Inventory Management System

Analysis phase(phase-1)

Minahil Hamid(13L-4050)

Hamza Saleem(13L-4106)

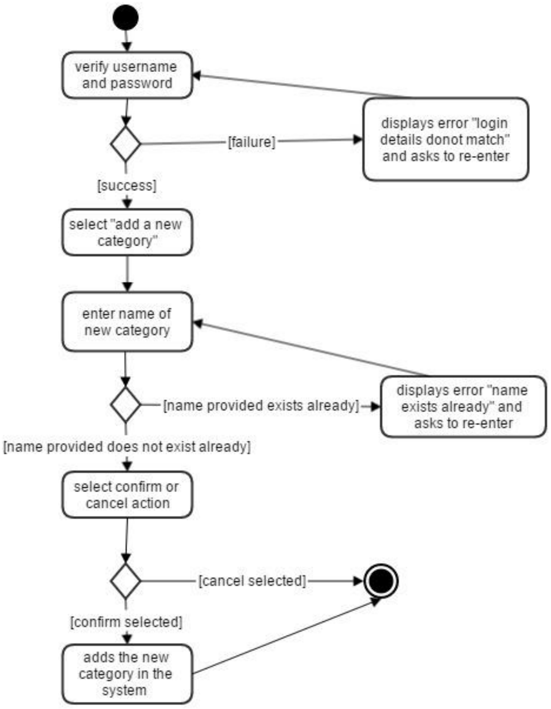
Muzammil Muneer(13L-4156)

Armish Ashraf(13L-4163)

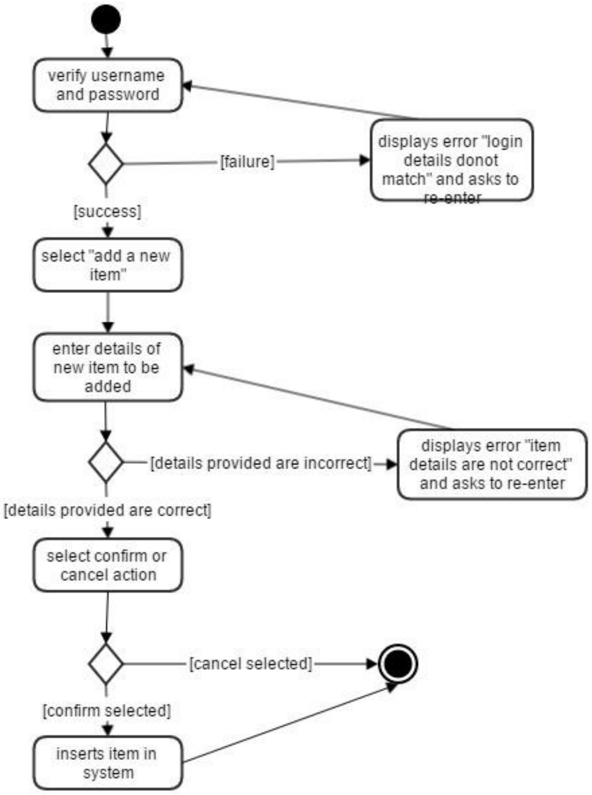
Muhammad Talal Rashid(13L-5883)

Activity Diagrams

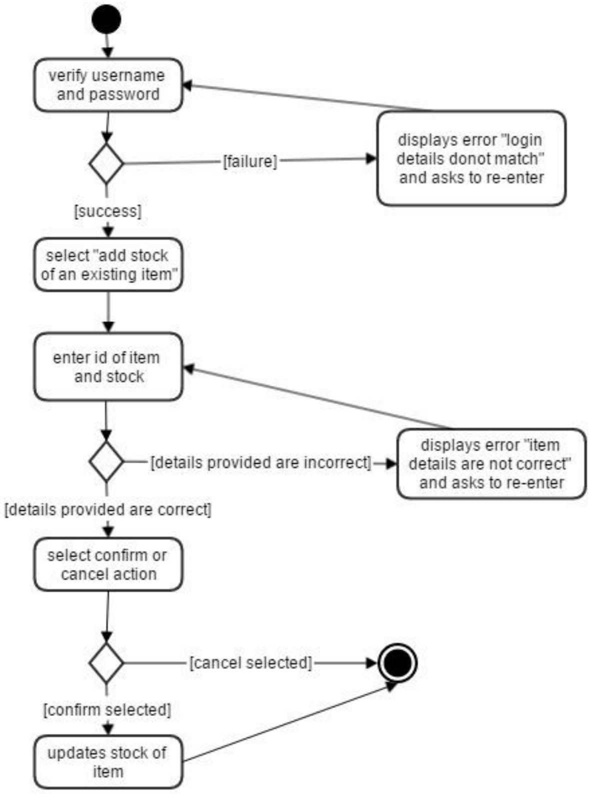
Add category


verify username and password
[failure]
displays error "login details donot match" and asks to re-enter
[success]
select "add a new category"
enter name of new category
[name provided exists already]
displays error "name exists already" and asks to re-enter
[name provided does not exist already]
select confirm or cancel action
[cancel selected]
[confirm selected]
adds the new category in the system

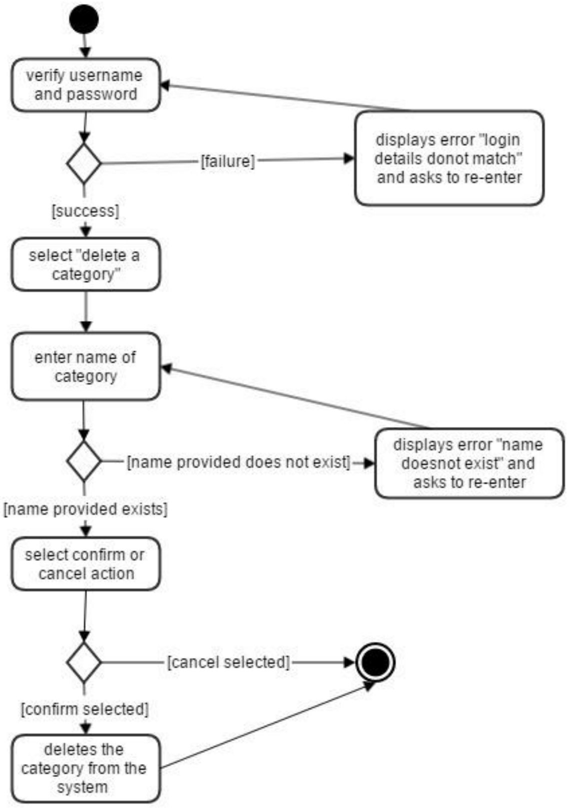
Add new Item


verify username and password
[failure]
displays error "login details donot match" and asks to re-enter
[success]
select "add a new item"
enter details of new item to be added
[details provided are incorrect]
displays error "item details are not correct" and asks to re-enter
[details provided are correct]
select confirm or cancel action
[cancel selected]
[confirm selected]
inserts item in system

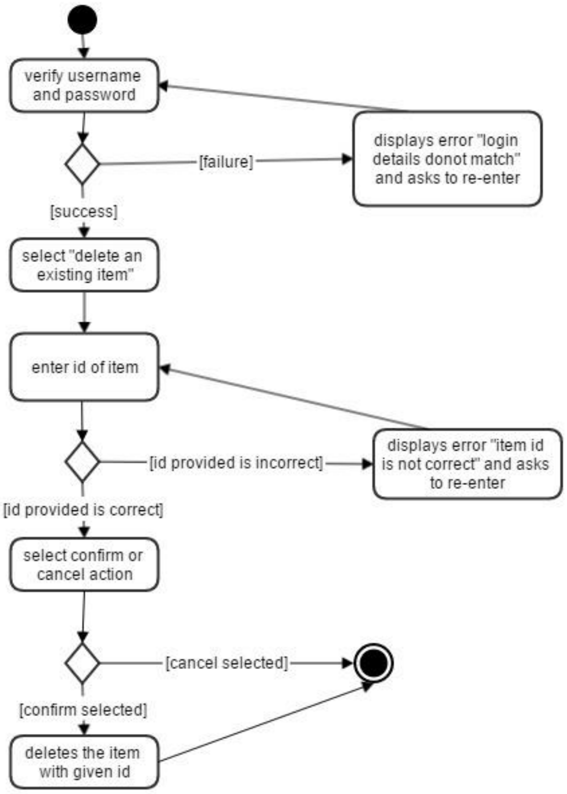
Add stock of existing item


verify username and password
[failure]
displays error "login details donot match" and asks to re-enter
[success]
select "add stock of an existing item"
enter id of item and stock
[details provided are incorrect]
displays error "item details are not correct" and asks to re-enter
[details provided are correct]
select confirm or cancel action
[cancel selected]
[confirm selected]
updates stock of item

