AUTOMATIC TICKETING MACHINE

DELIVERABLE 4

by

AKINRINADE, Oyeyemi Akintoyese (ID: 27017766)
MAKKAR, Gagandeep Kaur (ID: 26613748)
Gagandeep Kaur (ID: 27683731)
BANIK, Biswajit (ID: 26831419)
LIN, Xuefei (ID: 27312377)

A
submission in partial fulfillment
of the requirements of
COEN 6312

March 21, 2016

Table of Contents

1.0 An Automatic Ticket Machine (ATM) State Machine Diagram……………………………….. 2
1.1 About UML State Machine Diagram 2
1.2 Corresponding Classes (for UML State Machine Diagram) 2
1.2.1 Customer class (State Machine Diagram):.. 2
1.2.2 Automatic Ticket Machine class (State Machine Diagram): 3
1.2.3 Bank Class (State Machine Diagram):.. 5
2.0 Action Specifications (for corresponding State Machine Diagrams) 6
2.1 Action specification for the corresponding state machine scenarios………………………. 5
2.1.1 Action specification for Customer class... 6
2.1.2 Action specification Automatic Ticket Machine class.. 6
2.1.3 Action specification Bank Class 7
3.0 Tool Support 7
3.1 Eclipse and Papyrus (For State Machine Diagram).. 7
References.. 8

1.0 An Automatic Ticket Machine (ATM) State Machine Diagram

This document describes state machine diagrams for three (3) classes in partial fulfillment of our team project [1] for COEN 6312. To begin with, a state machine diagram (also known as State chart) is outlined as a machine that defines completely different states of an object and these states are controlled by external or internal events. They outline completely different states of an object throughout its lifetime and these states are modified by events. Therefore State chart diagrams are helpful to model reactive systems. A reactive system is outlined as a system that responds to external or internal events. The purpose of a state machine diagram is to model dynamic facet of a system and additionally to model the life time of a reactive system. It will describe completely the different states of an object throughout its life time and it outlines a state machine to model the states of an object. (We will use UML to describe state machine diagram)

1.1 About UML State Machine Diagram

We are using UML to describe the state machine behavior of 3 classes (described below) specified in the previous milestone. All the terminology follows UML 2.0 standards to describe state, transitions and related descriptions.

1.2 Corresponding Classes (for UML State Machine Diagram)

We have chosen to describe the state machine diagrams for the following classes.

- Customer class
- ATM class
- Bank Class

1.2.1 Customer class (State Machine Diagram)

(This class covers operations like place order, make payment and pass recharge. Also contains document related operations like collect tickets/card)

Using some of the functionalities of customer class, we have presented below the state machine diagram for the customer class which signifies the corresponding states for customer class; there is no terminate state, only the initial state as can be seen below.

- Enter_order_details state
- Insert_card state
- Idle state
- Make_payment state
- Remove_card state

The roles of the states are specified in the action specification.

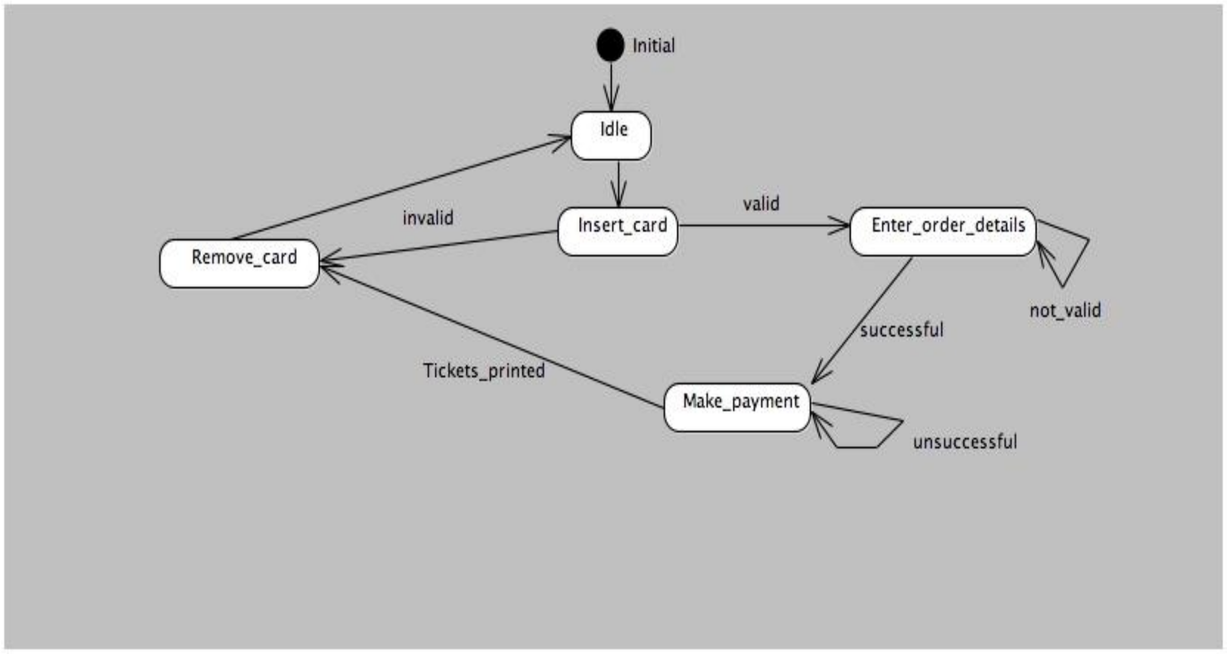

Figure 1: Customer Class SMD (State Machine Diagram)

