

Clothing Store Point of Sale System

Cariza Shine Sta Maria
Juan Alberto Loza-Franco
Tong Yang

System Description

The Clothing Store Point of Sale System is designed to process customer transactions and maintain an accurate inventory.

SWA Diagram

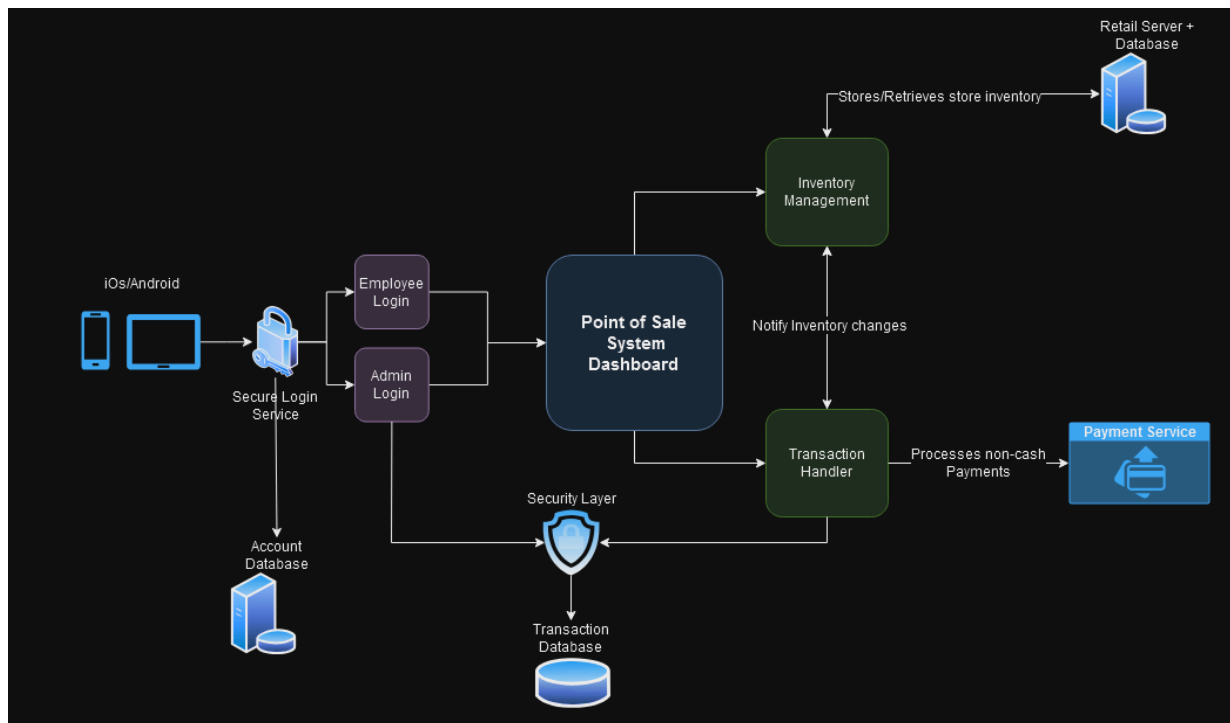
The software system's user interface is an application on small devices such as phones or tablets, supporting iOS or Android. A secure login service is implemented to allow access to only employees and administrators, prompting the user to enter a unique, self-made username and password. This account information is stored inside its own separate database, along with an employee's name and identification number. Once a valid account is logged in, the Point of Sale System Dashboard is accessed, which allows the user to either manage the store's current inventory or handle customer transactions.

Regarding inventory management, our second database is implemented to store information about products. Both employees and administrators have the ability to register new products and update current product information such as quantity and descriptions. The user can also receive information from our database to check what products are available in store.

Regarding transactions, this is also accessible to both employees and administrators. For customers using credit cards or debit cards, a third party payment service will take over and process transactions. Cash transactions are handled in person.

Both aspects are connected as the inventory management depends on both the user and transactions. Regardless of it being a transaction or a refund, this automatically updates a product's quantity and sends it to the database.

