Clothing Store Point of Sale System

Team Members: Mark Canilang, Mirek Kupczynski Jr, Angel Guzman

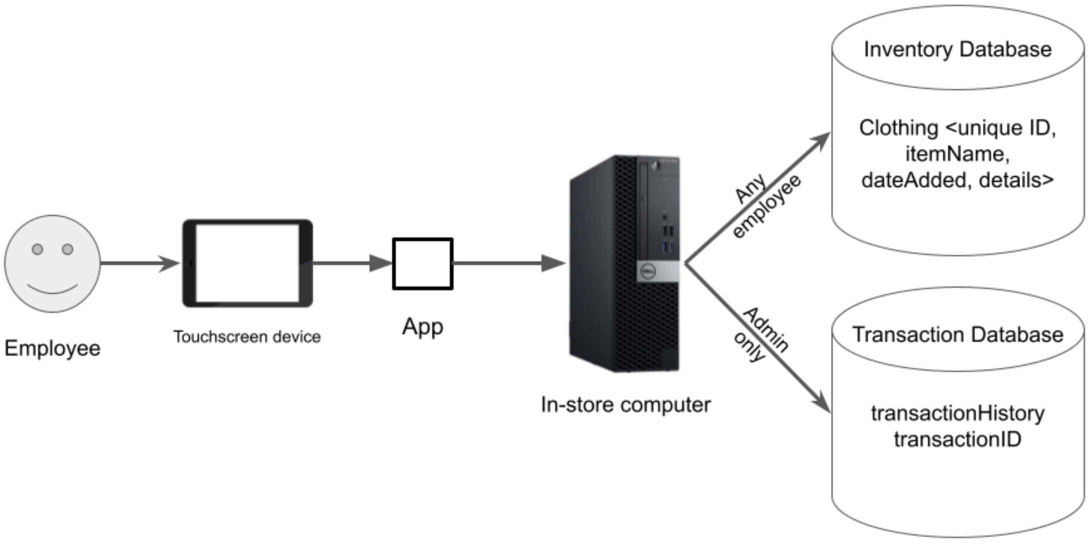
System Description

The system that is being developed here is what is referred to as a point of sale system. This type of system is a computer-based system that helps workers of a clothing store handle transactions and manage inventory, as well as assist management with keeping track of inventory and sales across different stores of the same company.

This system will help streamline many tasks that the workers of stores do frequently day to day. It will make their jobs easier, thus allowing for higher efficiency work to be done, as well as to increase customer satisfaction by giving them a smoother shopping experience.

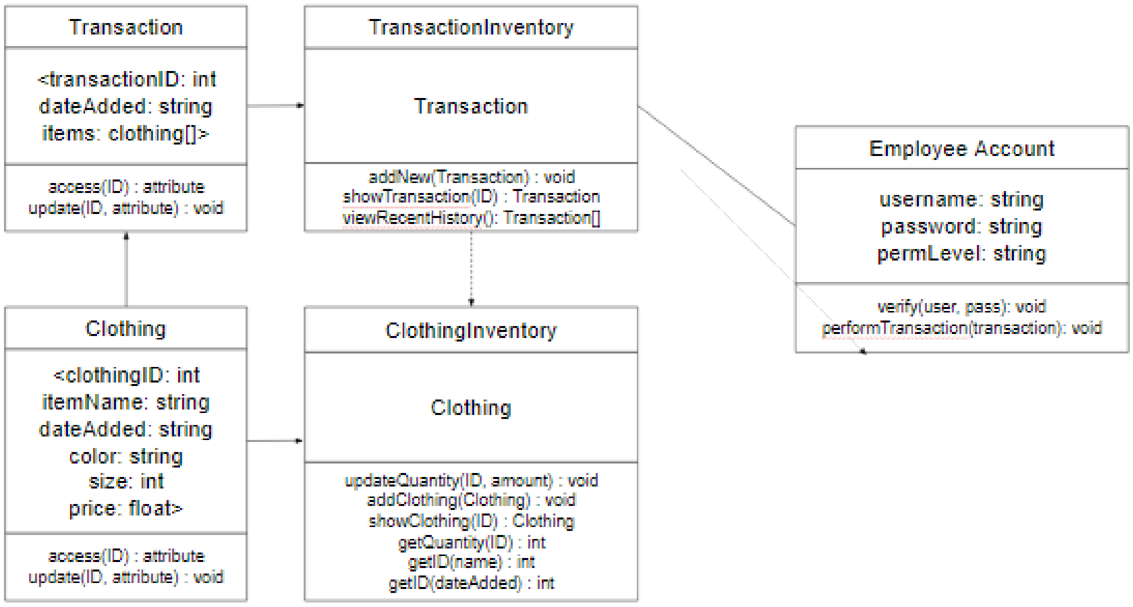
Software Architecture Overview:

Architectural Diagram Overview:



This diagram illustrates how all major components interact with each other. Employees will interface with the system using the in-store touch screen systems (mainly Apple and Android devices), all of which have an app that acts as the inventory/transaction software. This app will connect to the store's main computer that is accessing the relevant information from the cloud databases. There are two separate databases, one for the inventory, which all employee users can access, the other will be the transaction database that can only be viewed by administrative users.

UML Class Diagram:



This diagram illustrates how the primary data structures and classes of this system will operate and interact with each other. There are three primary classes, clothing and transactions are related to the databases, and these will be accessed by the database files, which are accessed by the employee account class.

The clothing classes consists of basic information about individual clothing items, such as:

- unique numerical id (integer)
- the date it was added to the system (string)
- the name of the item (string)
- basic information about the clothing, such as color (string), size (int), price (float),

(Note: different colored versions and different sizes of the same item are considered different, they will share identical information except for the total quantity.)

Next we have the transaction class, which contains its own numerical id, the date of the transaction, and an array of clothing items. The transaction, when being displayed, also shows the total price calculated from its constituent clothing items.

Lastly we have the Employee Account class, which stores expected information such as the username and password (both strings), and an access level, which can range from ordinary employee to administrator.

All three of these classes contain the appropriate getters and setters, which have been generalized to simplify the diagram. The databases mentioned above utilize files that contain instances of the two storage classes, with appropriate methods to access and add new items.

We also have two inventories listed here: ClothingInventory and TransactionInventory. TransactionInventory will be the one modified by Employee Account as only logged in employees will be able to both add transactions, view transactions, and view the recent history after logging in and having the correct permissions. Furthermore, transaction inventory will contain the clothing inventory and modify it when corresponding transactions are added or refunded.

To perform these operations:

- addNew(Transaction) will just add a new transaction.
- showTransaction(ID) show prior transactions corresponding to the given ID.
- viewRecentHistory() will display the transactions as a list.

Each of these are associated with the clothing inventory operations:

- updateQuantity(ID, amount) when a transaction is performed.
- addClothing(Clothing) if new clothing needs to be added
- showClothing(ID) to get information about an already added clothing.
- getQuantity(ID) to see the stock.
- getID(name) to view clothing corresponding to an ID.
- getID(dateAdded) to get an ID based on what day it was added.

Development Plan and Timeline:

1- Develop main clothing system (2 weeks) : Mark & Mirek

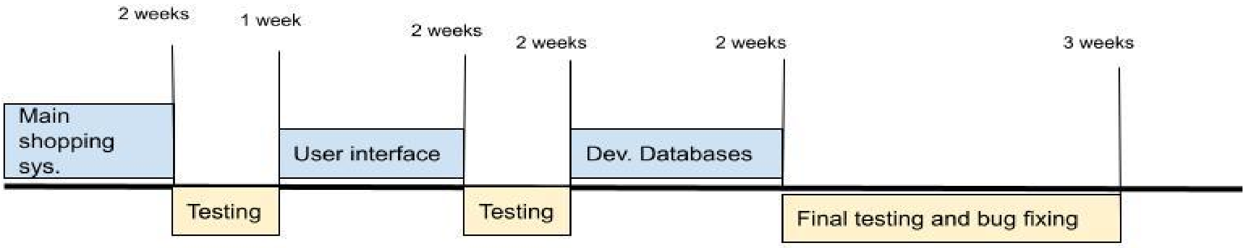
2- Testing main clothing system (1 week) : Mark & Mirek

2- Develop User interface (2 weeks) : Angel & Mark

3- Testing and bug fixing (2 weeks) : Mark & Angel & Mirek

4- Develop databases (2 weeks) : Angel & Mirek

5- Final testing (3 weeks) : Angel & Mark & Mirek


2 weeks
1 week
2 weeks
2 weeks
2 weeks
3 weeks
Main shopping sys.
Testing
User interface
Testing
Dev. Databases
Final testing and bug fixing