

380 Project

Contributors: Thomas Hanley, Josh Meritt, Jondn Tryniski

Project Description:

The system created will be an inventory and sales tool. The tool will have users which allow us to login and track individual sales. The tool will also allow for us to track inventory. It will allow us to input what inventory we have through commands built into the tool. We will be able to input a command to check and see general inventory as well as what is currently low on inventory. We want to be able to look at the inventory by the department type. The tool will also go through and give suggestions on what to order based on what is selling well, and current stock levels. The tool will be able to give daily and weekly reports of what is sold, overall sales, as well as individual users sale totals. If the system is shut down, the tool should have saved the information and be able to load if after turning the system on again. The tool will also allow for general editing of price of inventory so that changes may be made over time. The tool should allow different employees to have different levels of access in the inventory. Another tool is the different level of memberships for customers. Every customer does not have to be a member but it is a choice and the more money a member spends, the greater the rewards are. The three levels of membership are: bronze, silver and gold with gold being the highest level and bronze being the lowest. The system keeps track of how much each member spends and moves them up in level once they hit the threshold for amount spent.

System Requirements:

Requirement Scale Priority:1-5 1-Highest, 5-Lowest

Requirement 1: P1 The system will track sales totals.

Requirement 2: P3 The system will allow for creation of users.

Requirement 3: P3 The user will be able to authenticate into the system via a password.

Requirement 4: P4 The system will track individuals sales totals.

Requirement 5: P4 The system will alert when running low on certain products once a threshold is met.

Requirement 6: P5 The system will take into account current sales to help predict when to order products.

Requirement 7: P2 The system will have a command for showing current stock levels.

Requirement 8: P2 The system will allow for adding prices to items in inventory.

Requirement 9: P2 The system will allow for editing of item's prices in inventory.

Requirement 10: P5 The system will have daily and weekly reports of sales.

Requirement 11: P3 The system will be organized so that one can see inventory in different departments.

Requirement 12: P3 The system will be able to be shut down and loaded again with having the same information as before it was shut down.

Requirement 13: P4 The system will be able to calculate the net profit of the inventory daily and weekly.

Requirement 14: P4 The system will allow different levels of access depending on your position in the company.

Requirement 15: P2 The system will allow users to add and remove items manually.

Requirement 16: P4 The system will have a members system.
Requirement 17: P5 The system can add or delete members.
Requirement 18: P5 The system will give different level of members better deals.

User Stories:

User Story 1: As a manager of a store, I want to be able to load in my inventory after shutting the system down. Time: 8
User Story 2: As a manager of a store, I want to be able to add new users so that I can have more cashiers. Time: 5
User Story 3: As a manager of a store, I want to be able to see what items are popular so that I can restock accordingly. Time: 4
User Story 4: As a manager of a store, I want to be able to see what is sold daily and weekly in order to see what is selling. Time: 5
User Story 5: As a manager of a store, I want to be able to count the inventory so that I can see what needs to be restocked. Time: 4
User Story 6: As a manager of a store, I want to be able to see how much money was made each day and each week. Time: 5
User Story 7: As a manager of a store, I need to see how much my employees are selling in order to see how productive they are. Time: 6
User Story 8: As a manager of a store, I want to have restrictions on who has access to certain parts of the inventory. Time: 4
User Story 9: As a cashier, I want to be able to log into my personal profile so that I can keep track of what I sell. Time: 5
User Story 10: As a cashier, I want to be able to cash people out so that I can make sales and keep my job. Time: 4
User Story 11: As a customer, I want to be able to become a member to get better deals. Time: 4

Traceability Matrix:

Use Cases Case 1: Load in Inventory
 Case 2: Add Users
 Case 3: Save Inventory
 Case 4: See Inventory
 Case 5: See the Distribution of Items sold
 Case 6: Have a rating of how much employees are selling
 Case 7: Sell items
 Case 8: Return items
 Case 9: Log in
 Case 10: Levels of access for users
 Case 11: See Profits
 Case 12: Add or Delete Members
 Case 13: Change level of membership

Use Case	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13
User Story													
1	X												
2		X					X						
3	X				X								
4	X			X	X		X						
5	X		X	X			X	X					
6											X		
7		X				X	X		X	X			
8										X			
9		X							X	X			
10		X					X		X				
11												X	X

Use Case 1, Load in Inventory: Related User Stories: User Story 1, 3, 4, 5. Initiating Actor: Manager, Key Holder. Actor's Goal: To load in either the starting inventory or an updated inventory. Participating Actors: Inventory, Item, InventoryParser. Preconditions: -The inventory is blank or up to date with new additions and how it was previously. -The inventory needs to be updated. Postconditions: The inventory will be updated and be able to used by Register. Flow of Events: -New/Updated inventory is given to the manager or key holder -The employee initiates loading the inventory into the system. Includes InventoryParser and .parse(). -The inventory is now loaded and can be used in the register.

