

Software Project Phase III

Keisi Breshanaj
Loard Bejko
Dejvi Kocilja
Ifigjenia Sopiqoti

ii. Software Architecture

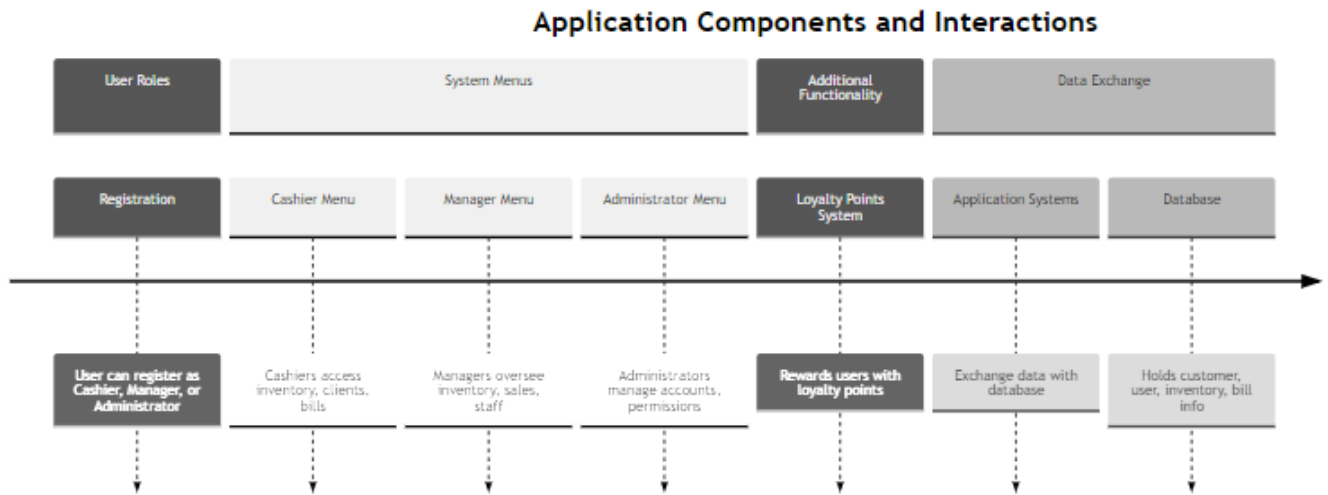
a. System Architecture:

As mentioned in the previous phases of this project, EasyMart software has 3 fundamental roles that the Cashier, Manager, and Administrator Modules perform. The Cashier module handles sales transactions and customer interactions; the Manager module is in charge of general operations and inventory control; and the Administrator module is in charge of user accounts and permissions. Depending on the type of user role that was selected during the registration process, each type of user will have their own distinct menus to perform different operations. These modules work well together, utilizing real-time database data to facilitate seamless collaboration and effective decision-making.

On top of that, a loyalty program is in place to reward the most loyal customers of the supermarket with points which can be converted into discounts on future purchases. The system tracks customer purchases and dynamically updates customer loyalty points during checkout. Moreover, the Loyalty Points System interfaces with the database, storing and retrieving customer loyalty point balances with precision. This ensures accurate tracking and seamless redemption of points, enhancing the overall customer experience and solidifying the supermarket's relationship with its clientele.

A strong database system functions in tandem with these components, acting as the central repository for crucial data such as product details, sales histories, user accounts, billing information etc. By facilitating smooth communication between modules, this database makes sure that data is consistent and of high quality throughout the entire system. The EasyMart Supermarket ERP software optimizes user interaction and system functionality with a layered architecture that includes Presentation (Front-End), Application, Business Logic, and Data Access Layers. This gives supermarket managers and staff the ability to make well-informed decisions and provide outstanding customer service.

b. Component Diagram:



ii.ii. Detailed design

a. Class diagram

EasyMart (main class)
cashiermodule: Cashier Module
managermodule: Manager Module
adminmodule: Administrator Module

