Team name: NewGen

Product Name: NewGen POS

Scrum Master: Michael Robertson (mirob2005@gmail.com)

Members:

Name: Xinchao Liu
Email: lookfor@188.com
Background: Knowledge of Microsoft Office suite, C++ & Java Programming Languages, Web Programming (PHP & MySQL)

Name: Wee Siang Wong
Email: willydk@gmail.com
Background: Knowledge of Microsoft Office Visio, Java, C++.

Name: Bryan Tamada
Email: btamada@gmail.com
Background: Knowledge of MS Office, C++, Java, PHP, and MySQL.

Name: Michael Robertson
Email: mirob2005@gmail.com
Background: Knowledge of MS Office, C++, Java, Scripting Languages (Python, Ruby/Rails, Perl), Database Programming (MySQL), Web and Graphics Programming

Name: Wei Jen Lin
Email: littlemike44@hotmail.com
Background: Knowledge of MS Office, Web Programming (PHP & MySQL)

Management:

Development Process

- Scrum with some of the UP practices
- XP practices with additional practices as necessary.

Software Tools

- MS Office (Visio, Word, etc.)
- Java IDE (Eclipse, NetBeans, XCode, etc.)
- GUI Framework (Qt)
- Version Control (GitHub)

Programming Languages

- Java

Operating System

- Windows
- Unix

Database Management

- Some version of SQL (MySQL, SQLite, etc.)

Computing Environment

- PC
- Mac

The team should be able to provide for all software/hardware necessary to complete this project.

Product Vision Statement:

Our Point-Of-Sale (POS) system is intended for employees of a grocery chain who need a user-friendly POS system to record sales of products to customers. The Point-Of-Sale system is an information system that will allow cashiers to record sales of products to customers including the sale data such as date, product name, price, total sale price, method of payment, along with allowing management to keep track of all product sales. Unlike the current system, our product will be user-friendly as well as being simple and easy to operate.

Product Scope:

Our Point-Of-Sale (POS) system will provide the employees of the grocery chain to record sales of any products sold along with storing sale data such as the date of the sale, employee who made the sale, specific items sold, price of each item, total sale price, method of payment, customer information, etc. The system will support such payment methods such as credit, cash, and check along with each method's required information. The system will calculate the total price including tax for the selected items rung up for sale. The cashier will record the items selected to be purchased and the system will respond with a total sale price along with a request of payment method. The cashier will then select the payment method and the system will respond accordingly. If the payment method is cash, the cashier will be asked to input the amount of cash given by the customer and the system will display the amount of cash back. If the payment method is credit, the system will request the credit card type along with the card number and expiration date. If the payment method is check, the system will request the name and bank information. Once the cashier provides this information, the system will authorize the information and return with either an accepted or declined form of payment. If declined, a new payment method must be used. The system will also keep track of employees allowing different levels of access to regular employees such as cashiers and those with higher access such as management. Cashiers' access will be limited to the features stated above, while management will be able to track sales by date, cashier, or customer. The system will produce accurate calculations as well as being easy to use, fast. and reliable. The system will be designed as a complete replacement for the current Point-Of-Sale (POS) system of the grocery chain.

Use-Case 1:

ID: 1

Name: Process Sale

Description: The POS system will record purchased items, calculate a total along with product names, customer payment method, update store inventory and print out a summary of purchased items in the form of a receipt for the customer.

Actors: cashier

Pre-condition: cashier ready

Basic steps:

1. customer checkout with goods
2. cashier start a new sale
3. cashier enter items
4. system present item name and price

cashier repeat step 3-4 until record all goods

5. system present total price with tax
6. cashier asks for payment
7. customer pays and system handles payment
8. system logs sales
9. system prints receipt
10. customer leaves with receipt and goods

Post-condition: amount compute correctly, payment authorized, sales saved, inventory updated, receipt generated.

Priority: high

Special requirements:

large font text to guarantee screen visibility
quick payment authorization response

Memo (open issues):

need to support manager's override operation

Use-Case 2:

ID: 2

Name: Handle Returns

Description: The POS system will record returned items, calculate a total along with product names, customer refund method, update store inventory and print out a summary of returned items in the form of a receipt for the customer.

Actors: cashier

Pre-condition: cashier ready

Basic steps:

1. customer presents returning goods to cashier
2. cashier start new return process
3. cashier enter items
4. system present returned item name and price
cashier repeat step 3-4 until all goods are recorded
5. store inventory gets updated
6. system present total price and tax of returned goods
7. cashier pays customer refund amount and system handles refund
8. system logs return
9. system prints receipt
10. customer leaves with receipt and refund

Post-condition: refund amount computed correctly, refund authorized, refund saved, inventory updated, receipt generated.

