CS3365 – Software Engineering **Fall 2017**

Due: December, 2017

Automated Supermarket Checkout System (AutoScouts)

Your group has been assigned to develop a client/server system for an **Automated Supermarket Checkout System**. Your group has been given the Requirements specification for the **AutoScouts System**, which describes a use case model, consisting of a description of the actors and use cases that fully define the system. Each use case is described in terms of the actors and their interactions with the system.

The system is to be developed as follows:

Requirements Analysis: Meet instructor for consulting – Week of 16th, October

a) Develop the static model for the AutoScouts system, which depicts the classes and their relationships. A class can be classified as a boundary (interface) class, entity class, control class, or application logic class. Describe each class as to what it does or why it is needed. Define the attributes of each entity class and describe each attribute shortly.

b) Develop the interaction model (using communication diagram or sequence diagram) that depicts objects participating in each use case and the sequence of interactions among the objects. A use case is modeled using a communication diagram or sequence diagram, which describes both a successful scenario (common scenario) and alternative scenarios.

c) Identify the operations of classes using the interaction model in (b).

Design and Implementation: Meet instructor for consulting – Week of 27th, November

d) Design the software architectural model for the AutoScouts system where the model is defined in terms of subsystems and their interactions. Each subsystem should be represented with objects supporting the subsystem. Define the communication styles between subsystems.

e) Describe how to implement the software architecture model in (d), which is a client/server architecture. You need to describe what programming languages for client and server, communication between them, database, or any techniques are used.

f) Design database tables for the AutoScouts system. Define the relational database tables for the system. Define the primary key and/or secondary key for each database table, and describe data schema for each database table.

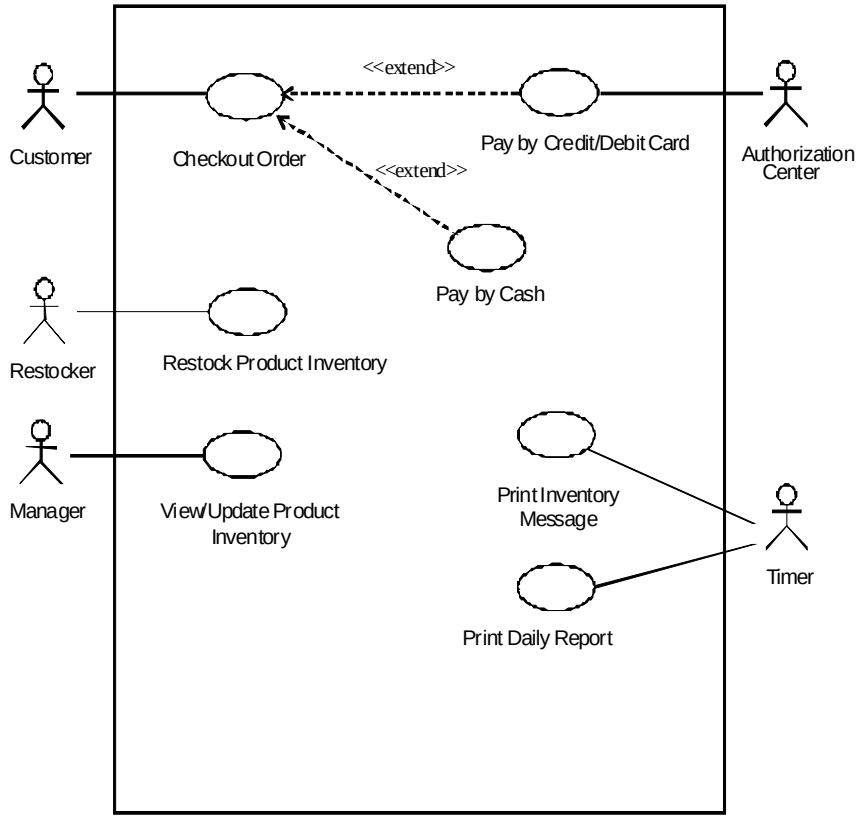
g) Implement the AutoScouts system. Design test cases that consist of test data and expected results for use cases. Test your AutoScouts system using test data for each use case and then record the test results. You are required to turn in your test results with test cases.

State any assumptions you make.

Use Case Model for AutoScouts System

In this section, the use case model of AutoScouts is presented. The following shows the use case diagram of AutoScouts, followed by descriptions of each use case.