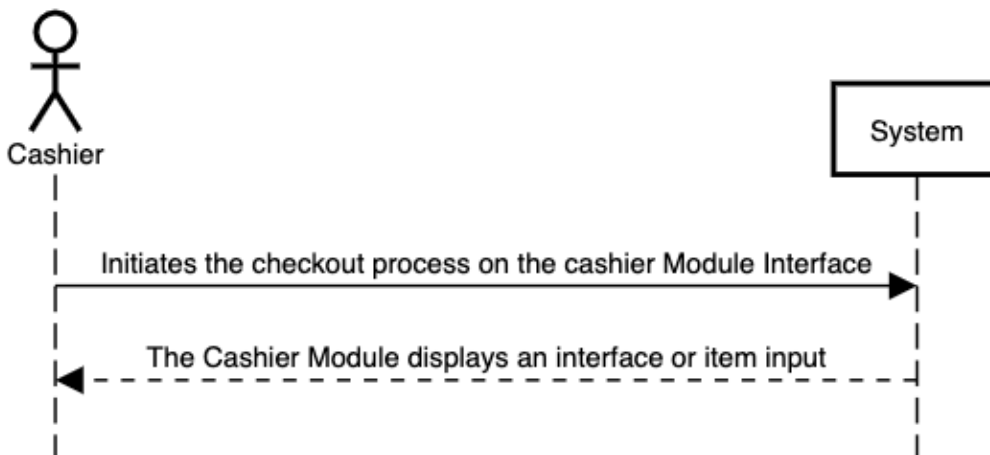
Cashier Module
processTransation()
applyLoyaltyPoints()

Manager Module
manageInventory()
monitorSales()
overseeStaff()

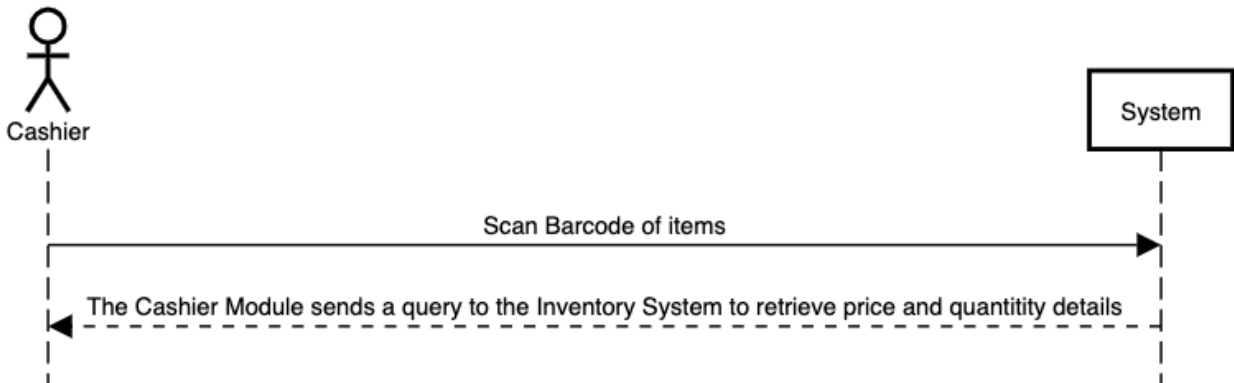
Administrator Module
manageUsers()
managePermissions()

