

# **POS Software Design Specification**

Completed by: Nick Russert, Christopher Ong, Hsa Moo

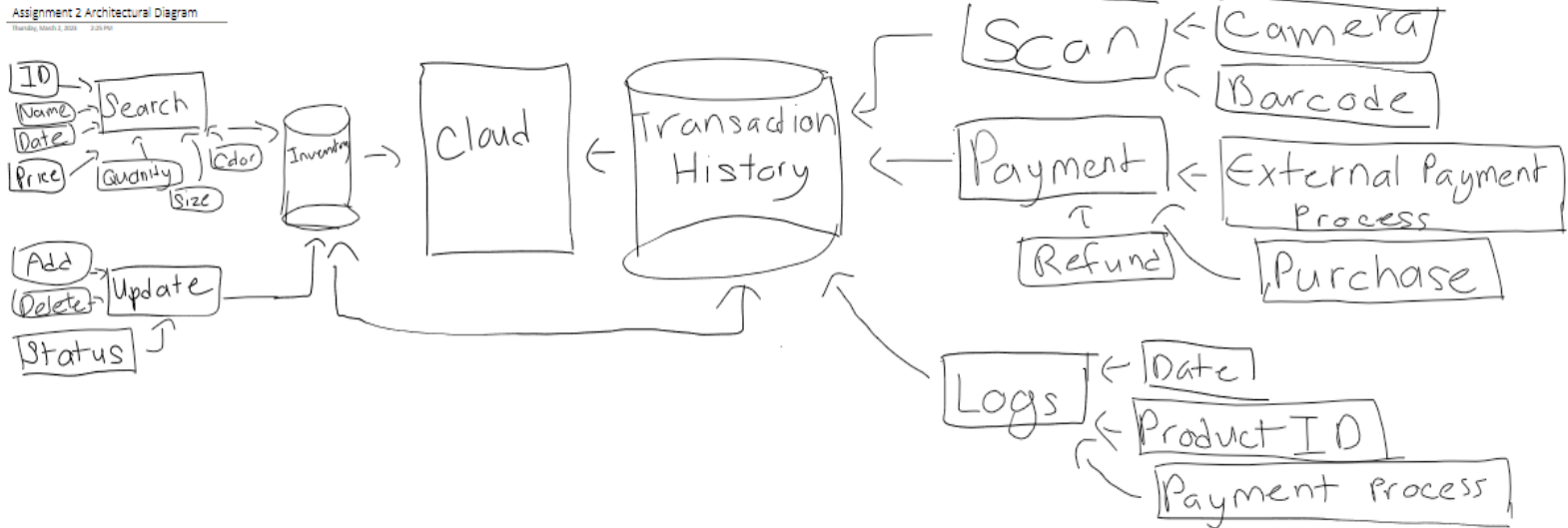
March 10, 2023

## **1 System Description**

This point of sales system will be designed to make daily sales efficient and organized for employee and manager use. It will enable employees and managers to update and view store inventory to see what is available in stock, as well as handle transactions which include purchases and returns. The POS system will accept credit cards, debit cards, and cash and will calculate the total of the transaction including tax. The system will track inventory and after a sale is made, the store inventory will be automatically updated. This system is intended for daily use by both store employees, as well as managers. Further, this document will specify in depth the user requirements, the system requirements, including transactions, refunds, inventory management, and payment histories, and other features needed in the implementation of the software.