1.2.2 Automatic Ticket Machine class (State Machine Diagram)

The following state machine diagram presents the corresponding states like idle state, when customer inserts the pass (card), it gets into an active state while validating the pass. If the pass is found valid, it will prompt the customer to enter order details and if the order details are valid, that order will get processed. The processing of order undergoes the process of making payment. Once the payment is successfully done, the ticket(s) and/or the receipt is printed for the customer. The system reaches the “final” state. The various states in this state machine diagram are:

- Idle state
- Final state
- Validate_card state
- Make_payment state
- Prompt_to_remove_card state
- Print_tickets_and_receipt state
- Process_order state
- Accept_order_data state

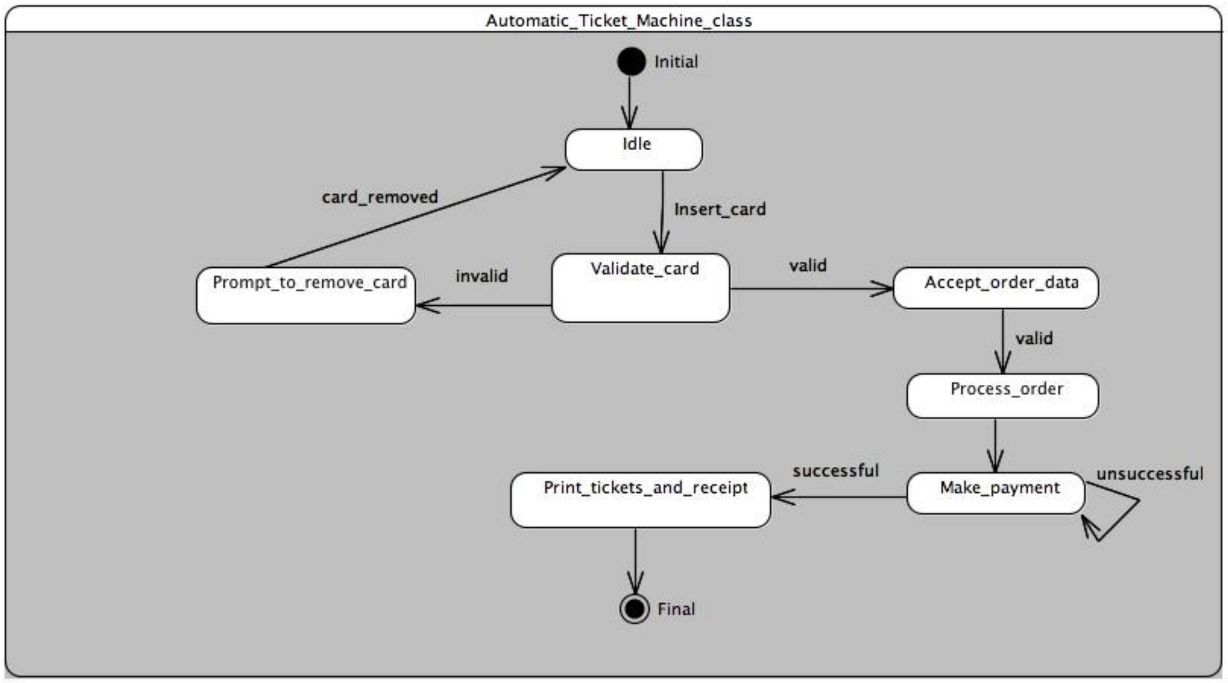

Figure 2: Automatic Ticket Machine Class SMD

1.2.3 Bank Class (State Machine Diagram)

The function of Bank class is to access the account of the customer and process the payment for the requested order. The payment request is checked for its validity and if valid, the payment is made otherwise, it goes to "cancel" state and the request is cancelled, advancing the system to the "final" state. The various states in this bank class are:

- Idle state
- Final state
- Payment_request state
- Validate_request state
- Process_payment state
- Cancel state