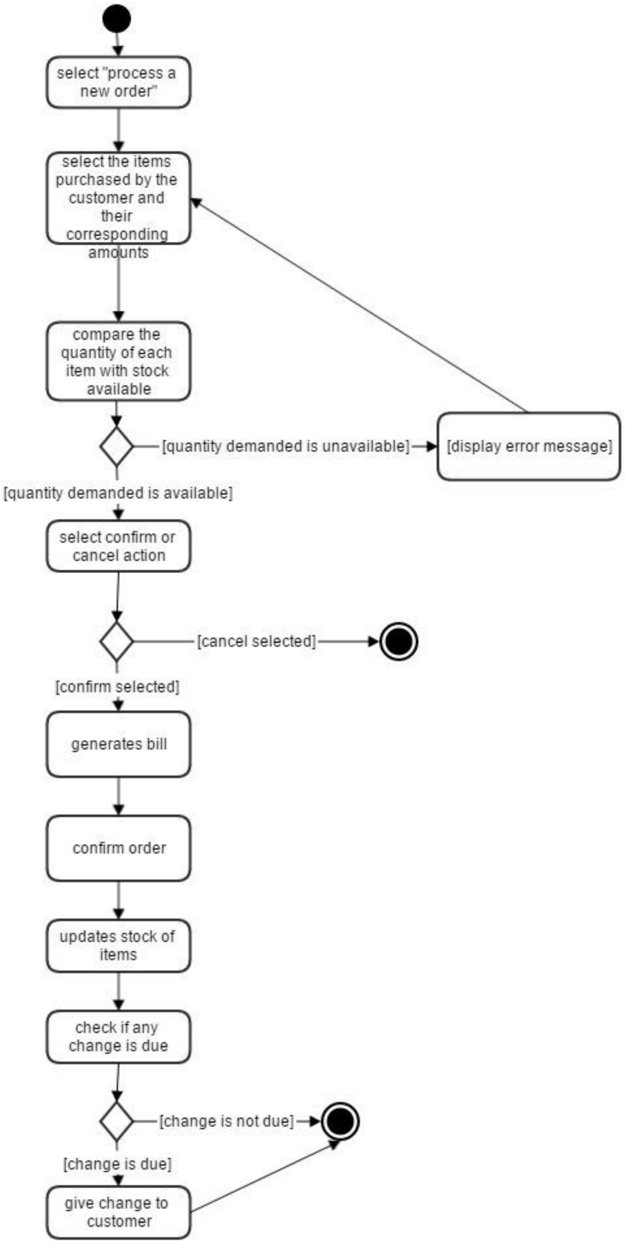
Delete category


verify username and password
[failure]
displays error "login details donot match" and asks to re-enter
[success]
select "delete a category"
enter name of category
[name provided does not exist]
displays error "name doesnot exist" and asks to re-enter
[name provided exists]
select confirm or cancel action
[cancel selected]
[confirm selected]
deletes the category from the system

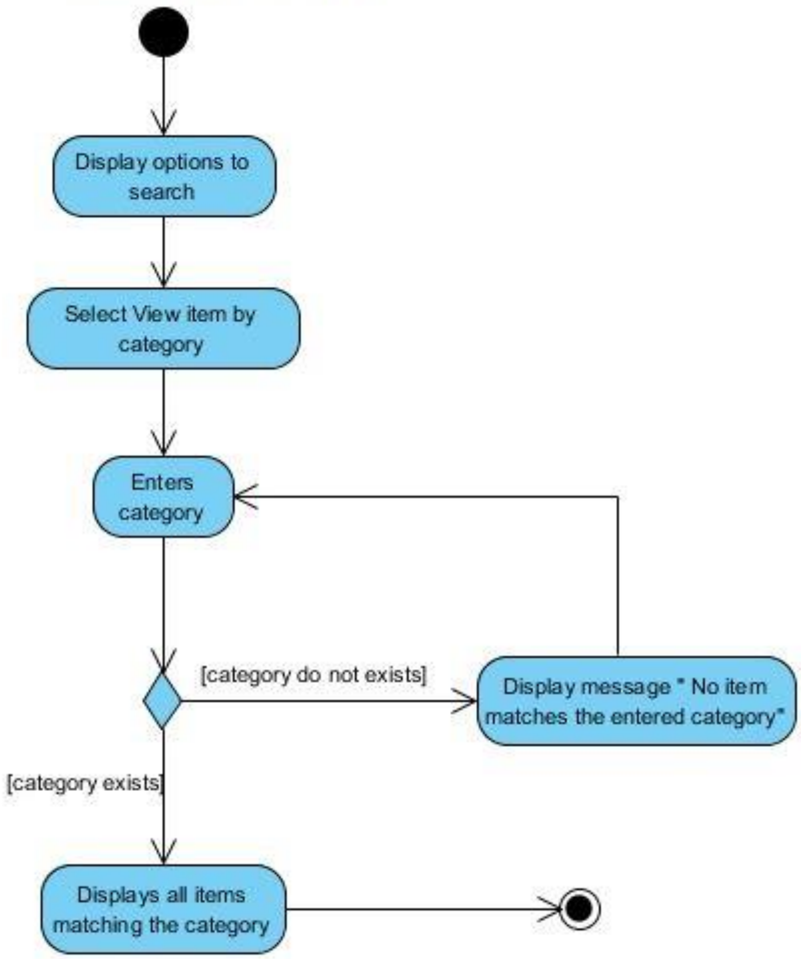
Delete an item


verify username and password
[failure]
displays error "login details donot match" and asks to re-enter
[success]
select "delete an existing item"
enter id of item
[id provided is incorrect]
displays error "item id is not correct" and asks to re-enter
[id provided is correct]
select confirm or cancel action
[cancel selected]
[confirm selected]
deletes the item with given id

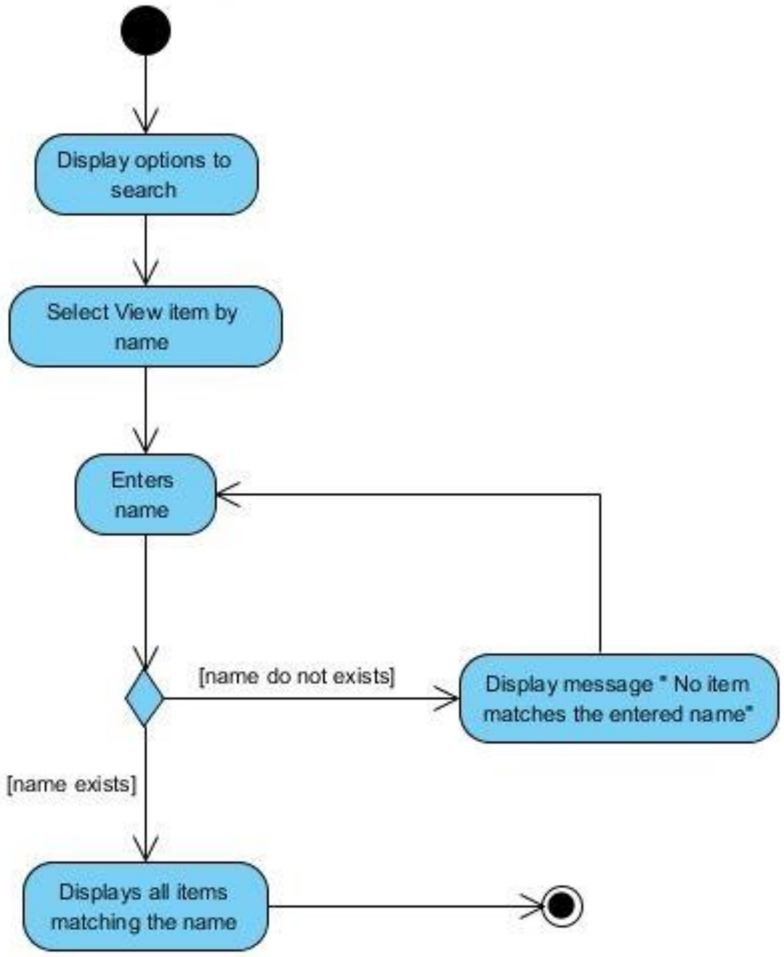
Process an order


select "process a new order"
select the items purchased by the customer and their corresponding amounts
compare the quantity of each item with stock available
[quantity demanded is unavailable]
[display error message]
[quantity demanded is available]
select confirm or cancel action
[cancel selected]
[confirm selected]
generates bill
confirm order
updates stock of items
check if any change is due
[change is not due]
[change is due]
give change to customer

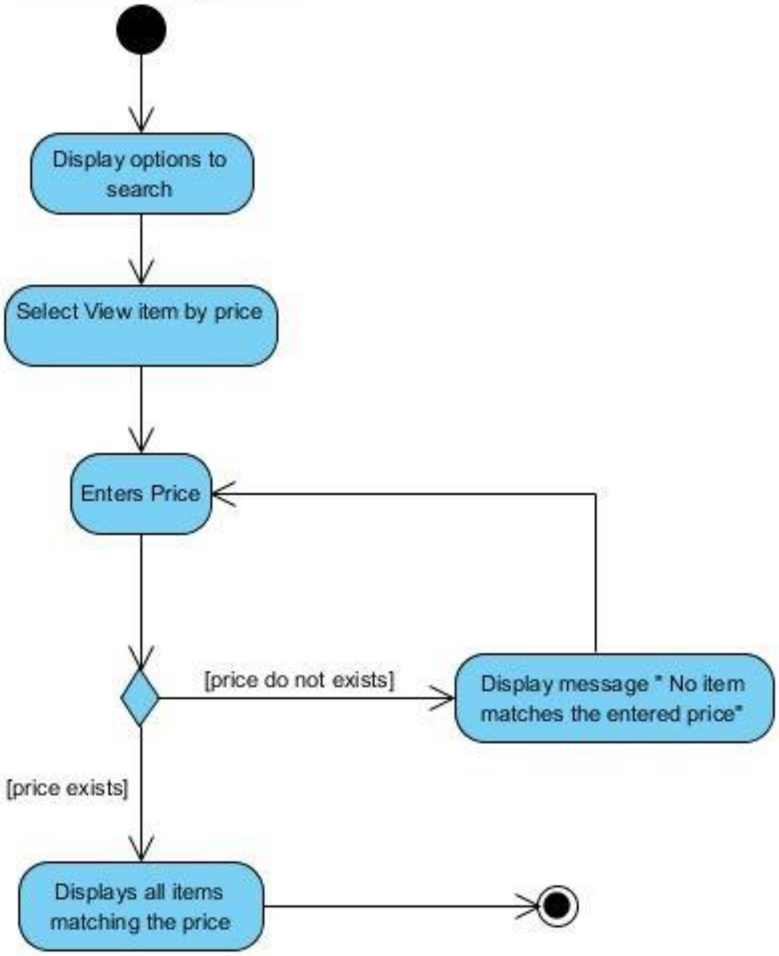
View item by category


Display options to search
Select View item by category
Enters category
[category do not exists]
Display message " No item matches the entered category"
[category exists]
Displays all items matching the category

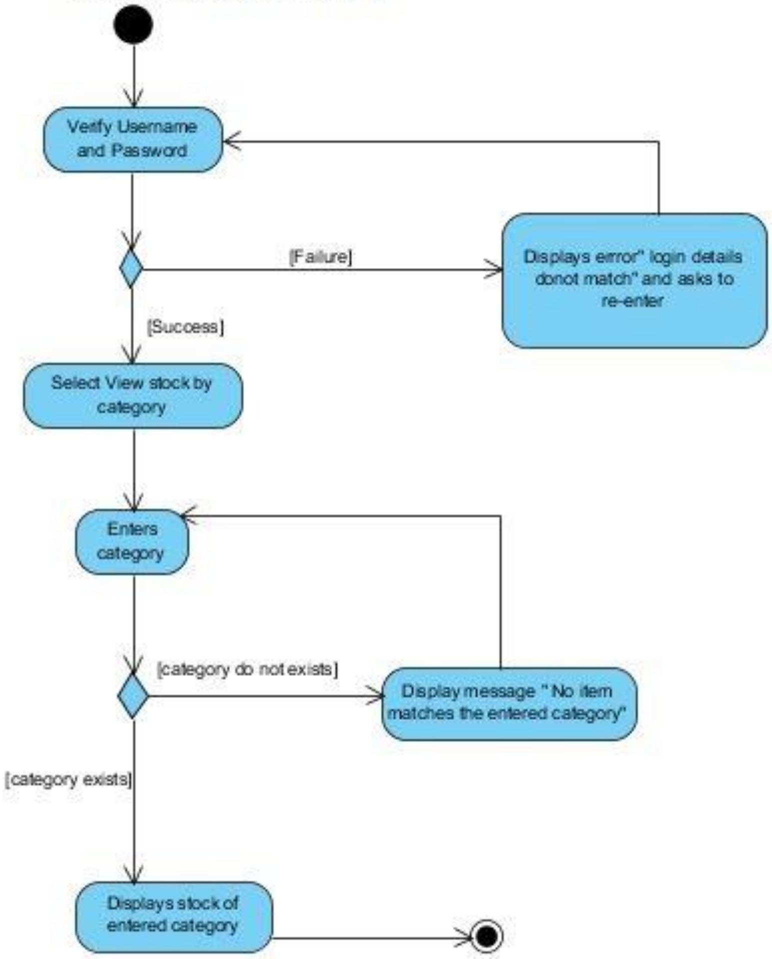
View item by name


Display options to search
Select View item by name
Enters name
[name do not exists]
Display message " No item matches the entered name"
[name exists]
Displays all items matching the name

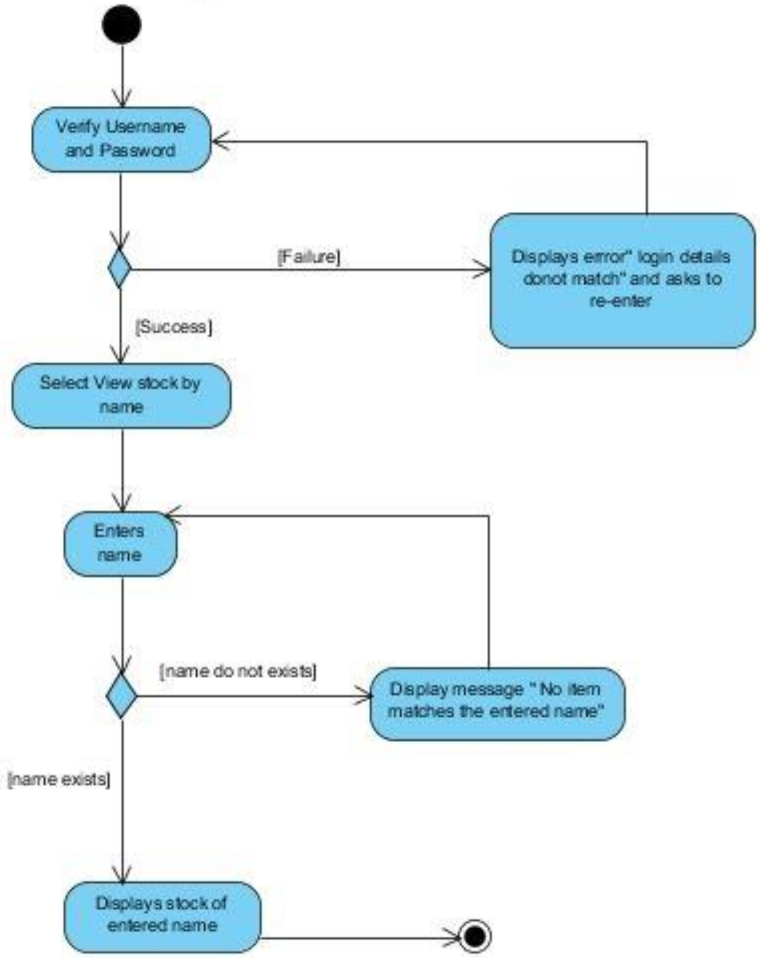
View item by price


Display options to search
Select View item by price
Enters Price
[price do not exists]
Display message " No item matches the entered price"
[price exists]
Displays all items matching the price