Use Case Diagram



Use Cases Descriptions

Use Case Name: Checkout Order
Summary: Customer checks out items.
Actors: Customer
Precondition: Self-Checkout Station is ready for use and displaying welcome message.
Description:
1. Customer presses the Start Checkout key.
2. System displays a prompt "please scan items".
3. Customer scans items.
4. System displays and prints the item description and regular price for each item scanned.
5. Customer presses the TOTAL key when finished scanning items.

6. System displays and prints total amount due.
7. System prompts customer to select a payment method.

Alternatives:

- If customer selects PAY BY CREDIT/DEBIT CARD, extend to Pay by Credit/Debit Card use case (extend after step 7 above).
- If customer selects PAY BY CASH, extend to Pay by Cash use case (extend after step 7 above).
- If the customer presses SUBTOTAL key while scanning items, the total price and tax are computed then displayed and printed.
- If the customer presses CANCEL CHECKOUT key at any time, the entire transaction is canceled and the self-checkout station returns to idle state.
- If the customer presses the CANCEL PAYMENT key, then the current payment method is canceled and the system prompts for a payment method.
- If number of items goes below specified threshold, then send inventory message to supermarket printer and to system message buffer.

Postcondition: Customer has been checked out.

Use Case Name: Pay by Credit/Debit Card
Summary: Customer pays with credit/debit card authorized by Authorization Center.
Dependency: Extension of Checkout Order use case.
Actor: Customer, Authorization Center
Precondition: Total has been computed.
Description:

1. Customer selects Pay by Credit/Debit card.
2. System reads the card.
3. If the card is a debit card, the system prompts for PIN. Customer enters a PIN on keyboard.
4. System sends message containing customer information to the appropriate Authorization Center.
5. If accepted, the Authorization Center returns an authorization code.
6. System prints last four digits of card number and authorization code on the customer receipt.
7. System prints two copies of a receipt on the card receipt printer.
8. Customer signs one receipt and leaves it with the supermarket.

Alternatives:

- If the card is not accepted and no other payment is offered, the order is canceled.
- If the system does not recognize the card, the card is ejected.

Postcondition: Payment has been made with a credit/debit card.

Use Case Name: Pay by Cash
Summary: Customer pays by cash.
Dependency: Extension of Checkout Order use case.
Actor: Customer
Precondition: Total has been computed.
Description:

1. Customer selects Pay by Cash.

2. The system prompts for cash.
3. Customer puts money into bill and coin readers.
4. If the system determines sufficient payment is made by cash, it dispenses change if necessary and prints receipt.
5. System displays welcome menu.

Postcondition: Payment has been made with cash.

Use Case Name: Restock Product Inventory
Summary: Restocker restocks inventory into system. Information about each product being restocked is entered into the system.
Actor: Restocker
Precondition: Menu is displayed to restocker.
Description:
1. Restocker selects Restock from menu.
2. The system prompts restocker to enter items.
3. Restocker scans items to be restocked.
4. The system prompts for the quantity.
5. The restocker enters the quantity to be restocked.
6. The system displays the quantity.

Alternatives:
- If the scanned item is new then the system prompts and the restocker enters the item description, price, discount information, and quantity.

Postcondition: Inventory has been restocked.

Use Case Name: View/Update Product Inventory
Summary: Information about a product is viewed/updated including price, descriptions, and discount information.
Actor: Manager
Precondition: Menu is displayed to manager.
Description:
1. Manager selects View/Update from menu.
2. System prompts and manager selects an item to view/update.
3. System displays the price, description, and discount information of the item.
4. Manager updates any information by typing over if necessary and enters when done (OK, SUBMIT, ENTER key or graphical user interface button).
5. System updates information if necessary and displays new values.

Postcondition: Product inventory is viewed/updated.

Use Case Name: Print Daily Report
Summary: Daily Reports are provided periodically. They are sent to a printer for the manager.
Actor: Timer
Precondition: The transaction log is stored.
Description:

1. Timer signals to the system to prepare daily reports at midnight.
2. System reads transaction log to compute the number of items sold for each product and the total revenue for the day.
3. System prints report on supermarket printer.

Postcondition: The daily reports have been completed

Use Case Name: Print Inventory Message
Summary: Buffered inventory messages are printed periodically.
Actor: Timer
Precondition: Inventory messages have been stored by system.
Description:

1. Timer signals to the system to prepare daily inventory report at midnight.
2. System prints daily inventory report on supermarket printer.

Postcondition: Inventory messages have been printed.