Ticket Vending Machine

This is a two person project : by khushal amir khan ID: 13721

and shahabud din ID : 13791

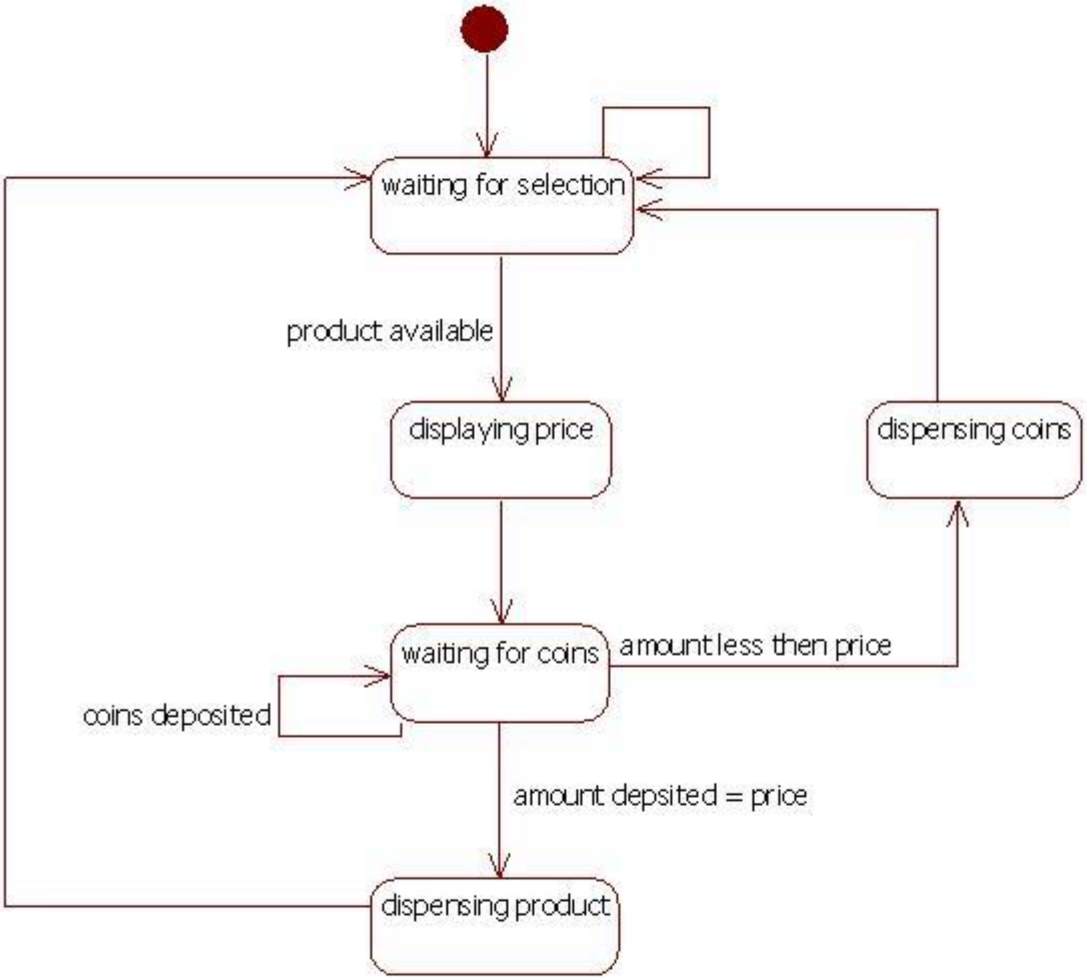
***ClassDiagram** :*Model with a class diagram the following System: Vending Machine.