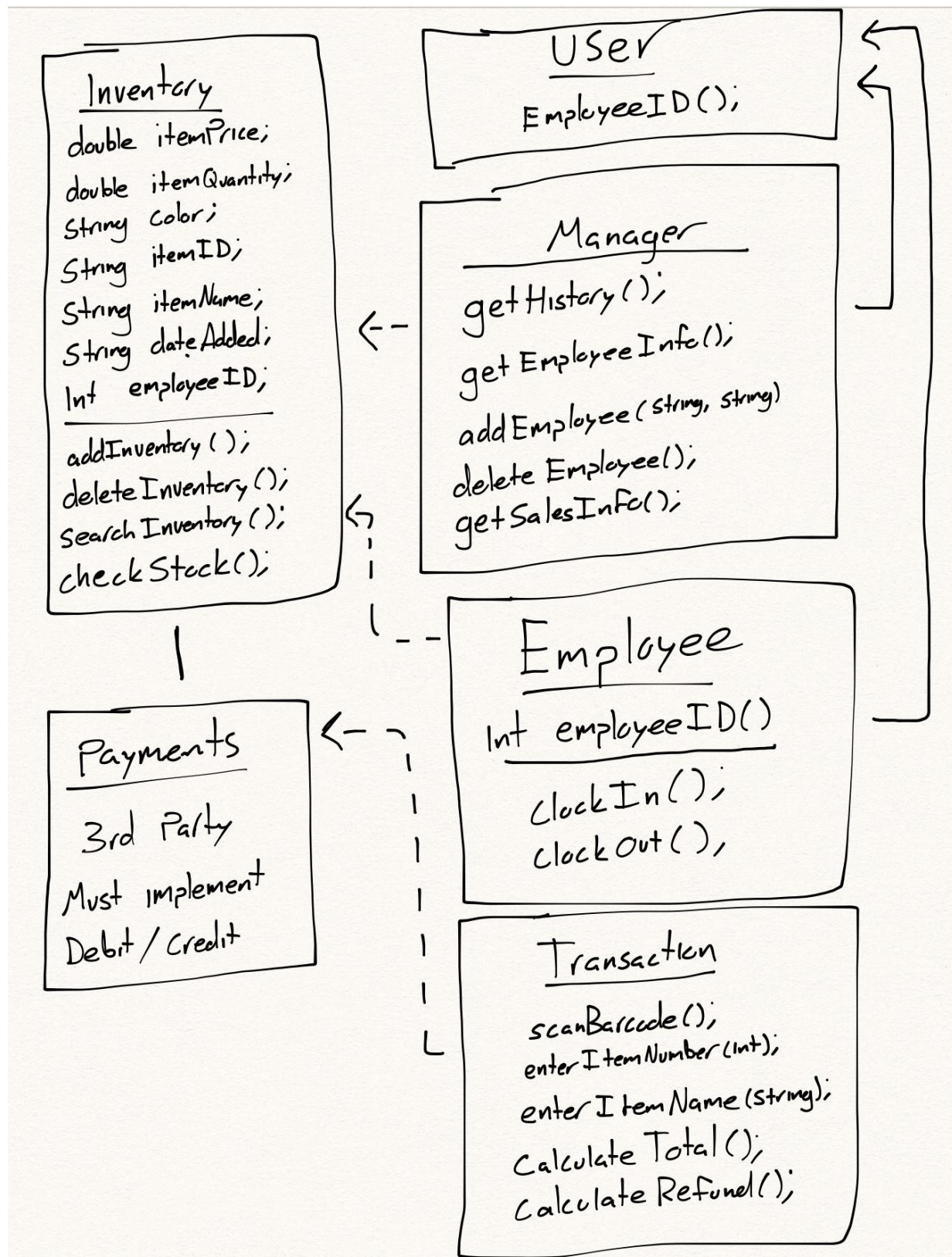
## 2 Software Architecture Overview

Architectural diagram of all major components:



Brief description of Architecture diagram: This architecture diagram consists of two major databases, which are the inventory and transaction history database. These two databases are different from each other, but they are both supported and stored in the cloud. The inventory database has access to methods in relation to search and updating, and is able to access information from the transaction history database. The transaction history database has access to methods in relation to scanning, payment or recording data logs, and is able to access information from the inventory database. For more information about these databases, methods and classes within this architecture diagram, refer to the description of classes section of this document.

# UML Class Diagram:



Brief description of UML class Diagram: One of the main components of the POS software is the inventory. The inventory class is used by almost every other class and holds a lot of data types. The main classes consist of the inventory, the user, manager, employee, transaction, and payments. For more information on these classes, data types, and methods see the description of classes section of this document.

**Description of classes:**

1. Inventory: The inventory class will be the main database that will store all the objects and functions that will be used by other classes. The objects stored inside the inventory class are:
  - a. itemPrice: The price of item
  - b. itemQuantity: The number of item
  - c. Color: The color of item
  - d. itemID: The item's id
  - e. itemName: The item's name
  - f. dateAdded: The date the item's was added
  - g. employeeID: The ID of employee

The function inside the inventory are:

- a. addInventory(): Function to add items into inventory
- b. deleteInventory(): Function to delete the item in inventory
- c. searchInventory(): Function to search for specific item in the inventory

- d. `checkStock()`: Function to check if item are in inventory and how many are left
- 2. Manager: The manager class will allow users who are manager to have authorization to view and make changes to the inventory class. The function manager class contain are:
  - a. `getHistory()`: Get the history of transactions and employee clock in and clock out time
  - b. `getEmployeeInfo()`: Get employee information from the inventory
  - c. `addEmployee()`: Add employee into inventory
  - d. `deleteEmployee()`: Delete employee from the inventory
  - e. `getSalesInfo()`: Get the sales information from inventory
- 3. Payments: The payments class allows 3rd parties to pay using debit/credit for items transferred from the transaction class and it will be stored into the inventory.
- 4. Transaction: The Transaction class will store items and functions that will be used to make payments:
  - a. `scanBarcode()`: Scans the items barcode and matches it itemID
  - b. `enterItemNumber()`: Allows the user to enter item number
  - c. `enterItemName()`: Allows the user to enter item name
  - d. `calculateTotal()`: Calculates the total cost of an item
  - e. `calculateRefund()`: Calculate the refund total of an item