Priority: high

Special requirements:

quick payment authorization response

Memo (open issues):

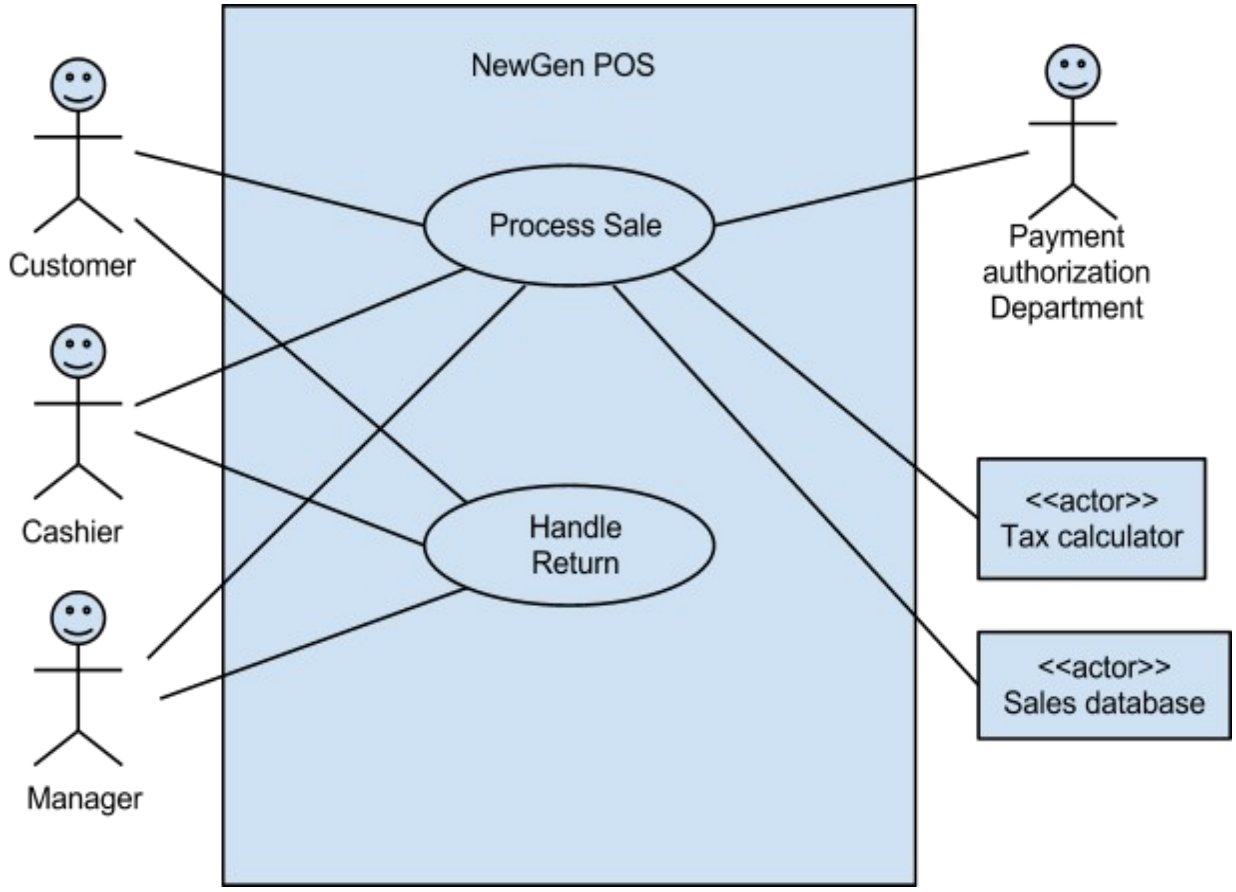
need to support manager’s override operation

Use-Case Justifications:

For a real-life Point-Of-Sale system, processing sales is the most important function. The system should allow a cashier to complete a sale by having the cashier enter the selected items. The system will then calculate the total price after taxes, allow the cashier to enter a payment method, and then record the transaction once completed.

After a sale is completed, the system should also be able to handle a refund in case the customer decides to return an item. In this case, the system will perform a similar procedure to processing sales but in the opposite direction and once again record the transaction in the database once completed.

Use-Case Diagram:



Activity Diagram: