

Clothing Store Point of Sale System

Group 5

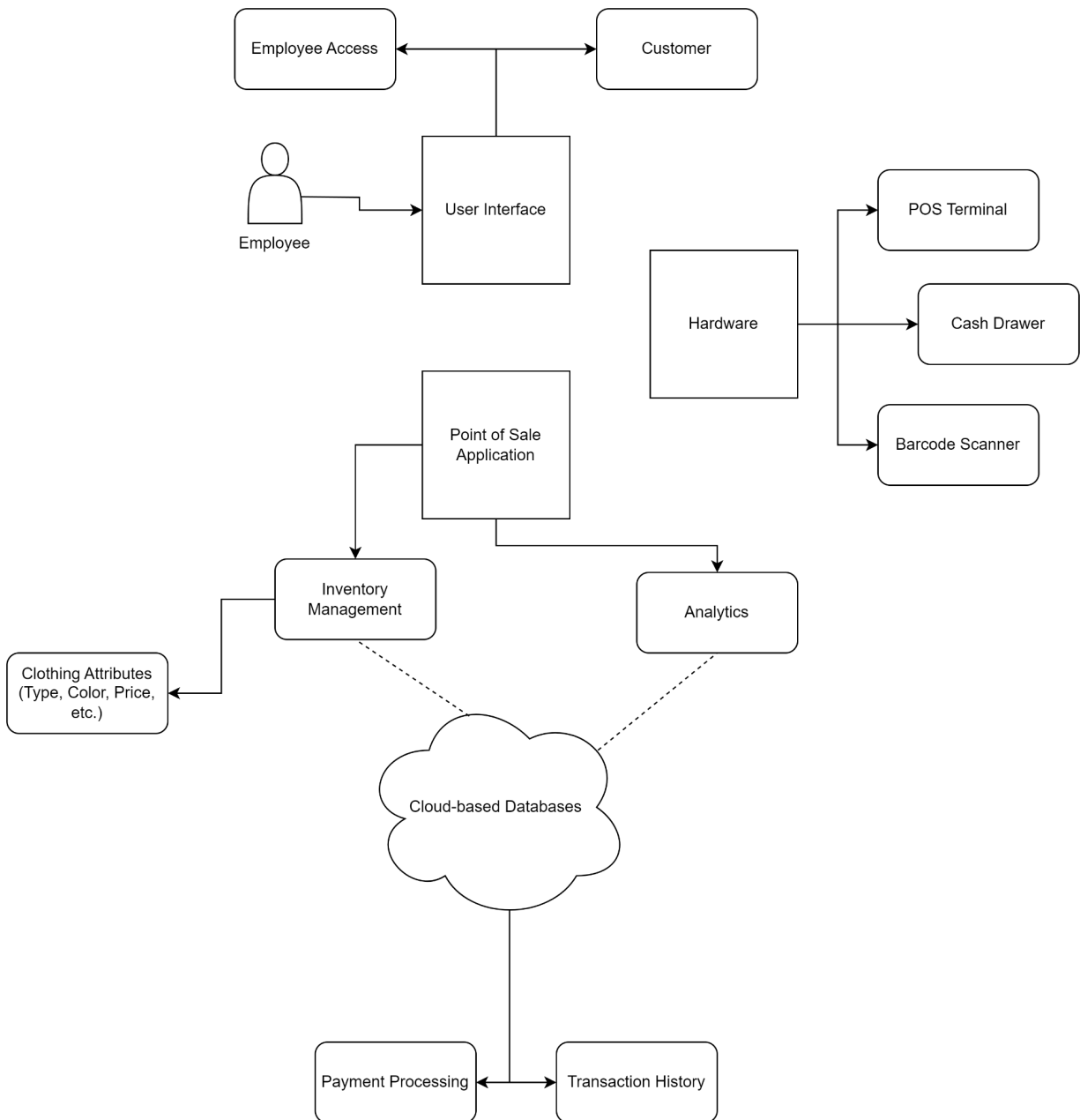
Miguel Melo Ochoa
Jose Hernandez Sanchez
Gabriel Sawaya

System Description

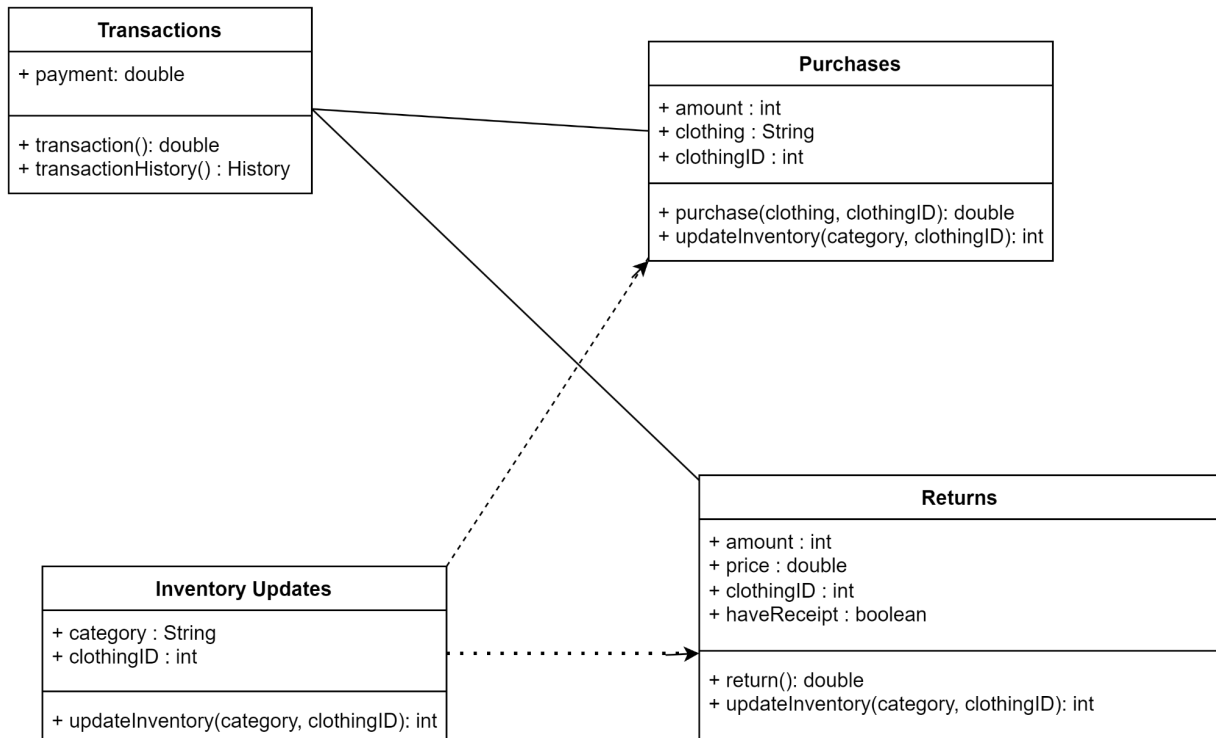
The Point of Sale (POS) system is designed to allow employees to accept payments, process transactions, and keep track of inventory throughout the store. The system is aimed at making sales efficient for employees, keeping items located in the designated store organized through inventory updates, and handling transactions such as purchases and returns. The POS system will also aid in keeping track of sales through analytical reports which can be accessed by higher-ranked employees. Employees will be able to scan clothing items via barcode scanner, once scanned, the item will be added towards the customers total transaction. Once all items are scanned, the employee can then process the payment via add-on card reader then can accept credit, debit, or cash, which will utilize a register to store/take out change. Once items have been scanned and purchased, the inventory of those items will be adjusted accordingly. Returns will work similarly, with the customer's returned item(s) being scanned as a return, payment being refunded, and inventory being updated.