5. Employee: The Employee class allows users to clock in and clock out using the clockIn() and clockOut() function along with an employeeID() to identify which employee is working or not today and when they are done with work.
6. User: The User class contains the employeeID() which will be used to identify which class between the manager and employee is using the system.

**Description of attributes:**

There are multiple different types of attributes this system should contain in order to fully operate to the best of its ability. The first attribute is that the user should be able to login into the system, whether it be using their own employee or manager login. This attribute is important since this helps keep track of who's using the system at all times and this attribute provides access for the user to be able to access this system.

Another attribute this system is required to have is the ability to update inventory and search through the inventory. The user should be able to add or remove items within the system to update the inventory of the store and should be able to search for certain products within the system, whether it be searching for item's price, the quantity of an item, the color of the item, or based off the item's ID or name. This is the most important attribute within the system since we want to ensure that we have enough products for the customers to purchase and order any more products we are low on. By having this attribute, this will help the users stay organized with their products and help keep customers satisfied.

The third attribute this system should have is the ability to be able to use a scanner to scan a product's designated barcode when a customer wants to purchase or refund an item and enter the product's name or number to purchase or refund an item. This attribute should be able to update the inventory of the system automatically if a customer decides to purchase or refund a product. This attribute should be also able to calculate the total cost of products purchased or refunded and be connected to an external third-party system to conduct payments. This is an important attribute since this will help the user keep an updated inventory and help conduct product transactions for the customers.

Lastly, there should be an attribute where the user is able to clock in and clock out for their shifts and the manager should be able to search the employee's info and history, be able to add or delete employees within the system, and search the total sales of the store or of certain products. These attributes are intended to help run the work system by being able to clock in and out, and aids managers in the ability to update the work roster, see what employees are doing, and help them see the total sales of products.

#### **Description of operation:**

Summed up, the POS system should be designed with the ability of the user in mind to utilize the system in handling transactions including purchases and refunds through credit, debit, or cash. The system must operate on either apple or android operating devices, including phones or tablets, utilizing a camera for scanning

barcodes. This section will go more in depth on the operations during payment processing, online inventory, security and storage, and user functions.

To begin, during a transaction, the system must have the ability to ring up items. Operations used to accomplish this must include the ability to enter a unique item number, scan an item's barcode, or manually enter an item ID, all of which is stored in the online inventory. The system must include an operation to automatically calculate the total price, including sales tax, and after a transaction, will automatically remove the item from the store's inventory. Credit and debit card operation will all be handled by an external payment processor but the two systems must operate in harmony with one another.

Inventory is one of the most important elements of the POS system and must operate efficiently and effectively. The inventory operates through an online system and is used as a tool for employees to see what is available in the store's stock. Functions and operations the inventory system includes are the ability to search for specific items and see whether or not the store has a particular item in stock, or if a neighboring store has the item. The search operation will require either an item ID, the item's name, or the sata it was added to the inventory. Employees will also operate the inventory by adding or removing items and assigning items with relative information such as the item's price, color, size, quantity, and ID number.

The system will also operate with the help of the cloud for memory and database storage and security. All transaction history and sales will be stored securely and



separately from inventory. The system will include a special manager operation that will allow accessing the secure data and for creating new employee accounts.

### **3 Development plan and timeline**

Partitioning of tasks:

In the first 2 month:

- Someone to supervise the project and make sure that the clothing store point of sale system is being created and is on time
- Someone to communicate with the client and update the team on what the client wants in the system

In the following months till end of project:

- Programmers that will write the code need for the system to look and function how the client wants it
- Someone to test if the system that is created is the same as what the client asked for

Team member responsibilities:

- Project Managers are team members responsible for overseeing the project, making sure that the team members are enough for the assigned tasks, and will be responsible for how the system will be created.

- Back-end developer are team member responsible for writing code to implement into the system that follows the UML class diagram
- Front-end developer are team members who are responsible for making the display of the website to what the clients want and so they are responsible for communicating with the clients and the back-end developers as well
- Quality tester will be responsible for following a series of test on the system and reporting the bugs to other team members