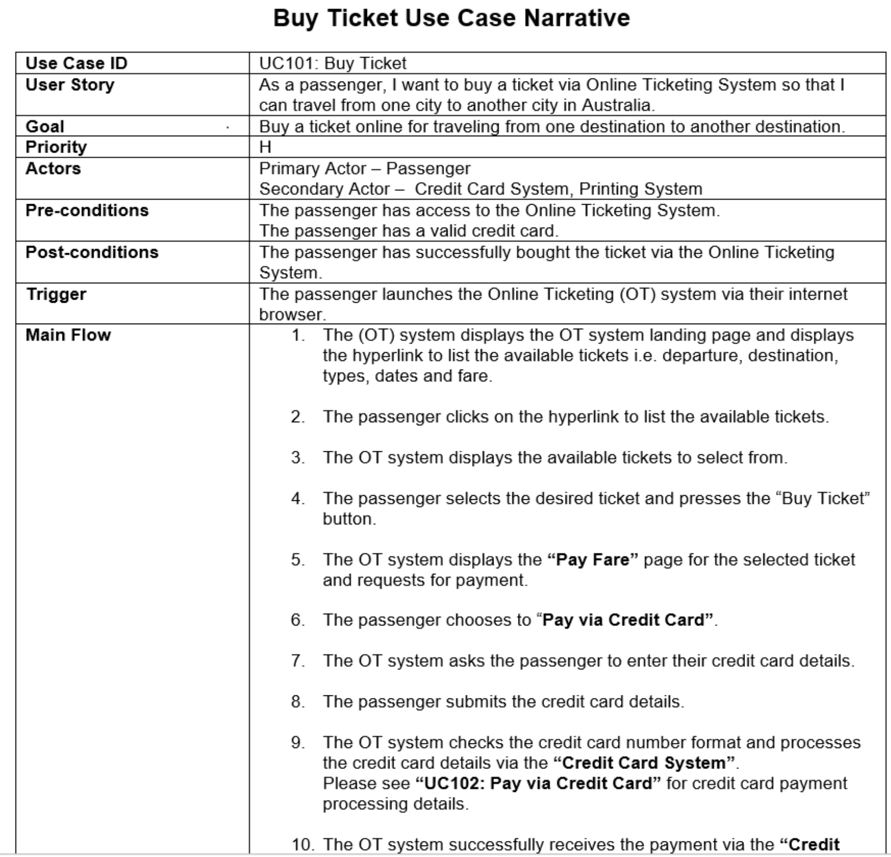
State Transition Diagram

- Ticketing System Example

Use Case Diagram for Ticketing System

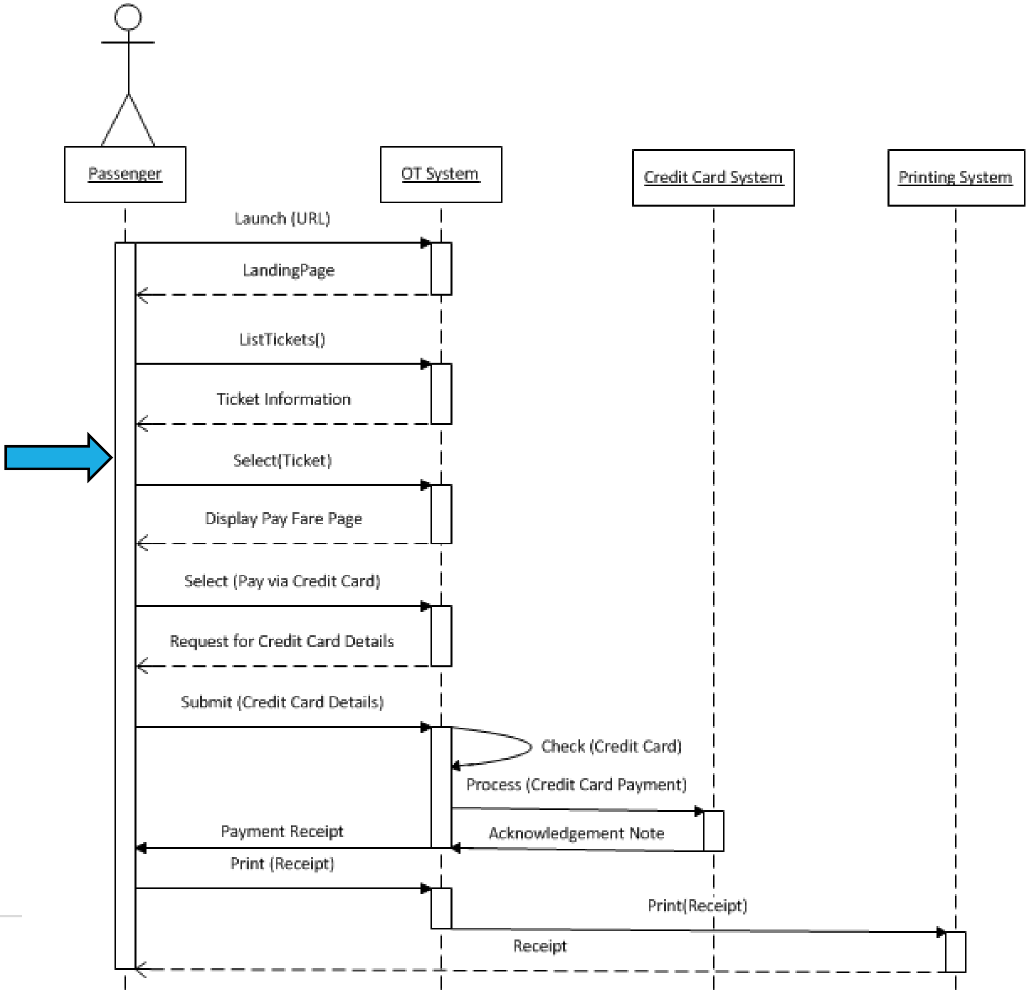


System Sequence Diagram for "Buy Ticket"



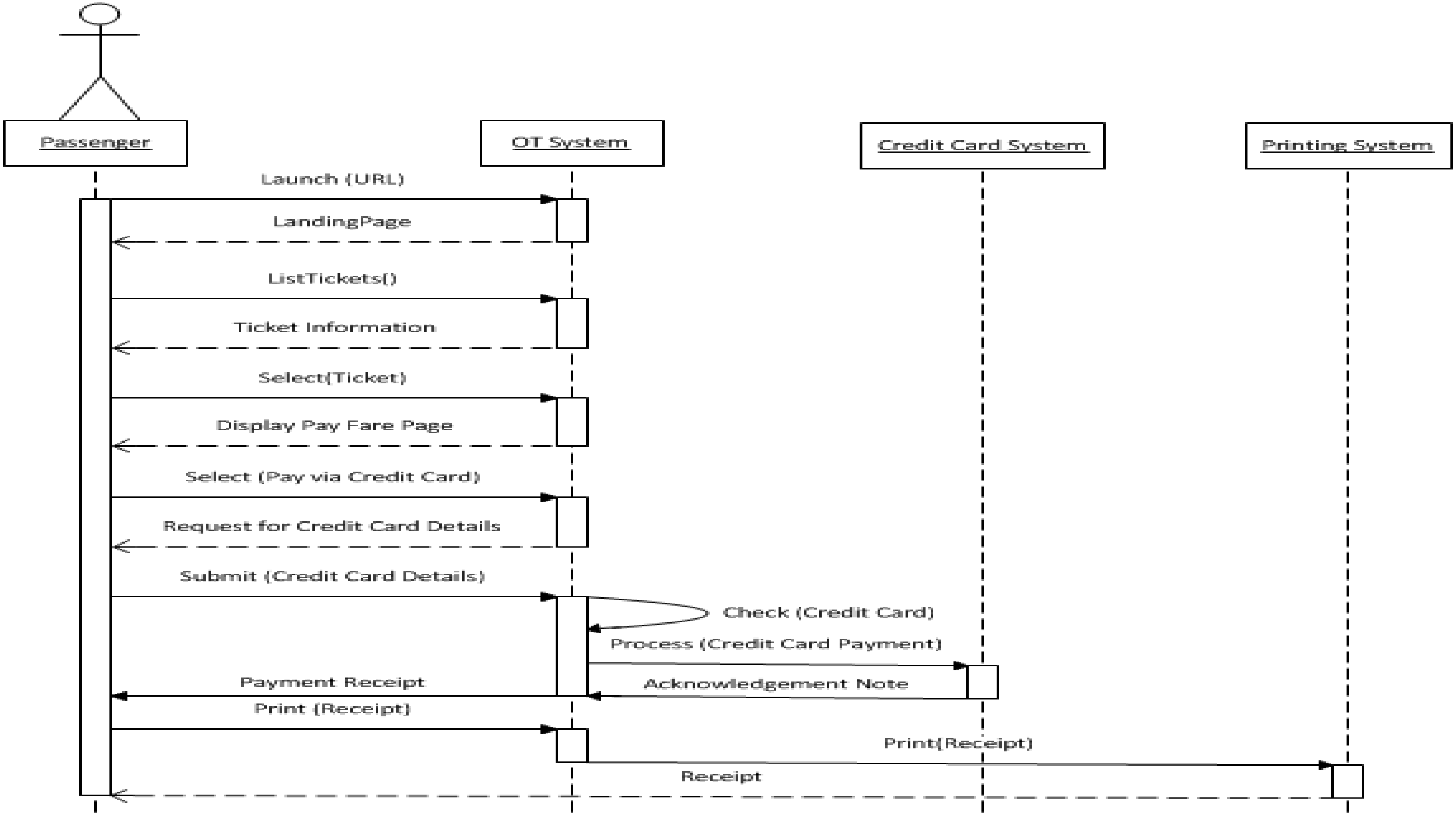
Buy Ticket Use Case Narrative

Use Case ID	UC101: Buy Ticket
User Story	As a passenger, I want to buy a ticket via Online Ticketing System so that I can travel from one city to another city in Australia.
Goal	Buy a ticket online for traveling from one destination to another destination.
Priority	H
Actors	Primary Actor – Passenger Secondary Actor – Credit Card System, Printing System
Pre-conditions	The passenger has access to the Online Ticketing System. The passenger has a valid credit card.
Post-conditions	The passenger has successfully bought the ticket via the Online Ticketing System.
Trigger	The passenger launches the Online Ticketing (OT) system via their internet browser.
Main Flow	1. The (OT) system displays the OT system landing page and displays the hyperlink to list the available tickets i.e. departure, destination, types, dates and fare. 2. The passenger clicks on the hyperlink to list the available tickets. 3. The OT system displays the available tickets to select from. 4. The passenger selects the desired ticket and presses the "Buy Ticket" button. 5. The OT system displays the **"Pay Fare"** page for the selected ticket and requests for payment. 6. The passenger chooses to "**Pay via Credit Card**". 7. The OT system asks the passenger to enter their credit card details. 8. The passenger submits the credit card details. 9. The OT system checks the credit card number format and processes the credit card details via the **"Credit Card System"**. Please see **"UC102: Pay via Credit Card"** for credit card payment processing details. 10. The OT system successfully receives the payment via the **"Credit**