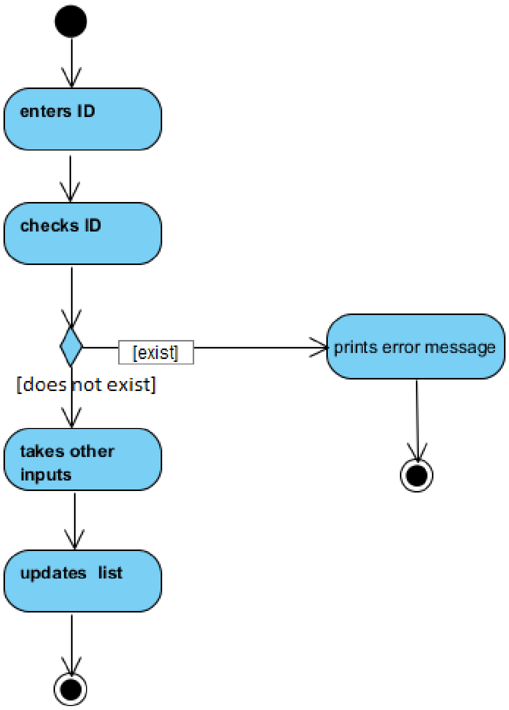
View stock by category


Verify Username and Password
[Failure]
Displays error" login details donot match" and asks to re-enter
[Success]
Select View stock by category
Enters category
[category do not exists]
Display message " No item matches the entered category"
[category exists]
Displays stock of entered category

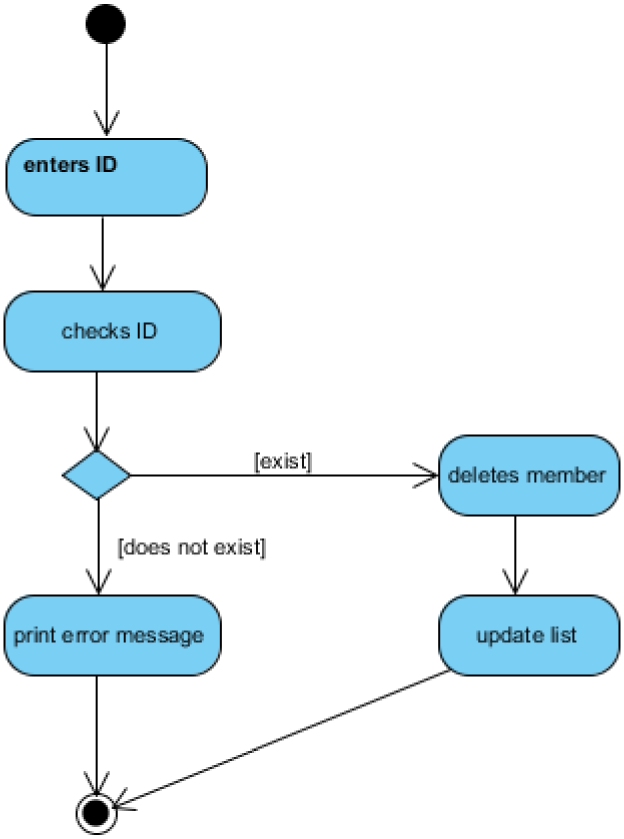
View stock by name


Verify Username and Password
[Failure]
Displays error" login details donot match" and asks to re-enter
[Success]
Select View stock by name
Enters name
[name do not exists]
Display message " No item matches the entered name"
[name exists]
Displays stock of entered name

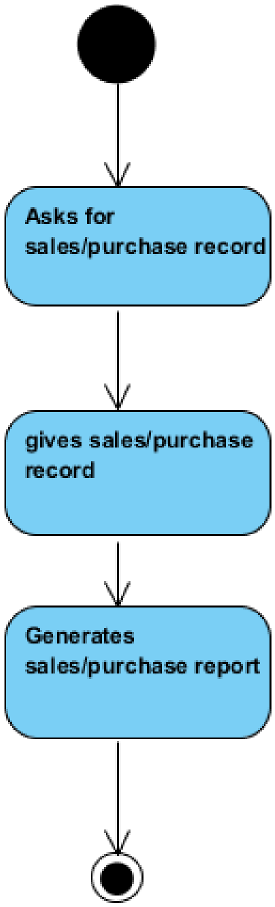
adds staff member



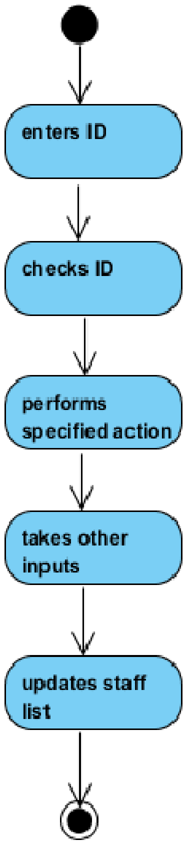
Removes staff member



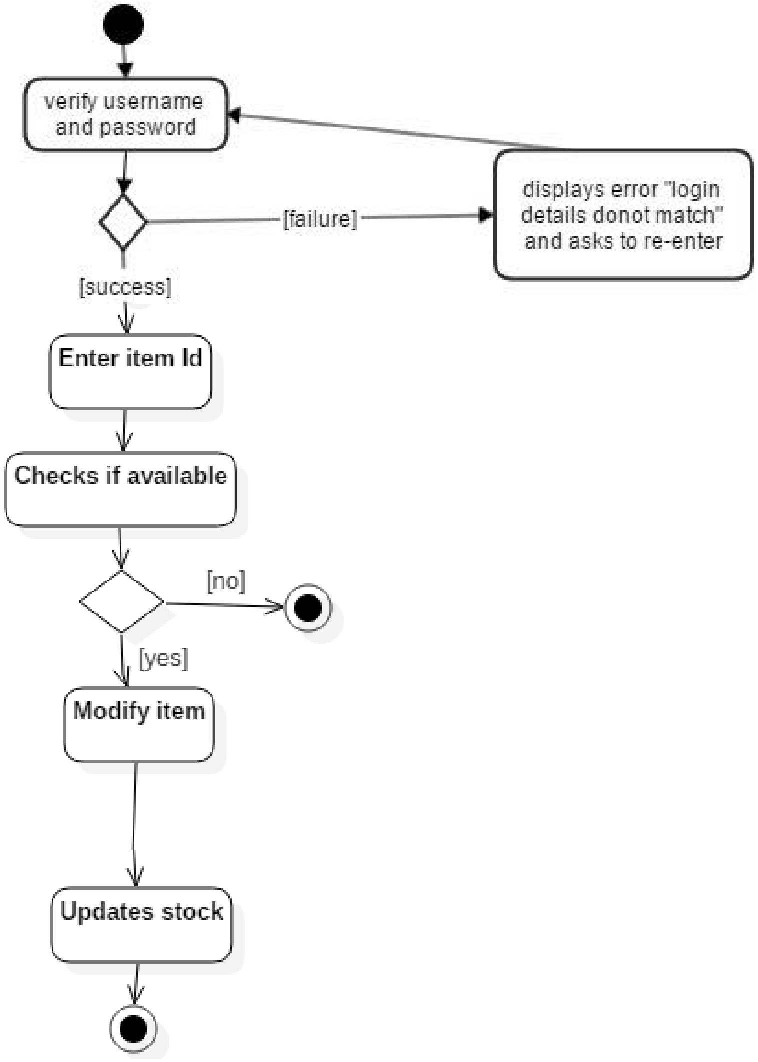
generate report


Asks for sales/purchase record
gives sales/purchase record
Generates sales/purchase report