A vending machine sells small, packaged, ready to eat items (chocolate bars, cookies, candies, etc.). Each item has a price and a name. A customer can buy an item, using a smart card (issued by the vending machine company) to pay for it. No other payment forms (i.e. cash, credit card) are allowed. The smart card records on it the amount of money available. The functions supported by the system are:

Sell an item (choose from a list of items, pay item, distribute item) Recharge the machine

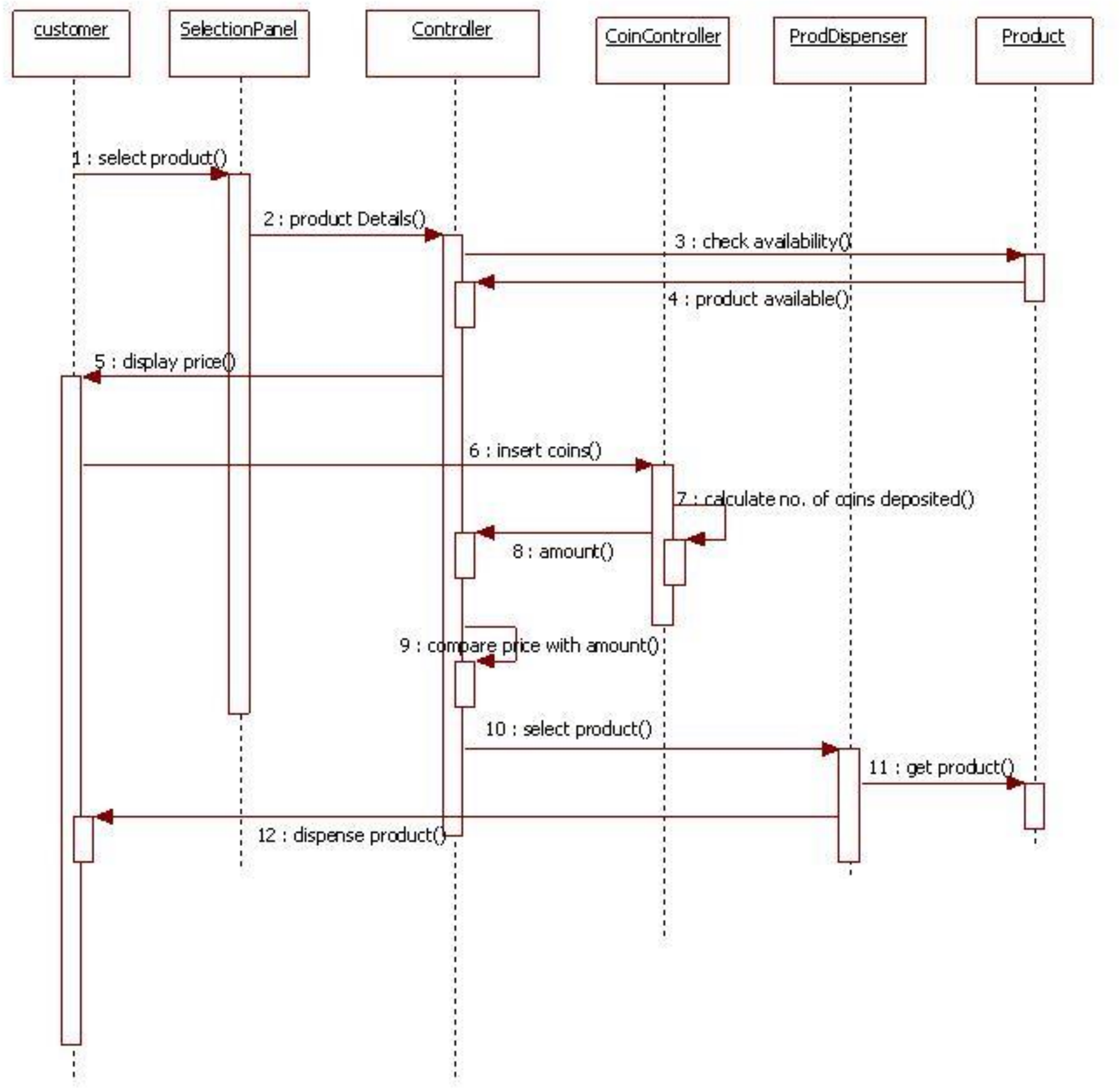
Set up the machine (define items sold and price of items)

Monitor the machine (number of items sold, number of items sold per type, total revenue).The system can be used by a customer, a maintenance employee (who recharges items in the machines), an administrator (who sets up the machine).



