

Sheet1

ID	Title	STORY	ACCEPTANCE CRITERIA	TASKS	TIME ESTIMATE	STATUS
1	Update product quantity	As a cashier I want to remove a product item from the inventory when I sell it so that we can keep count of how many items we have left in the warehouse.	<p>The program should have a list of products with their item quantities. A product could have a name, id, and item quantity. (Prices will not be saved here because they will be registered in a separate payment system.)</p> <p>The cashier can choose or search a product in the product pool when he or she sells one and then update the number of items left. The cashier needs to enter a receipt number when removing an item. The first version of the program could accept one item per receipt. (If there's time to refine this, several items per receipt could be updated.)</p> <p>When removing an item, a transaction should be save stating which product item was removed and the receipt number.</p> <p>This function could be called 'remove item'. The last item of a product cannot be sold.</p>	<ul style="list-style-type: none"> - create Product class - create ProductPool class - create method for viewing ProductPool (or make it MVC) - create a method for choosing a product from the ProductPool - create method for updating item quantity - ask for receipt number (- later add check to see that receipt isn't already registered) - check that the last item isn't sold - create a Cashier user role class that can call a method to remove an item - create a text file with some products in it - create a FileHandler class that can read the product pool from file - create a Transaction class that has a transaction type (enum), product, id and quantity - create a transaction when an item is removed - save transaction in a transaction pool 	2 days	done (version with one item per receipt)

Sheet1

2	Check which products are soon out of stock	As a cashier I want to check which products are soon going to be out of stock so that I can place an order on them.	The program should present the cashier with the option of checking which products have only few items left. Then it should show a list of those products.	<ul style="list-style-type: none"> - create a method in Cashier for viewing 'soon out of stock' products - get a list of products with less than 6 items left from the product pool - send list of products to the class that displays lists of InventoryDataItems 	1 day	done
3	Place an order	As a cashier I want to place an order on a product that is going to be out of stock so that we won't run out of products to sell.	The cashier can choose a product from the list and choose to place an order on it. Then the program probably asks for a quantity and then generates an order and saves it in a list of orders.	<ul style="list-style-type: none"> - create a 'processing behavior' that allows the cashier to create an order - ask how many items should be ordered - Create a class called Order - save product and ordered quantity in an Order - create an OrderPool class that keeps all orders - save Order in OrderPool - before creating a new order check if product is already ordered - if already ordered ask cashier if a new order should be made anyway - create a Transaction with the type 'order': save product and ordered quantity 	1 day	done

Sheet1

4	Return item	As a cashier I want to be able to return an item so that it can be registered again in the product pool.	A cashier can search for a transaction by receipt number and delete it from the system. After that the item will be placed back into the product pool	<ul style="list-style-type: none"> - create a method for searching for a transaction by receipt number in TransactionPool - create a method for deleting a transaction - check with Cashier that the product is correct and ask for confirmation - return the item to the Product Pool - delete the item's REMOVAL transaction from the TransactionPool 	1 day	done
---	-------------	----------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------	------

Sheet1

5	Review order list	As a manager I want to review the order list so that I can either cancel or confirm the orders.	A manager can see a list of all the orders created by cashiers. Then the manager can select one order from the list and choose to either confirm or cancel it. A canceled order is removed. A confirmed order is marked as confirmed. A third action could be to mark an order as arrived when the ordered products are received. Then the product quantity could be updated and the order deleted.	<ul style="list-style-type: none"> - create a class called OrderPool for saving the list of orders - create e method in Manager class for reviewing the list of orders - get the list of orders from OrderPool - send list to the class that displays InventoryDataItems - create a 'processing behavior' that allows manager to confirm or reject an order from the list - a rejected order is deleted from the list - a confirmed order's status should be set as confirmed. - an arrived order's product quantity should be added to product pool - create methods for saving orders to and reading them from file 	1,5 day	done
---	-------------------	-------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------	------

Sheet1

6	Several cashiers at the same time	As a cashier I want to be able to use the system at the same time with my colleagues so that we can work efficiently.	Two or more cashiers should be able to perform operations on the products concurrently. The program should keep the data up to date and there shouldn't be clashes occurring if two cashiers manipulate same data or file at the same time.	<p>SPIKE:</p> <ul style="list-style-type: none"> - implement a Thread that locks the product pool file for writing operations <p>SPIKE:</p> <ul style="list-style-type: none"> - Implement the Subject - Observer design pattern with the ProductPool as subject and the FileHandler as Observer - make the ProductPool notify the FileHandler each time changes are made in the ProductPool - when the FileHandler is notified, save the current state of ProductPool to file <p>SPIKE:</p> <ul style="list-style-type: none"> - create also a mechanism that works the other way, to get the latest changes in file back to the pool - try to do this with a FileWatcher having a WatchService and daemon thread running in the background - if FileWatcher doesn't work out, update pool from file every time it is shown with a call to the FileHandler 	2 days	<p>Observer - Subject pattern: done</p> <p>FileWatcher: in progress</p> <p>Pool update before display: done</p>
---	-----------------------------------	-----------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------	-----------------------------------------------------------------------------------------------------------------

Sheet1

7	Save order details	As a manager I want the details of a confirmed order to be saved so that we can track our orders.	When an order is confirmed, it shouldn't disappear from the system. The manager should be able to come back to the confirmed orders and check the details. Once the order arrives the manager should also be able to check the order as arrived and update the ProductPool with the newly arrived items.	<ul style="list-style-type: none"> - save all orders to file - create a FileHandler for OrderPool - implement Subject – Observer pattern also for the OrderPool and FileHandler - display Order status in the View of OrderPool so that manager can see which ones are confirmed - create a method that allows the manager to check an order as arrived and save items in ProductPool 	1 day	done
8	Access everything	As an admin I want to be able to access everything in the system so that I can perform administrative tasks on the data.	The admin should have access to all the inventory data. The admin should be able to view the different data.	<ul style="list-style-type: none"> - create an admin class - create a 'processing behavior' that displays an InventoryDataItem - create methods that allow the admin to view items from the ProductPool, TransactionPool and OrderPool 	1 day	done