The Point of Sale system implements the third and final database, storing transaction history. Due to privacy concerns, an extra security layer will be added before accessing the actual database as it stores sensitive information like customer information. This will only be accessible to administrators.



UML Diagram

The Store class only holds attributes regarding store information. storeName holds an actual store name, and for scalability reasons, a storeId is also required to distinguish between different store locations. The Store class is associated with almost every class, except the EmployeeLogin class.

The Item class also holds attributes only regarding product information. Included are product descriptions such as itemType, color and size, an itemId attribute for searching a product precisely, and a price attribute. The Item class is associated with the Store and Inventory class.

Contrary to the Item and Store class, the Inventory class only holds operations. There are multiple overloaded search functions which are called based on user preference. Users can search based on a specific itemId, a description such as a color, size or actual product name, and the date a product was registered. The sort() function will sort products lexicographically based on its name. The count() function will return the number of products in stock at the specific store. The add(Item) function takes in an Item object, requiring users to fill out specifics of the item descriptions when adding a product to the database and the delete(int id) function removes a product from the inventory database, requiring an item identification number for precision. The updateItem(Item) function allows users to change product descriptions, working directly with the parameter. The Inventory class is associated with the Store class as inventories differ from each store, and with the Item class in order to perform operations on Item objects.

The TransactionHandler class has one attribute, the transactionID, which allows for manual input during transactions. Operations include a calculateTotal() function that adds up the price of all items, including sales tax. The confirmPurchase() method checks to see that the third party payment service or cash transaction is approved and went through properly. The issueRefund() allows for the refund process to take place. The updateInventory() function changes the quantity of products in stock at the store. For cash transactions, a changeBack() method is added to calculate how much cash is given back to the customer. This class is associated with the Store class and TransactionHistory class, as the transactional data will be sent to the database.

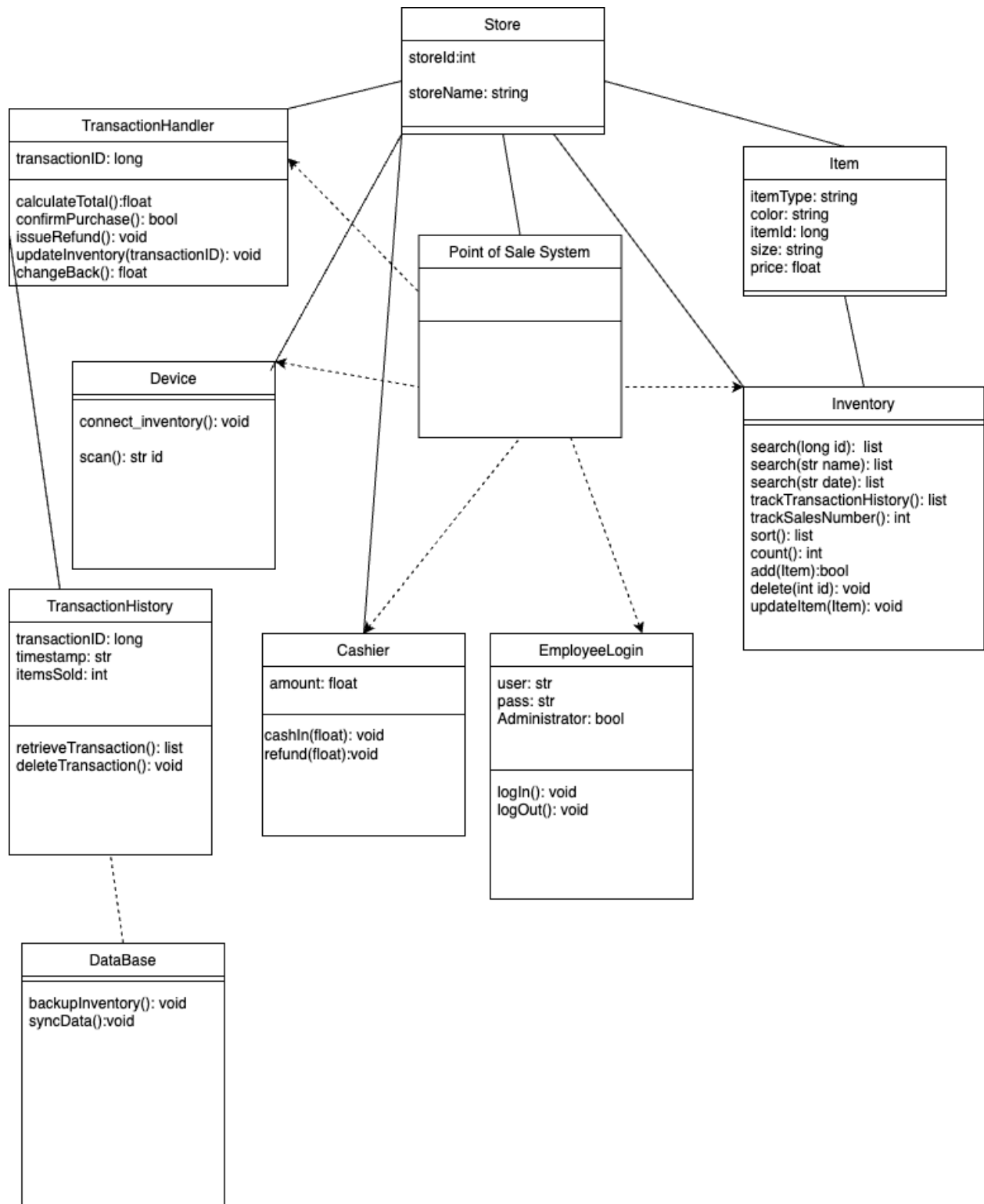
The Device class includes two operations. One operation is connect_inventory(), which allows the user to access the inventory. The scan() operation is a quick way to grab the item ID for transactions. This class is associated with the Store and Cashier class, as this is mainly related to the transaction aspect and useful for the Cashier class.