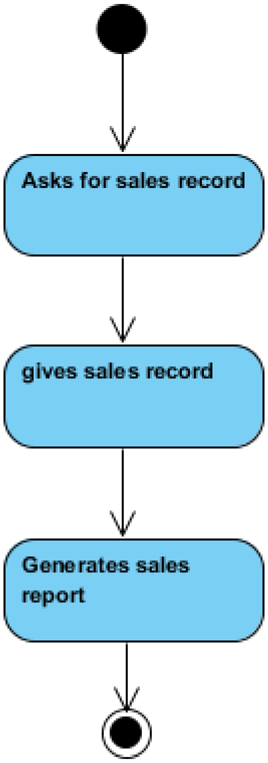
Modify staff

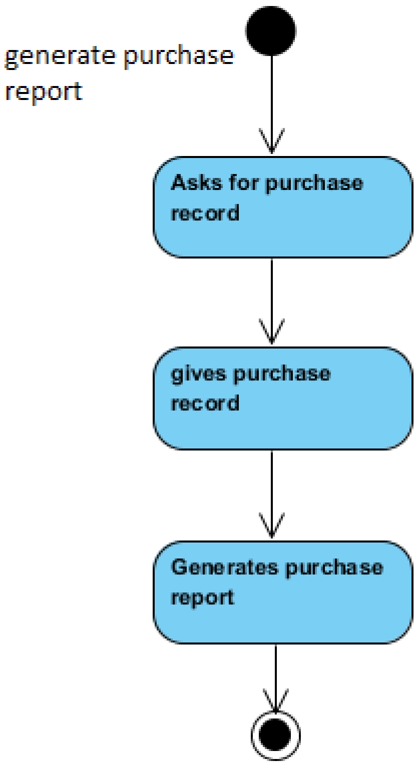


Modify item

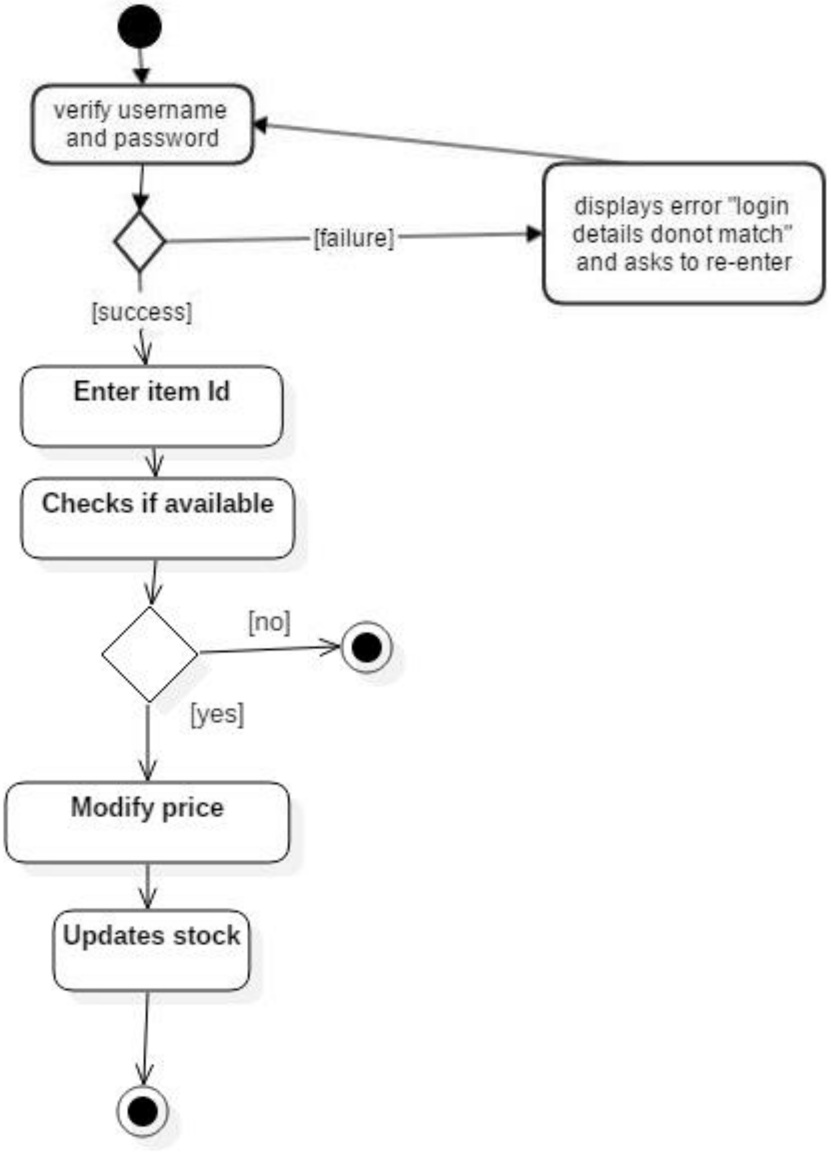

verify username and password
[failure]
displays error "login details donot match" and asks to re-enter
[success]
Enter item Id
Checks if available
[no]
[yes]
Modify item
Updates stock

generate
sales

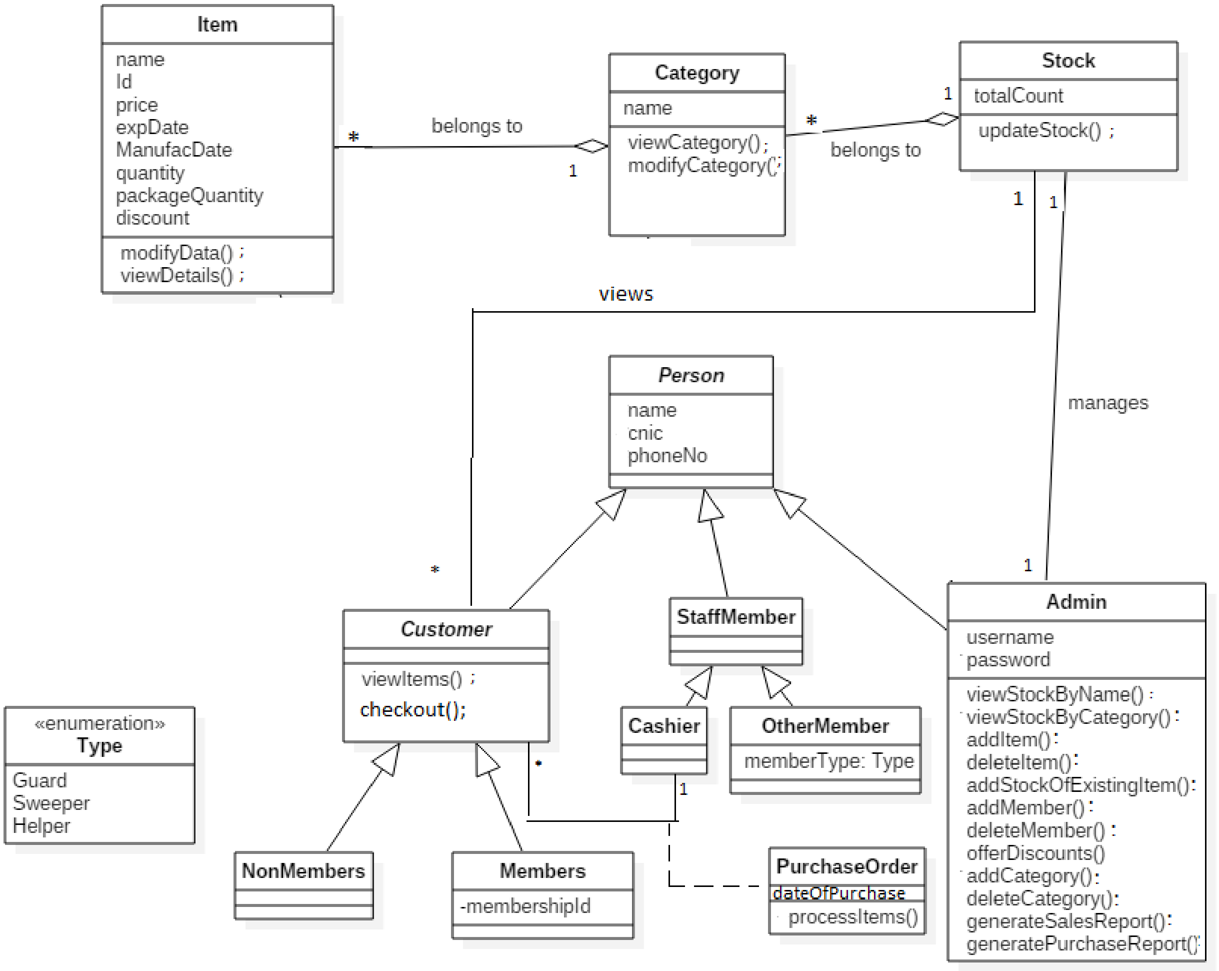



generate purchase report
Asks for purchase record
gives purchase record
Generates purchase report