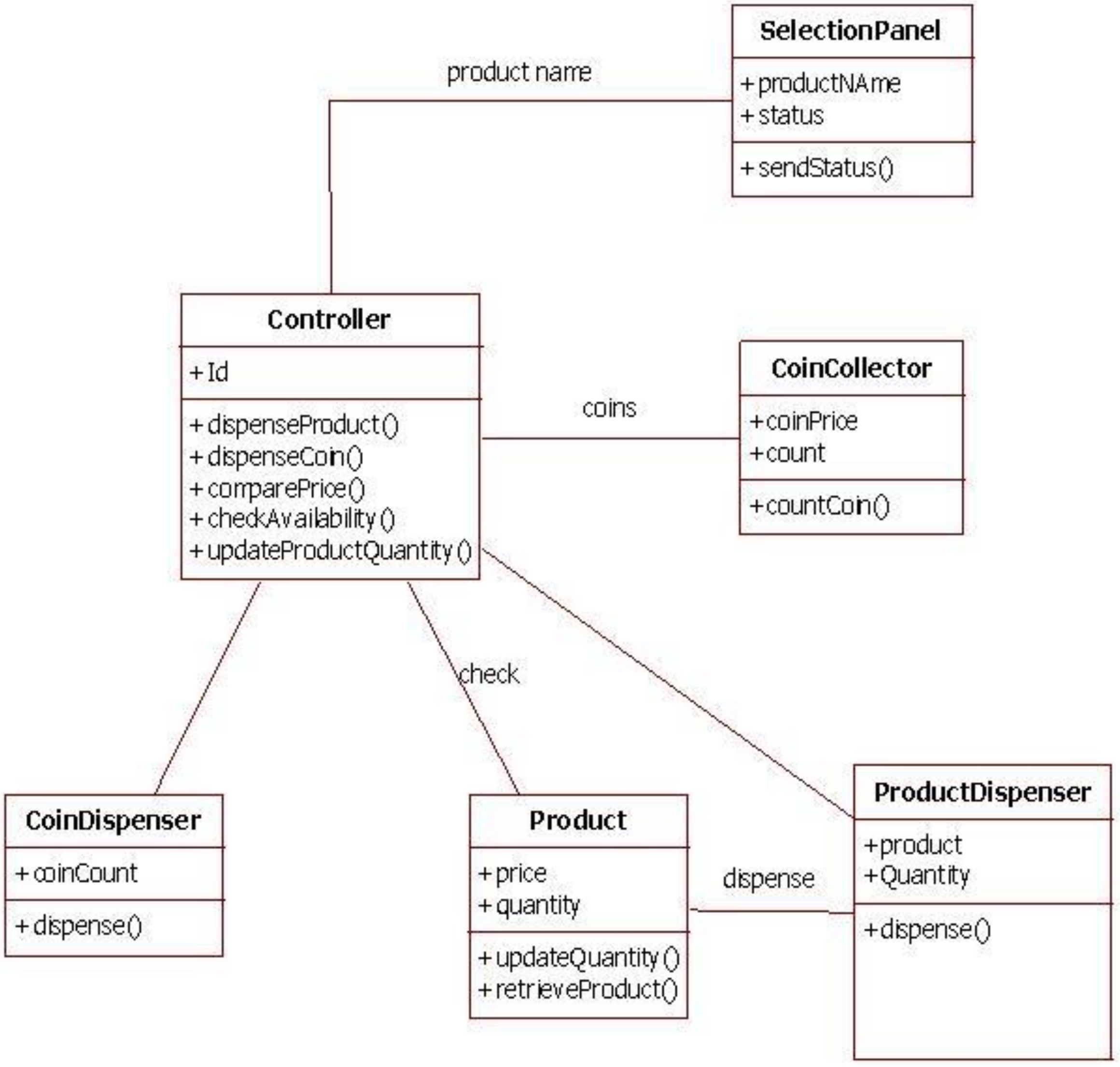
Ticket Vending Machine

Sequences Diagram *:*This Sequence Diagram example shows a sequence diagram for the buy tickets use case. This use case is initiated by the customer at the ticket vending machine communicating with the box office. The steps for the make charges use case are included within the sequence, which involves communication with both the ticket vending machine and the credit card service. This sequence diagram is at an early stage of development and does not show the full details of the user interface. For example, the exact form of the seat list and the mechanism of specifying seats must still be determined, but the essential communication of the interaction has been specified by the use case.



Ticket Vending Machine

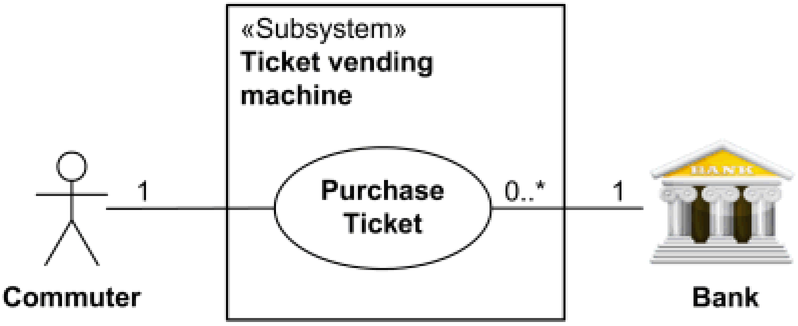
State Diagram *:* State Model, which represents the temporal, behavioral and control aspects of the system. A state machinediagram specifies the possible states that a model element may assume, the transitions allowed at each state, the events that can cause transitions to occur and the actions that may occur in response to events. Events, states and transitions are the basic components of a state machine diagram. States of an object are essentially determined by the values that certain variables (attributes) of the object may assume. Conceptually, an object continues to remain in a state, until an event causes it to transit to another state. An event is any noteworthy occurrence. An event occurrence may be of some consequence to the system. However, the same event can have