Use Case 2, Add Users: Related User Stories: User Story 2, 7, 9, 10. Initiating Actor: Manager. Actor's Goal: To create new users to use the register for returns, sales, changing item prices, loading inventories. Participating Actors: User Preconditions: -The user does not already exist. -The user name is valid. -The user password meets minimum security specifications. Postconditions: The new user is created. Flow of Events: -Manager gets information for the new user. -The manager inputs the new user information setUsername, setPassword. -The new user is now created.

Use Case 3, Save Inventory: Related User Stories: User Story 5. Initiating Actor: Manager Key Holder, Sales Clerk. Actor's Goal: To save the inventory in order to be loaded again. Participating Actors: Inventory, InventoryParser. Preconditions: -The inventory is not null. -The inventory is current. Postconditions: -The inventory is saved so it can be loaded in again. Flow of Events: -The user chooses to save the inventory close method in Register -The inventory is saved to a .xml file.

Use Case 4, See Inventory: Related User Stories: User Story 4, 5. Initiating Actor: Manager, Key Holder, Sales Clerk. Actor's Goal: To see the available inventory in the store. Participating Actors: Inventory, Register Preconditions: -The inventory is not null. -The inventory is loaded in. Postcondition: The inventory is displayed. Flow of Events: -User uses command to see the inventory checkInventory().

Use Case 5, See the Distribution of Items sold: Related User Stories: User Story 3, 4. Initiating Actor: Manager, Key Holder, Sales Clerk. Actor's Goal: To see what has been sold in order to get trends. Participating Actors: Inventory, Register. Preconditions: -The inventory is not null. -The inventory is loaded into the register -Items have been sold. Postcondition: The amounts that have been sold is displayed. Flow of Events. -Inventory is loaded into register -Sales, and returns occur sale(), return() -Displays of sales can be generated. dailySalesTotals(), weeklySalesTotals().

Use Case 6 Rating of how much employees are selling.: Related User Story 7. Initiating Actor: Manager. Actor's Goal: To see what has been sold. Participating Actors: Register, Inventory. Preconditions: -More than one user has sold inventory -Inventory has been sold -All inventory sold has been properly tracked and logged by sales clerks. Postcondition: The ratings will be displayed Flow of Events -Inventory is loaded into register -Sales and returns occur. sale(), return(). -Manager initiates getting ratings -Ratings are displayed dailyReport(), weeklyReport().

Use case 7: Sell Items Required Requirements: R1, R7, R15 Initiating Actor: Cashier Actor's goals: For cashier to be able to sell items Participating Actor: Register, Inventory, Items Preconditions: The items is in the inventory Postconditions: There is less of that item in the inventory Flow of events for Main success scenario: --> Customer brings item to Cashier to sell <-- Register uses UC4 and checks inventory for item to find price of item for customer to pay and finds money back

Use case 8: Return Items Required Requirements:R7, R15 Initiating Actor: Cashier Actor's goals: Return an item to the inventory and give money back to the customer Participating Actor: Item, Register Preconditions: The item was in the inventory Postconditions: The item now has one more to it's amount Flow of events for Main success scenario: --> Customer brings item to return <-- Register searches through inventory to find price of item <-- Register calculates how much money to give back to Customer

Use case 9: Log in Required Requirements:R2, R3 Initiating Actor:Manager, Cashier Actor's goals:Log in to the worker's account Participating Actor: User Preconditions: The user has a user name and password already in the system Postconditions: They user is logged in Flow of events for Main success scenario: --> Manager creates a new account (UC2) <-- User creates account for new worker if they don't have existing account UC2 <-- User must enter password of their user name to get access, otherwise they cannot get into their account

Use case 10: Levels of access for users Required Requirements: R2, R3, R14 Initiating Actor: Manager Actor's goals: The manager has access to everything while cashiers have limited access Participating Actor: User, Register Preconditions: The cashier that has an account cannot access a lot compared to the manager Postconditions: Manager has access to

everything Flow of events for Main success scenario: --> Manager creates account (UC2) <-- Register allows cashier to have less access and not view everything on the system

Use case 11: See profits Required Requirements: R1, R4, R5, R7, R10, R13, Initiating Actor: Manager Actor's goals: View the profits of the company Participating Actor: Register, Inventory Preconditions: Items have been sold Postconditions: profits has increased Flow of events for Main success scenario: --> Manager must log in (UC9) --> Manager can check weekly/daily sales totals due to access (UC10)

Use case 12: Add members Required Requirements: R16, R17 Initiating Actor: Cashier or Manager Actor's goals: add a member Participating Actor: Memberlist, Member Preconditions: - Member doesn't already exist - Information for member is correct Postconditions: Member is added Flow of events for Main success scenario: --> Customer is asked if they want to be a member and is asked information <-- information is entered and member is added

Use case 13: Change level of membership Required Requirements: R16, R18 Initiating Actor: Cashier or Manager Actor's goals: change level of membership for a member Participating Actor: Memberlist, Member Preconditions: -Member exists -Member is moving up to correct level Postconditions: Level of membership was changed Flow of events for Main success scenario: --> Member spends enough money <-- Membership level is changed

UML