b. Sequence Diagram

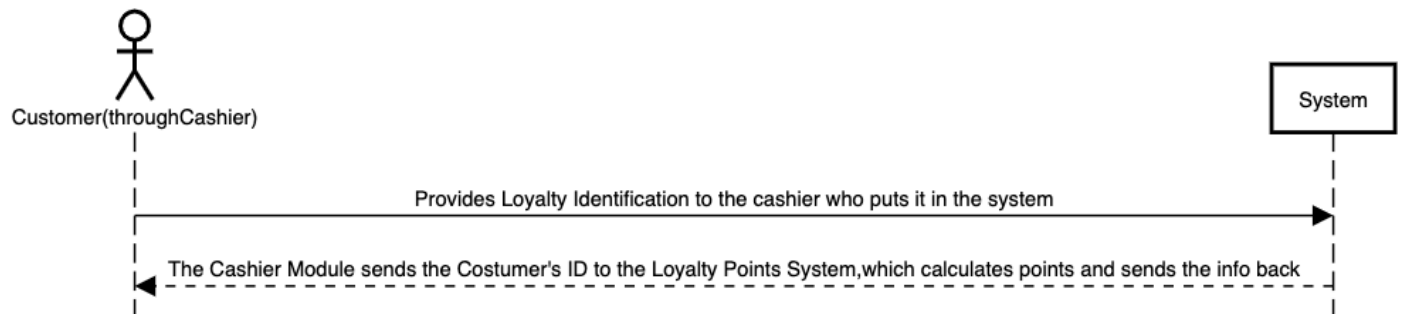
1. Start Transaction



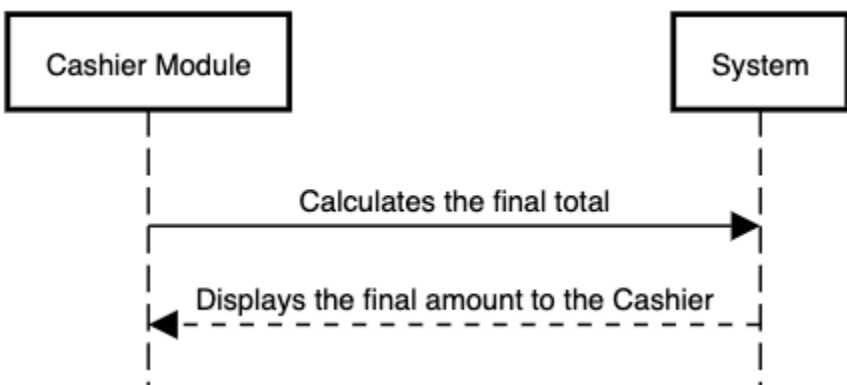
2. Scan Items



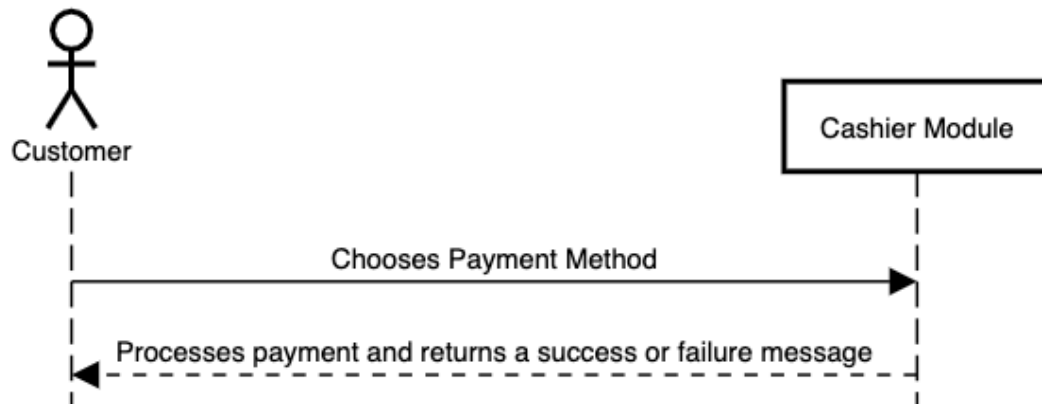
3. Identify Customer and apply Loyalty Points