different effects (or may even have no effect) in different states. A transition is a relationship between two states indicating a possible change from one state to another. Figure 1 shows the state machine diagram depicting the behaviour of the objects in a toy train machine. A state in a state machine diagram can either be simple or composite type. A simple state does not have any sub-states.



Ticket Vending Machine

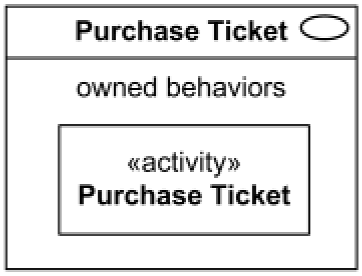
Use Case Diagram : Ticket vending machine, i.e. vending machine that sells and produces tickets to commuters, is a subject of the example use case diagram. This kind of a machine is a combination of both hardware and software, and it is only a part of the whole system selling tickets to the customers. So we will use «Subsystem» stereotype.

Ticket vending machine allows commuters to buy tickets. So Commuter is our primary actor.



The ultimate goal of the Commuter in relation to our ticket vending machine is to buy a ticket. So we have Purchase Ticket use case. Purchasing ticket might involve a bank, if payment is to be made using a debit or credit card. So we are also adding another actor - Bank. Both actors participating in the use case are connected to the use case by association.

Use case behaviors may be described in a natural language text (opaque behavior), which is current common practice, or by using UML behavior diagrams. UML tools should allow binding behaviors to the described use cases. Example of such binding of the Purchase Ticket use case to the behavior represented by activity is shown below using UML 2.5 notation.