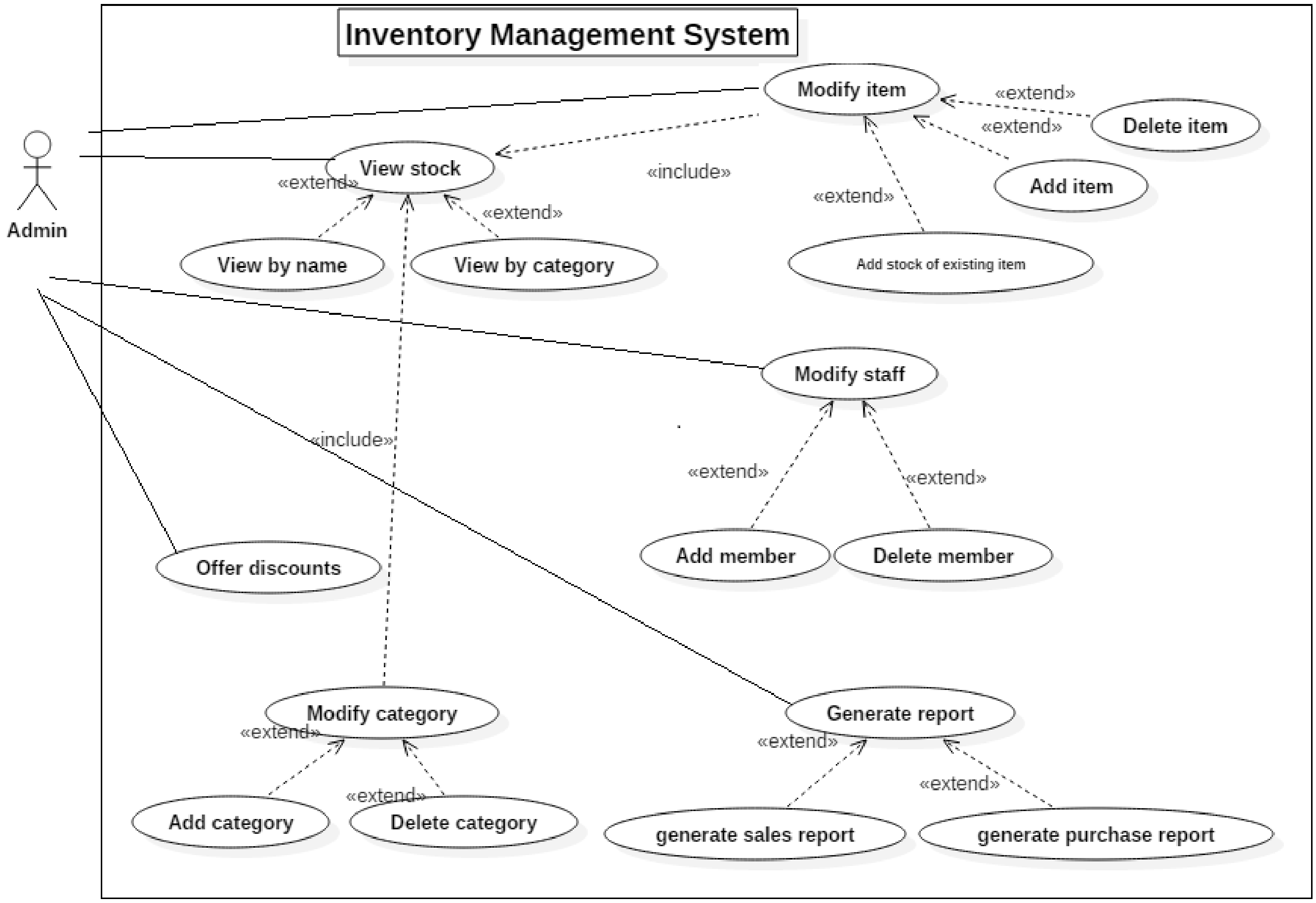
Offer discounts

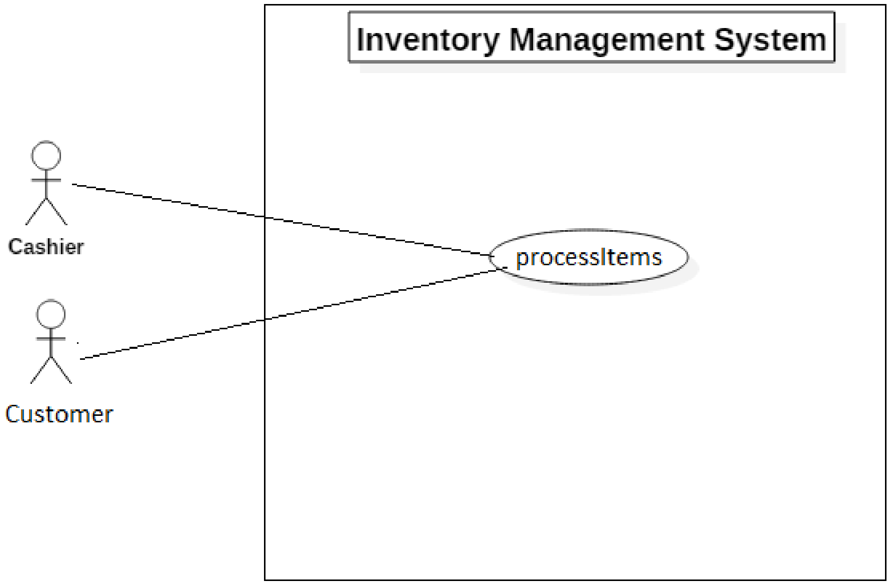

verify username and password
[failure]
displays error "login details donot match" and asks to re-enter
[success]
Enter item Id
Checks if available
[no]
[yes]
Modify price
Updates stock

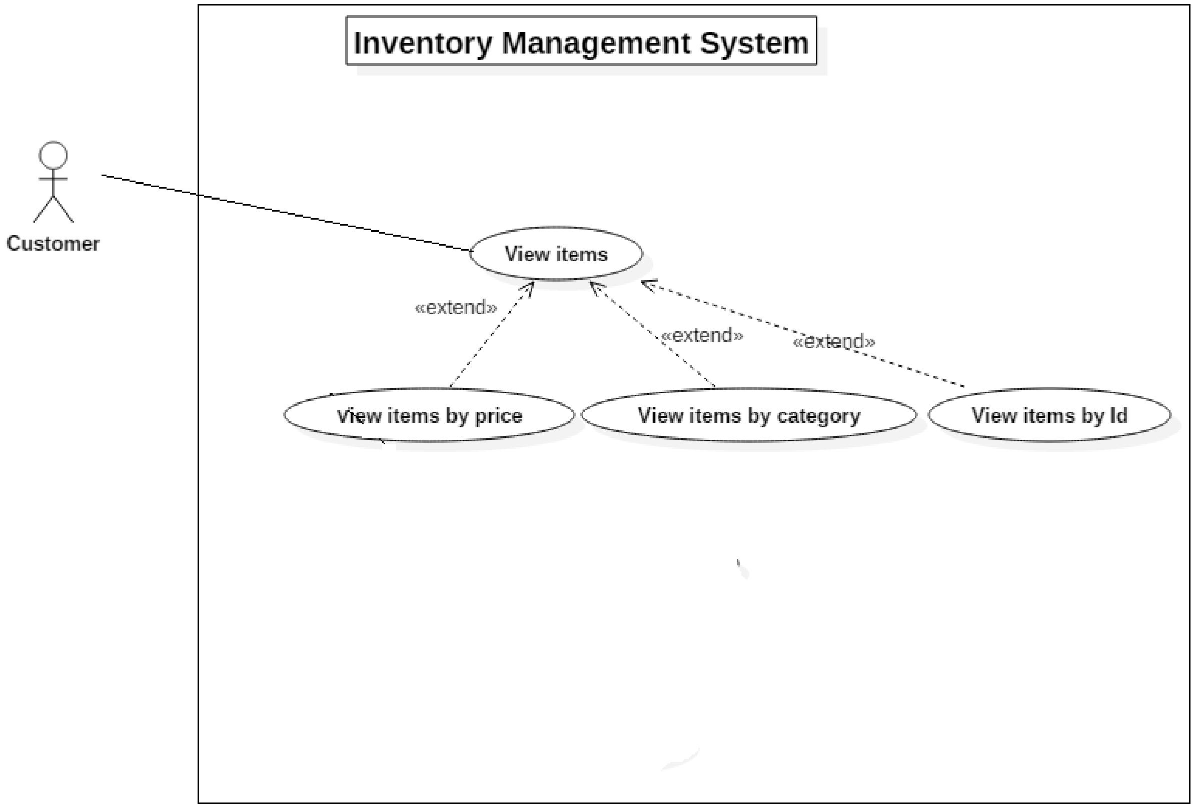
Analysis Class Diagram



Use-Case Diagrams




Inventory Management System
Cashier
Customer
processItems


Inventory Management System
Customer
View items
«extend»
«extend»
«extend»
View items by price
View items by category
View items by Id

Use-Case Templates

<table>
<tr><td colspan="2"><b>Identifier</b></td><td colspan="2">UC-1</td></tr>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">View Stock</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">This use case describe how an administrator view stock in the inventory.</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Administrator</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td colspan="2">System is online</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td colspan="2">Display Stock of every item in the inventory</td></tr>
<tr><th colspan="4">Typical Course of Action</th></tr>
<tr><th>S#</th><th colspan="2">Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td colspan="2">Enter username and password</td><td></td></tr>
<tr><td><b>2</b></td><td colspan="2"></td><td>Displays menu.</td></tr>
<tr><td><b>3</b></td><td colspan="2">Selects View Stock option either by category or by name.</td><td></td></tr>
<tr><td><b>4</b></td><td colspan="2"></td><td>Displays stock either by category or by name.</td></tr>
</table>

Identifier	UC-2
Name	View Stock by name
Summary	This use case describe how an administrator view stock in the inventory filtered by name.