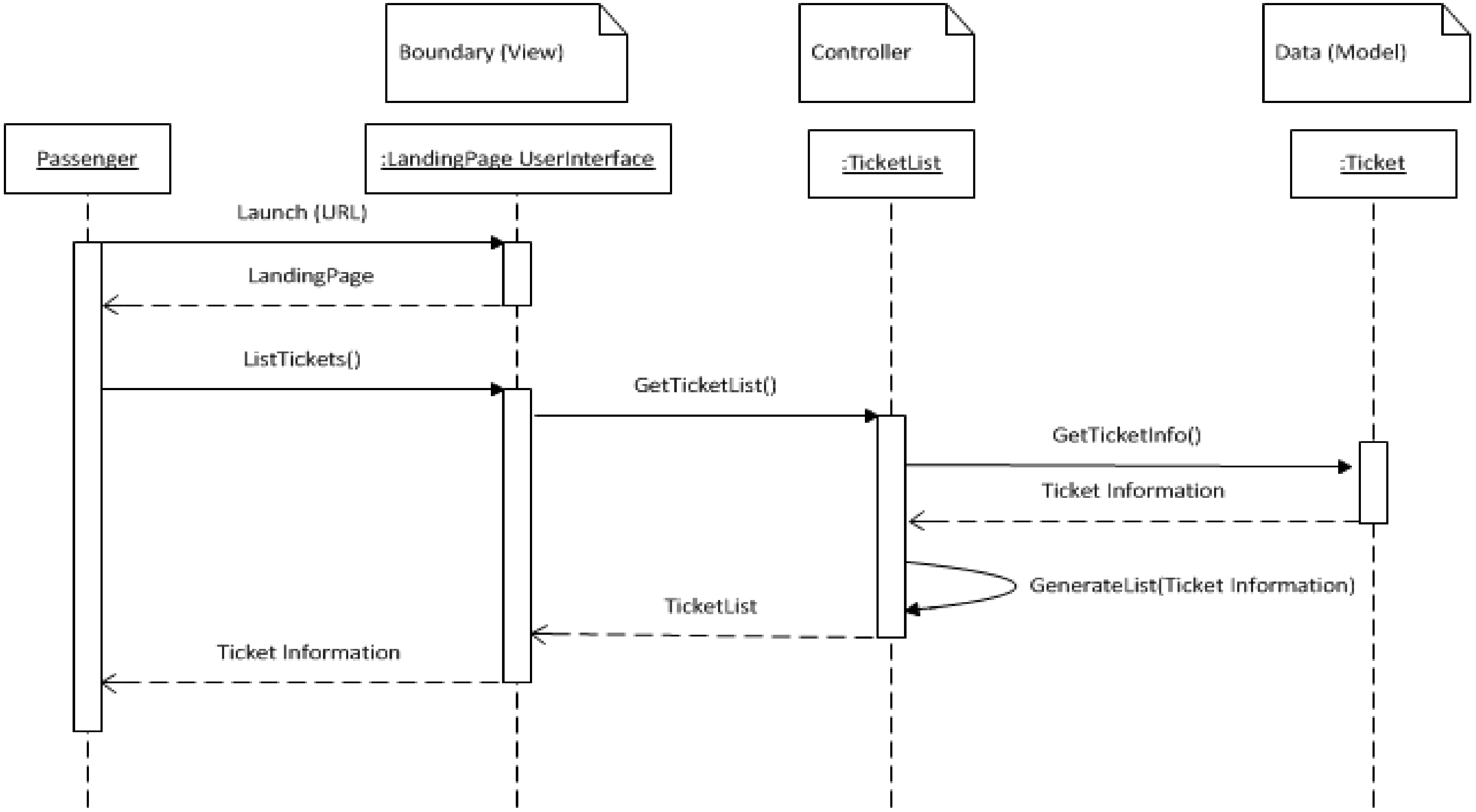


System Sequence Diagram shows interaction between Actor and System (OT System) as a black box.