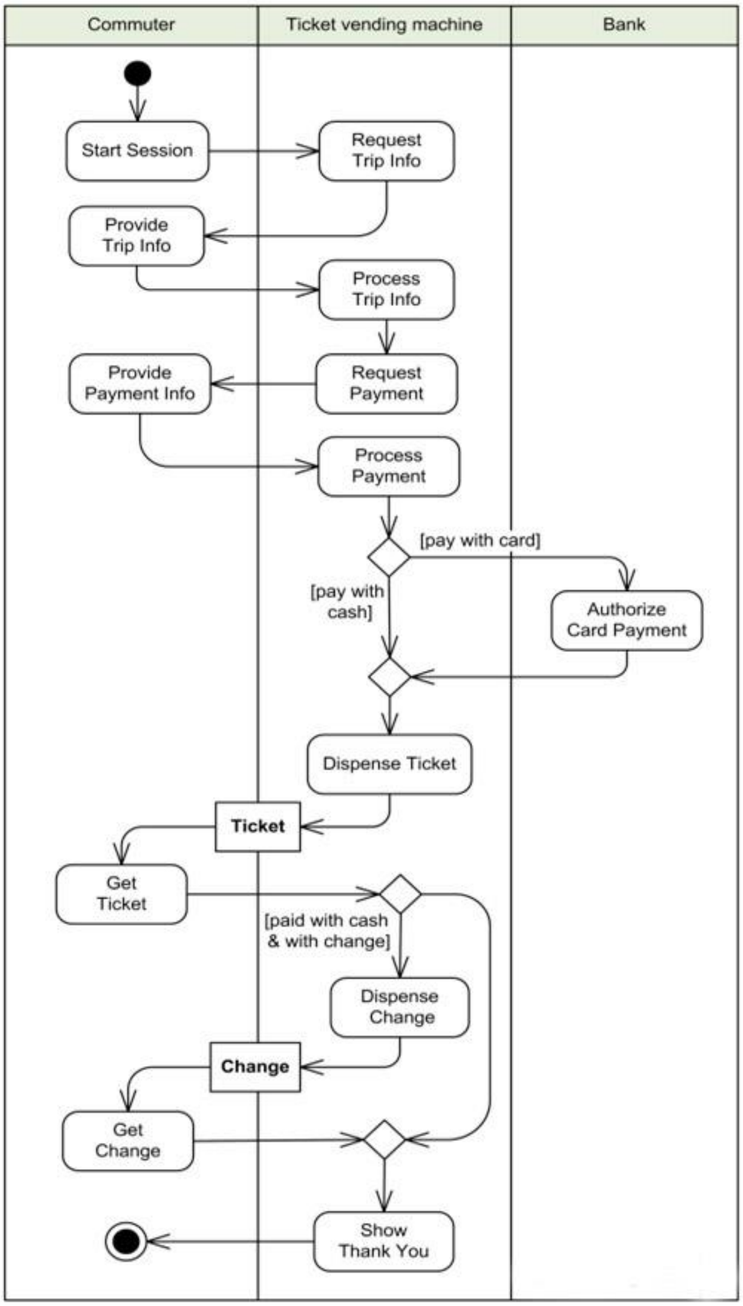
Ticket Vending Machine

Activity Diagram :

This is an example of UML activity diagram describing behavior of the Purchase Ticket use case.

Activity is started by Commuter actor who needs to buy a ticket. Ticket vending machine will request trip information from Commuter. This information will include number and type of tickets, e.g. whether it is a monthly pass, one way or round ticket, route number, destination or zone number, etc. Based on the provided trip info ticket vending machine will calculate payment due and request payment options. Those options include payment by cash, or by credit or debit card. If payment by card was selected by Commuter, another actor, Bank will participate in the activity by authorizing the payment.

After payment is complete, ticket is dispensed to the Commuter. Cash payment might result in some change due, so the change is dispensed to the Commuter in this case. Ticket vending machine will show some "Thank You" screen at the end of the activity.


Commuter
Ticket vending machine
Bank
Start Session
Request Trip Info
Provide Trip Info
Process Trip Info
Request Payment
Provide Payment Info
Process Payment
[pay with card]
[pay with cash]
Authorize Card Payment
Dispense Ticket
Ticket
Get Ticket
[paid with cash & with change]
Dispense Change
Change
Get Change
Show Thank You