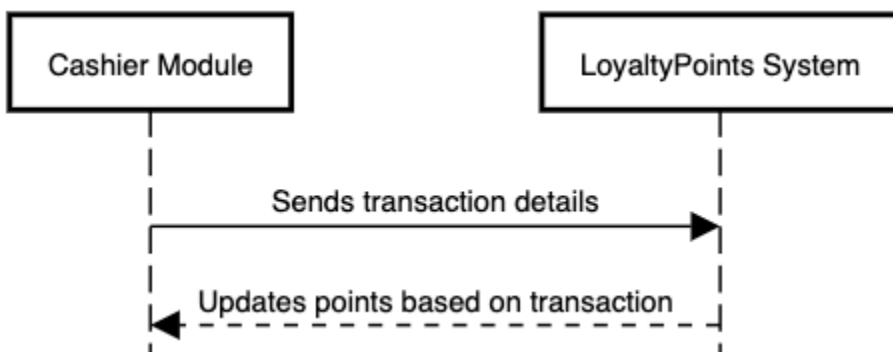
4. Finalize Total



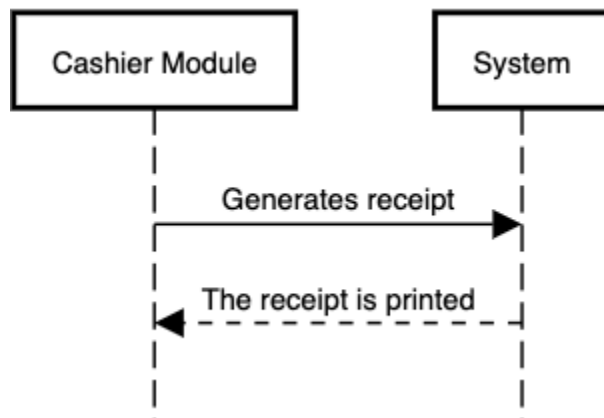
5. Process Payment



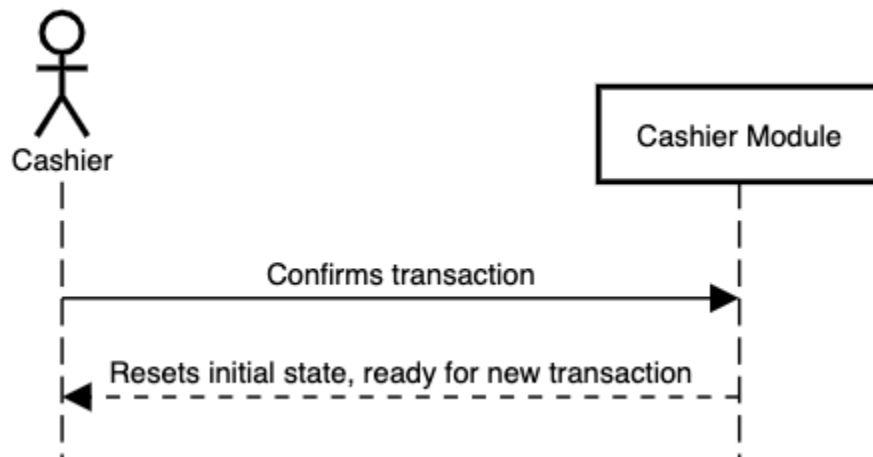
6. Update Loyalty Points



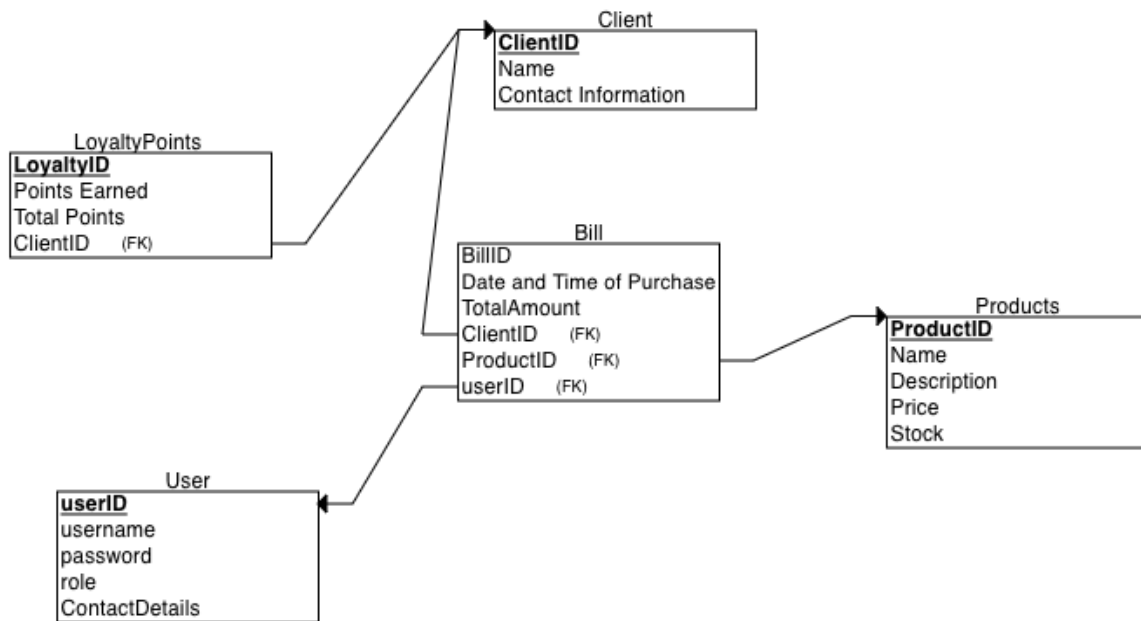
7. Generate and print receipt



8. End transaction



c. Database design:



- Relationships between tables:

1. Clients to Bills: One to Many

Each client can have multiple bills, but one bill is linked to one client.