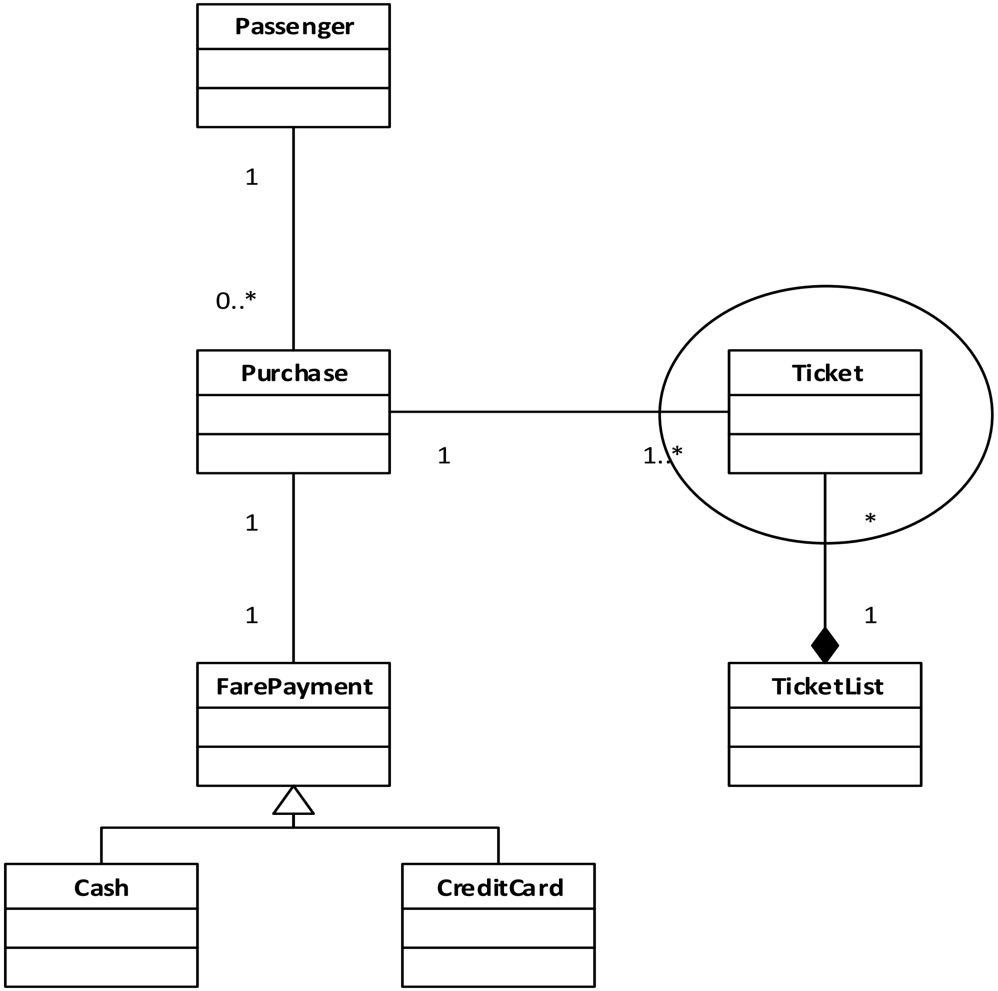
System Sequence Diagram for Ticketing System



Object Sequence Diagram for Ticketing System



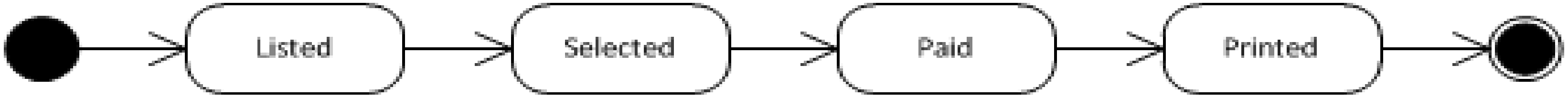
Class Diagram for Ticketing System



State Transition Diagram for “**Ticket**” object

- Using the Use Case Narrative/s and Sequence Diagrams, you can identify the several States of the “Ticket” object and the Events that cause this transition/changes to these states of the Ticket object.

  Below is a draft:



What is missing in this diagram?

Summary

- States (different states of an Object)
- Event (events that cause the transition/change in states)
- State Transition Diagrams (one for each Object/Class), so you can/may draw a separate diagram for each object/class in your class diagram to better understand the several States that an object goes through.

Assignment 2

- State Transition Diagram (one for **“Customer”** object)
- User Interface Requirements (refer ATM example provided in Task 5 of Week 3 Workshop)
- Security Requirements
- Performance Requirements
- Use Assignment 2 Template provided to you
- Refer FAQ – available soon
- ***Assignment 2 is due next week on 27 May at 10:00 PM***

Conclusion

- This Week's Workshop
  - Quiz 8 – Interaction Modelling (3 marks)
  - Tasks – State and Event Modelling

- Next Week's Lecture
  - Agile System Development
- Next Week's Workshop
  - Quiz 9 – State and Event Modelling (3 marks)
  - Tasks – Agile System Development