Priority	High
Actors	Administrator
Pre-condition(s)	System is online
Post-condition(s)	Display Stock of every item in the inventory

Typical Course of Action

S#	**Actor Action**	**System Response**
1	Enter username and password	
2		Displays menu.
3	Selects View Stock option	
4		Displays view stock options
5	Selects View Stock by name option	
6		Prompts for name
7	Enters name of the item	
8		Displays stock of the entered item

Alternate Course(s) of Action

S#	**Actor Action**	**System Response**
7	Enters name of the item.	
8		Displays message “No item matches the entered name”
9		Go to step 6.

Identifier	UC-3
Name	View Stock by category
Summary	This use case describe how an administrator view stock in the inventory filtered by category.
Priority	High
Actors	Administrator
Pre-condition(s)	System is online
Post-condition(s)	Displays Stock of every item in the category and waits for further instruction.

Typical Course of Action

S#	**Actor Action**	**System Response**
1	Enter username and password	
2		Displays menu.
3	Selects View Stock option	
4		Displays view stock options

5	Selects View Stock by Category option	
6		Prompts for category
7	Enters name of the category	
8		Displays stock of every item in the selected category
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
7	Enters name of the category.	
8		Displays message “ Entered category does not exists”
9		Go to step 6.

Identifier	UC-4	
Name	View item	
Summary	This use case describe how a customer view stock in the inventory.	
Priority	High	
Actors	Customer	
Pre-condition(s)	System is online	
Post-condition(s)	Display Stock of every item in the inventory	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1		Displays menu.
2	Selects View item option either by category, by name or by price.	
3		Displays items either by category, by name or by price.

Identifier	UC-5
Name	View item by price
Summary	This use case describe how a customer view stock in the inventory filtered by price.
Priority	High
Actors	Customer
Pre-condition(s)	System is online

Post-condition(s)	Displays item matching the price and waits for further instruction.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1		Displays options to search
2	Select view item by price option	
3		Prompts for price
4	Enters price.	
5		Displays all the items matching the price entered.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
7	Enters price.	
8		Displays message “ No item matching the price”
9		Go to step 3.

Identifier	UC-6	
Name	View item by name	
Summary	This use case describe how a customer view stock in the inventory filtered by name.	
Priority	High	
Actors	Customer	
Pre-condition(s)	System is online	
Post-condition(s)	Displays item matching the name and waits for further instruction.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1		Displays options to search
2	Select view item by name option	
3		Prompts for name
4	Enters name.	
5		Displays all the items matching the name entered.
Alternate Course(s) of Action		

S#	Actor Action	System Response
6	Enters name.	
7		Displays message “ No item matching the name”
8		Go to step 3.

Identifier	UC-7
Name	View item by category
Summary	This use case describe how a customer view stock in the inventory filtered by category.
Priority	High
Actors	Customer
Pre-condition(s)	System is online
Post-condition(s)	Displays item matching the category and waits for further instruction.

Typical Course of Action

S#	Actor Action	System Response
1		Displays options to search
2	Select view item by category option	
3		Prompts for category
4	Enters category.	
5		Displays all the items matching the category entered.

Alternate Course(s) of Action

S#	Actor Action	System Response
6	Enters category.	
7		Displays message “ No item matching the category”
8		Go to step 3.

Identifier	UC-8
Name	Modify Item
Summary	The admin can modify the items present in the stock. New items can be added. Existing items can be deleted and their quantity can also be modified.
Priority	High
Actors	Admin

<table>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">System is waiting for instructions.</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">Changes in the items are saved and the system waits for further instructions.</td></tr>
<tr><th colspan="3">Typical Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td>The admin provides his login details.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>The system checks these details and logs the admin in.</td></tr>
<tr><td><b>3</b></td><td>The admin selects either to add a new item or change the stock of any existing item.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>The system asks the admin the details needed.</td></tr>
<tr><td><b>5</b></td><td>The admin provides these details.</td><td></td></tr>
<tr><td><b>6</b></td><td></td><td>The system checks these details and asks the admin to confirm the action.</td></tr>
<tr><td><b>7</b></td><td>The admin confirms the action.</td><td></td></tr>
<tr><td><b>8</b></td><td></td><td>The system saves the changes made by the admin.</td></tr>
<tr><th colspan="3">Alternate Course(s) of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>9</b></td><td>The log in details provided by the user are not correct.</td><td></td></tr>
<tr><td><b>10</b></td><td></td><td>The system displays an error message and asks the user to re-enter log in details.</td></tr>
<tr><td><b>11</b></td><td>The parameters provided for modifying the items are inappropriate.</td><td></td></tr>
<tr><td><b>12</b></td><td></td><td>The system displays an error message and asks the user to re-enter the parameters.</td></tr>
<tr><td><b>13</b></td><td>The user cancels the action.</td><td></td></tr>
<tr><td><b>14</b></td><td></td><td>The system cancels the modification action.</td></tr>
</table>

Identifier	UC-9
Name	Add new item
Summary	The admin provides details of a new item and this item is added to the system.
Priority	High
Actors	Admin

Pre-condition(s)	System is waiting for instructions.
Post-condition(s)	Changes in the items are saved and the system waits for further instructions.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The admin provides his login details.	
2		The system checks these details and logs the admin in.
3	The admin selects to add a new item.	
4		The system asks the admin the details of the item to be added.
5	The admin enters these details.	
6		The system checks these details and asks the admin to confirm the action.
7	The admin confirms the action.	
8		The system saves the changes made by the admin.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
9	The log in details provided by the user are not correct.	
10		The system displays an error message and asks the user to re-enter log in details.
11	The parameters provided for the action are inappropriate.	
12		The system displays an error message and asks the user to re-enter the parameters.
13	The user cancels the action.	
14		The system cancels the addition action.

Identifier	UC-10
Name	Delete Item
Summary	The admin can delete an existing item from the system.
Priority	High
Actors	Admin
Pre-condition(s)	System is waiting for instructions.
Post-condition(s)	Changes in the items are saved and the system waits for further instructions.
Typical Course of Action	

S#	Actor Action	System Response
1	The admin provides his login details.	
2		The system checks these details and logs the admin in.
3	The admin selects to delete the existing item.	
4		The system asks the admin the id of the item.
5	The admin enters the id.	
6		The system checks the id and asks the admin to confirm the action.
7	The admin confirms the action.	
8		The system deletes that item.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
9	The log in details provided by the user are not correct.	
10		The system displays an error message and asks the user to re-enter log in details.
11	The id provided for the action are inappropriate.	
12		The system displays an error message and asks the user to re-enter the id.
13	The user cancels the action.	
14		The system cancels the addition action.

Identifier	UC-11	
Name	Add stock of existing item	
Summary	The admin provides the details of the item whose stock is to be added and the system updates its stock.	
Priority	High	
Actors	Admin	
Pre-condition(s)	System is waiting for instructions.	
Post-condition(s)	Changes in the items are saved and the system waits for further instructions.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The admin provides his login details.	

2		The system checks these details and logs the admin in.
3	The admin selects to add the stock of an existing item.	
4		The system asks the admin the id of the item and the stock to be added.
5	The admin enters these details.	
6		The system checks these details and asks the admin to confirm the action.
7	The admin confirms the action.	
8		The system saves the changes made by the admin.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
9	The log in details provided by the user are not correct.	
10		The system displays an error message and asks the user to re-enter log in details.
11	The parameters provided for the action are inappropriate.	
		The system displays an error message and asks the user to re-enter the parameters.
12	The user cancels the action.	
13		The system cancels the addition action.

Identifier	UC-12	
Name	Modify Category	
Summary	The admin can modify the product categories i.e. new categories can be added and existing categories can be deleted.	
Priority	High	
Actors	Admin	
Pre-condition(s)	System is waiting for instructions.	
Post-condition(s)	Changes made in the categories are saved and the system waits for further instructions.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The admin provides his login details.	