Software Architecture Overview

Architectural diagram of all major components



UML Class Diagram



Description of classes

- **Transactions**
 - This class will be in charge of covering the transactions of purchase. Should be able to see transaction history. It should be able to access the Purchases and Returns class
- **Inventory Updates**
 - This class will cover the inventory update for the shop. This class will allow the employee to update the inventory. Will print a list of all the items in inventory. It will also be able to print out the amount of one item. Should have an add and delete function.
- **Purchases**
 - This will administer the purchase process. This includes processing payment, removing the purchased item from inventory, and processing the cost.
- **Returns**
 - This will administer the returns. Will process the return of payment and add the returned item back on the inventory.

Description of attributes

- Transactions:
 - payment : double, payment is the money used to purchase clothing which is why its type double.
- Inventory Updates:
 - -category : String, category is what the type of clothing is associated with.
 - -clothingID : int, clothingID is the ID/tag of the clothing to know what and which clothing it is.
- Purchases:
 - -amount : int, amount is how many purchases of clothing there are.
 - -clothing : String, clothing is the type of clothing(e.g.shirt,pants,jacket).
 - -clothingID : int, clothingID is the ID/tag of the clothing to know what and which clothing it is.
- Returns:
 - -amount : int, amount is how many pieces of clothing were purchased.
 - -price : double, price is the price of each clothing item.
 - -clothingID : int, clothingID is the ID/tag of the clothing to know what and which clothing it is.
 - -haveReceipt : boolean, haveReceipt checks if the customer has the receipt to show proof of purchasing the item.

Description of operations

- Transactions:
 - transaction(): double - Function used to process a transaction. When a transaction is complete, returns money used for the transaction (i.e. bill/receipt).
 - transactionHistory() - Stores transaction result in transaction history
- Inventory Updates:
 - updateInventory(category:String, clothingID:int): int - This function will take in the item's attribute(category) and ID as parameters and should return a count of the number of those item(s) that are left in inventory.
- Purchases:
 - purchase(clothing:String, clothingID:int): double - This function will take in the item's attribute(category) and ID as parameters and should return the total price of all item's as a double.
 - updateInventory(category:String, clothingID:int): int - This function will take in the item's attribute(category) and ID as parameters and should return a count of the number of those item(s) that are left in inventory.

- Returns:
 - return(): double - This function returns the dollar amount returned to the customer after a return
 - updateInventory(category:String, clothingID:int): int - This function will take in the item's attribute(category) and ID as parameters and should return a count of the number of those item(s) that are left in inventory.

Development plan and timeline

Partitioning of Tasks

1. Gather Requirements
 - a. First we speak with the client to obtain as much information about the system as possible. We want to collect all possible information, including functional and non-functional requirements for the proposed system. Not only would we need to obtain software related information, but also hardware requirements too. Thus an on-site meeting would be best in order to cover as much space as possible before we develop the system and know if we have any physical limitations produced by the location of the store or client. This phase of the project is estimated to take about two weeks.
2. Design/Develop the Software
 - a. Next we would begin designing the system based on what we have gathered from the client. After finalizing the architecture of the system, we would develop the software and set up the necessary databases. We would also create safeguards to protect the databases from intrusions. This could take a month to a few months of work depending on the size of the team we are able to contract.
3. Implementation
 - a. Once we have identified all core functionalities of the system, we will begin development of these functionalities. We will incorporate all of our design decisions, software implementations, and code structure into a fully functional system prototype, which will then be used for testing. This phase must take substantial time, as we want to make sure we are as thorough as possible with our code to ensure testing goes with minimal risks/failures to the system.
4. Verify & Validate the Software
 - a. After the prototype has been developed, we will begin testing the product in its most practical use. We would execute all of our tests, including unit, integration,

and system testing. We will also conduct our user acceptance testing and gather feedback in order to refine the system. This phase of the project will take two weeks.

5. Maintenance/Other

- a. The system will be deployed for client use. The client will be able to obtain first-hand experience with the system and let us know if we have met their software expectations. We will also monitor the system after its initial rollout to make sure that there are no issues with the system. If the client would like us to continue working on the software then we could continue working with them and address any other concerns that they may have.

Team Member Responsibilities

Jose - Design/Develop the Software, implementation

Miguel - Implementation, Maintenance/Other

Gabriel - Gathering requirements, implementation, verify and validate the software