Our TransactionHistory class has both attributes and operations. There is a transactionID attribute that holds a unique number representing each transaction. The timestamp attribute gives a specific time and date the transaction took place, and itemsSold is the total quantity of the products bought. Our retrieveTransaction() operation, only accessible to the administrator, allows pulling up any transaction and deleteTransaction() removes a transaction from the history completely. This class is associated with the TransactionHandler class and dependent on the DataBase class. The DataBase class holds only operations, where backupInventory() keeps a copy of all inventory data and product information, and syncData() which allows inventory to be updated and precise.

Our Cashier class holds the attribute amount, which holds the total of the transaction. The two operations are cashIn(), which is for transactions paying with cash that takes in the total, and refund() for processing refunds. These both take in floats as arguments, as they work with price amounts.

The last EmployeeLogin class holds attributes for a username and associated password, and the administrator attribute which asks for an additional identification number to differentiate from the employee. The two operations, logIn() and logOut(), allows the user to login and access the Point of Sale System, and log out when finished.

The Point of Sale System class is dependent on the Inventory class as this is a big contributor to product and store organization, the TransactionHandler and Cashier class as this is the fundamental financial aspect of making the store sustainable, the Device class, as this is where you access the store inventory, and the EmployeeLogin class as this is the first prompted class users will encounter to access everything else.



Development Plan and Timeline

The Point of Sale System is expected to be fully functional within 10-12 months. Tasks are divided into features; inventory management, transaction processing, and additional hardware.

The task of designing inventory management will be taken up by Alberto. Alberto will implement a stable and reliable database to hold large amounts of product data, and will implement functions to keep it updated.

The task of designing transaction processing will be taken up by Tong. Tong will research and implement a secure third party system for credit and debit purchases. Functions he will implement will be responsible for proper calculation for cash purchases, and the refund process as well. Both Tong and Alberto will work together to ensure that transactions will properly update the inventory.

The task of designing additional hardware such as a barcode scanner will be taken up by Cariza Shine Sta Maria. Cariza is responsible for the user interface, making sure to develop and design the front-end application. Cariza is also responsible for implementing the secure login service as this is the gateway to sensitive information.