2		The system checks these details and logs the admin in.
3	The admin selects to add or delete a category.	
4		The system asks for the relevant details.
5	The admin provides these details.	
6		The system checks these details and asks the admin to confirm the action.
7	The admin confirms the action.	
8		The system saves the changes made by the admin.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
9	The log in details provided by the user are not correct.	
10		The system displays an error message and asks the user to re-enter log in details.
11	The parameters provided for modifying the categories are inappropriate.	
12		The system displays an error message and asks the user to re-enter the parameters.
13	The user cancels the action.	
14		The system cancels the modification action.

Identifier	UC-13
Name	Add Category
Summary	The admin can add new categories in the system.
Priority	High
Actors	Admin
Pre-condition(s)	System is waiting for instructions.
Post-condition(s)	Changes made in the categories are saved and the system waits for further instructions.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The admin provides his login details.	

2		The system checks these details and logs the admin in.
3	The admin selects to add a new category.	
4		The system asks for the name of new category.
5	The admin provides the name.	
6		The system checks if the name does not already exist and then asks the admin to confirm the action.
7	The admin confirms the action.	
8		The system saves the changes made by the admin.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
9	The log in details provided by the user are not correct.	
10		The system displays an error message and asks the user to re-enter log in details.
11	The name provided for adding the category is already present.	
12		The system displays an error message and asks the user to re-enter the name.
13	The user cancels the action.	
14		The system cancels the add category action.

Identifier	UC-14	
Name	Delete Category	
Summary	The admin can delete an existing category from the system.	
Priority	High	
Actors	Admin	
Pre-condition(s)	System is waiting for instructions.	
Post-condition(s)	Changes made in the categories are saved and the system waits for further instructions.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The admin provides his login details.	
2		The system checks these details and logs the admin in.

3	The admin selects to delete an existing category.	
4		The system asks for the name of category to be deleted.
5	The admin provides the name.	
6		The system checks if the name exists and then asks the admin to confirm the action.
7	The admin confirms the action.	
8		The system deletes the category and all items present in that category.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
9	The log in details provided by the user are not correct.	
10		The system displays an error message and asks the user to re-enter log in details.
11	The name provided for deleting the category is not present.	
12		The system displays an error message and asks the user to re-enter the name.
13	The user cancels the action.	
14		The system cancels the deletion of the category.

Identifier	UC-15	
Name	Process order	
Summary	The customer arrives at the counter with the order of the items to be purchased. The cashier calculates the bill and the customer make payment for his/her order.	
Priority	Medium	
Actors	Customer, Cashier	
Pre-condition(s)	The system is waiting for instructions.	
Post-condition(s)	The system processes the order, updates stock and starts waiting for instructions.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The customer reaches the counter.	
2		The system is waiting for instructions.

3	The cashier selects the option to process a new order.	
4		The system asks the cashier the items to be purchased and their quantities.
5	The cashier enters these details as per the customer demands.	
6		The system checks if the required quantity of each item is less than or equal to the stock of that item available and asks the cashier to confirm the action.
7	The cashier confirms the action.	
8		The system generates the bill by adding the prices of all items purchased. The system checks if any discounts are available on any item. These discounts are applied (if any). The system checks if the customer is a member. If he/she is, additional 5% discount is applied.
9	The cashier asks the customer to make payments.	
10		The system is waiting for cashier to confirm order.
11	The customer makes payments.	
12		The system is waiting for cashier to confirm order.
13	The cashier confirms the order	
14		The system updates the stock of items purchased.
15	The cashier checks if any change is due.	
16		The system is waiting for new transaction.
17	The cashier gives the change to the customer (if any).	
18		The system is waiting for new transaction.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
19	The quantity to be purchased of any item entered by the cashier as per customer demand exceeds the available stock.	
20		The system displays an error message and asks the user to re-enter the quantity of the items.
21	The parameters provided for the action are inappropriate.	
22		The system displays an error message and asks the user to re-enter the parameters.

23	The cashier cancels the action.	
24		The system cancels the generate bill action.

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-16</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Modify member</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Admin can add or remove a staff member</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">Admin</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">In order to add a staff member admin should check he/she is not a staff member already as every member has a unique id.</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">After adding or removing a staff member, the list is updated.</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Modifies list i.e. add or delete a staff member</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>Adds or remove member if possible.</td></tr>
<tr><td><b>3</b></td><td></td><td>Updates list.</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3"><b>Alternate Course(s) of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>4</b></td><td>Adds a member with wrong id.</td><td></td></tr>
<tr><td><b>5</b></td><td></td><td>Displays error message “staff member id already exists.”</td></tr>
<tr><td><b>6</b></td><td>Deletes a member with wrong id.</td><td></td></tr>
<tr><td><b>7</b></td><td></td><td>Displays error message “staff member id does not exist.”</td></tr>
<tr><td><b>8</b></td><td></td><td></td></tr>
</table>

Identifier	UC-17
Name	Add member
Summary	Admin can add a staff member
Priority	High
Actors	Admin

Pre-condition(s)	In order to add a staff member admin should check he/she is not a staff member already as every member has a unique id.	
Post-condition(s)	After adding a staff member, the list is updated.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Adds a staff member	
2		System update list if possible.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
3	Adds a member with wrong id.	
4		Displays error message “staff member id already exists.”

Identifier	UC-18	
Name	Delete member	
Summary	Admin can remove a staff member	
Priority	High	
Actors	Admin	
Pre-condition(s)	In order to remove a staff member, he/she should be in list.	
Post-condition(s)	After removing a staff member, the list is updated.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Removes a staff member	
2		Removes member if possible.
3		Updates list.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
4	Enters wrong id to remove.	
5		Displays error message “staff member with such id does not exist.”

Identifier	UC-19

<table>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">Generate reports</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">Admin can generate reports on total sales or purchase of items in stock at any time</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Admin</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td colspan="2">Admin should have access to each and every sales and purchase record.</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td colspan="2">System waits for further instructions.</td></tr>
<tr><td colspan="4"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">Prompts system to give all records</td><td></td></tr>
<tr><td><b>2</b></td><td colspan="2"></td><td>Gives required information</td></tr>
<tr><td><b>3</b></td><td colspan="2">Generates a detailed report on sales or purchase.</td><td></td></tr>
</table>

<table>
<tr><td colspan="2"><b>Identifier</b></td><td colspan="2">UC-20</td></tr>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">Generate purchase reports</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">Admin can generate reports on total sales of items in stock at any time</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Admin</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td colspan="2">Admin should have access to each and every purchase record.</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td colspan="2">System waits for further instructions.</td></tr>
<tr><td colspan="4"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">Prompts system to give purchase records</td><td></td></tr>
<tr><td><b>2</b></td><td colspan="2"></td><td>Gives required information</td></tr>
<tr><td><b>3</b></td><td colspan="2">Generates a detailed report on sales.</td><td></td></tr>
</table>

Identifier	UC-21
Name	Generate sales reports
Summary	Admin can generate reports on total sales of items in stock at any time
Priority	High

<table>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Admin</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td colspan="2">Admin should have access to each and every sales record.</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td colspan="2">System waits for further instructions.</td></tr>
<tr><td colspan="4"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">Asks system to give sales records.</td><td></td></tr>
<tr><td><b>2</b></td><td colspan="2"></td><td>Gives required information</td></tr>
<tr><td><b>3</b></td><td colspan="2">Generates a detailed report on sales.</td><td></td></tr>
</table>

<table>
<tr><td colspan="2"><b>Identifier</b></td><td colspan="2">UC-25</td></tr>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">Offer Discounts</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">Offering discounts to customers by minimizing the prices of items in stock.</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Admin</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td colspan="2">Item should be in stock.</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td colspan="2">Stock is updated.</td></tr>
<tr><td colspan="4"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td colspan="2">Searches for the item.</td><td></td></tr>
<tr><td><b>2</b></td><td colspan="2"></td><td>Prompts if available.</td></tr>
<tr><td><b>3</b></td><td colspan="2">Changes price.</td><td></td></tr>
<tr><td colspan="4"><b>Alternate Course(s) of Action</b></td></tr>
<tr><td><b>S#</b></td><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>4</b></td><td colspan="2">Searches for the item not in list.</td><td></td></tr>
<tr><td><b>5</b></td><td colspan="2"></td><td>Prompts “Not available”.</td></tr>
</table>