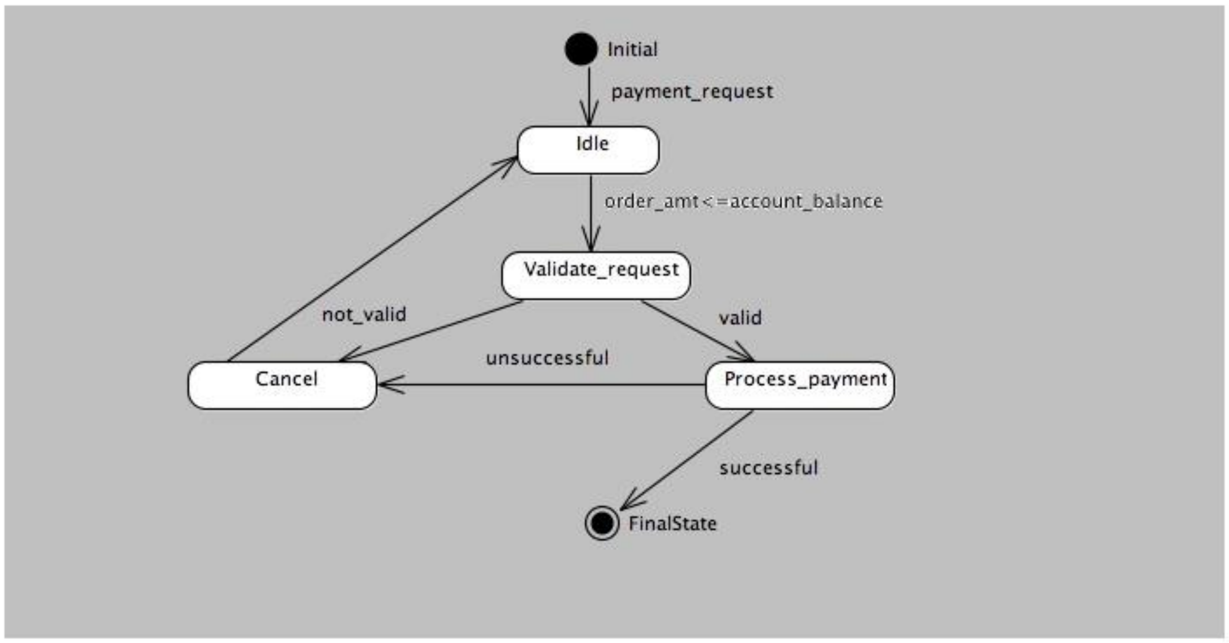

Figure 3: Bank Class SMD

2.0 Action Specifications (for corresponding State Machine Diagrams)

When an incident instance is sent, the state machine responds by performing actions, like changing a variable, performing I/O operations, invoking a function, generating another event instance, or changing to a different state. Any parameter values associated with the current event are available to any or all actions directly caused by that event.

2.1 Action Specification for the Corresponding State Machine Scenarios

2.1.1 Action Specification for Customer class

- Enter_order_details (count_of_tickets, ticket_type)

After, validating the pass card, and getting the order data from customer, which is count of tickets and the type of tickets (one way/two way), the order is finalized by calculating the price to be charged to customer. If order details are valid, the payment is processed.

- Make_payment (order_amount)

Once, the price is calculated and system reaches the "make payment" state, the customer is charged the amount of his/her order. Finally, after successfully making the payment, tickets are printed, or appropriate credits are applied to the pass.

2.1.2 Action Specification Automatic Ticket Machine class

- Validate_card (customer, w)

Validate_card will check whether or not the given person w is a registered member or not. It also checks for the validity of the pass. If customer is registered, he/she will be charged special rates. Only if the pass is valid will appropriate credits be applied to it. If not valid, customer will be asked to remove the card and system goes to "idle" state.

- Accept_order_data (count_of_tickets, type_of_tickets)

ATM class accepts the data from the order and calculates the amount to be charged to the customer for the order based on the number of tickets and the type of tickets.

- Print_tickets_and_receipt (count_of_tickets, w)

ATM class process the order and finally, after successful payment, the count of tickets as per the order is printed, or credits applied to the pass of customer w.

- Process_order (count_of_tickets, amount)

ATM class accepts the customer order of tickets along with their type and starts the processing of order by giving an order number and the amount of that order. It also takes the system date and time for order data.

2.1.3 Action Specification Bank class

- Payment_request (account_no, order_no, amount)

Payment_request is used by bank class to charge amount to the customer's account. This amount is equal to the price of order.

- Process_payment (account_no, amount)

If the account_no is valid and the balance in account is greater than order price, the amount of order is deducted from customer account and is paid towards order payment. If account_no was found invalid, the system advances to the "cancel" state where the payment is cancelled and system enters "idle" state. If payment is made successfully, system advances to the final state.

3.0 Tool Support

3.1 Eclipse and Papyrus (for State Machine Diagram)

We employed Papyrus [2] for eclipse to create the diagrams and describe different state machine diagrams for the classes presented above.

References

[1] https://moodle.concordia.ca/moodle/course/view.php?id=69350.

[2] Papyrus: http://www.eclipse.org/papyrus.