2. Bills to Products: Many to Many
Each bill has many products and each product can be in many different bills.
3. Users to Bills: One to Many
Each user can create multiple bills, but each bill is connected to only one user.
4. Client to Loyalty Points: One to One
Each client has their own loyalty points.

ii.iii Modeling

Use Case Diagram:

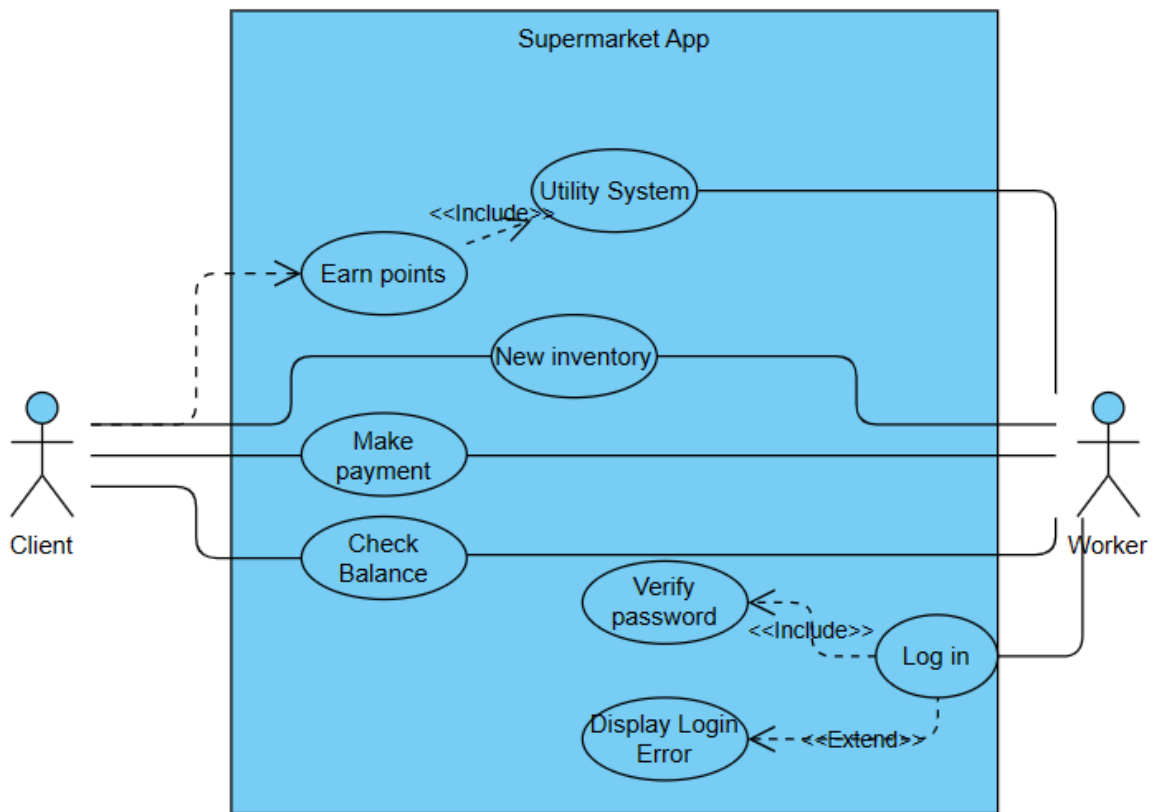
Actors:

- Workers
 1. Cashier
 2. Manager
 3. Administrator
- Client

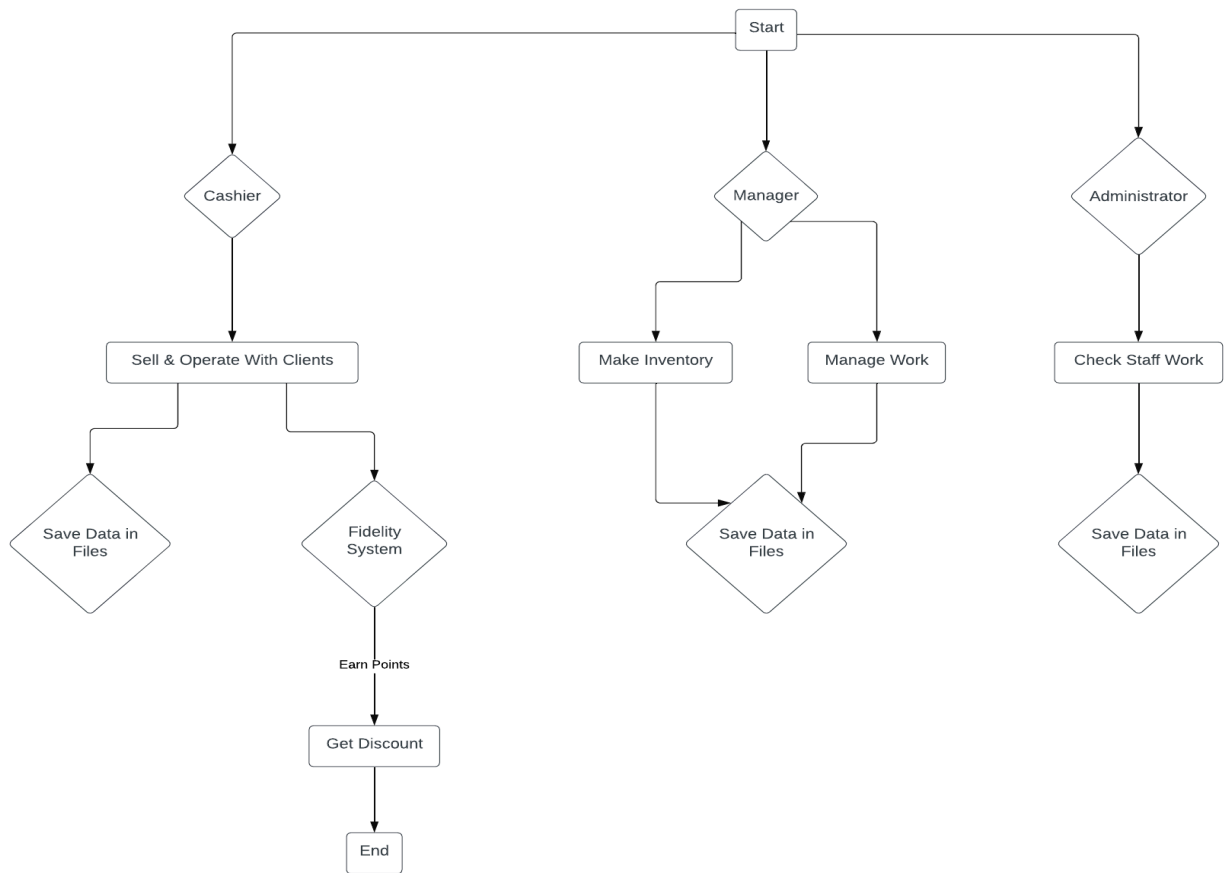
Use Cases:

- Cashier: Sells products, Process payments, Manage client operations
- Manager: Manage inventory, Manage staff work
- Administrator: Monitor staff work, Access system logs

- Use Case Diagram:



- Activity Diagram :



- State Diagram:

States:

1. Login State: Represents the initial state when a user attempts to log in.
2. Authenticated State: Represents the state when a user is successfully authenticated and logged into the system.

