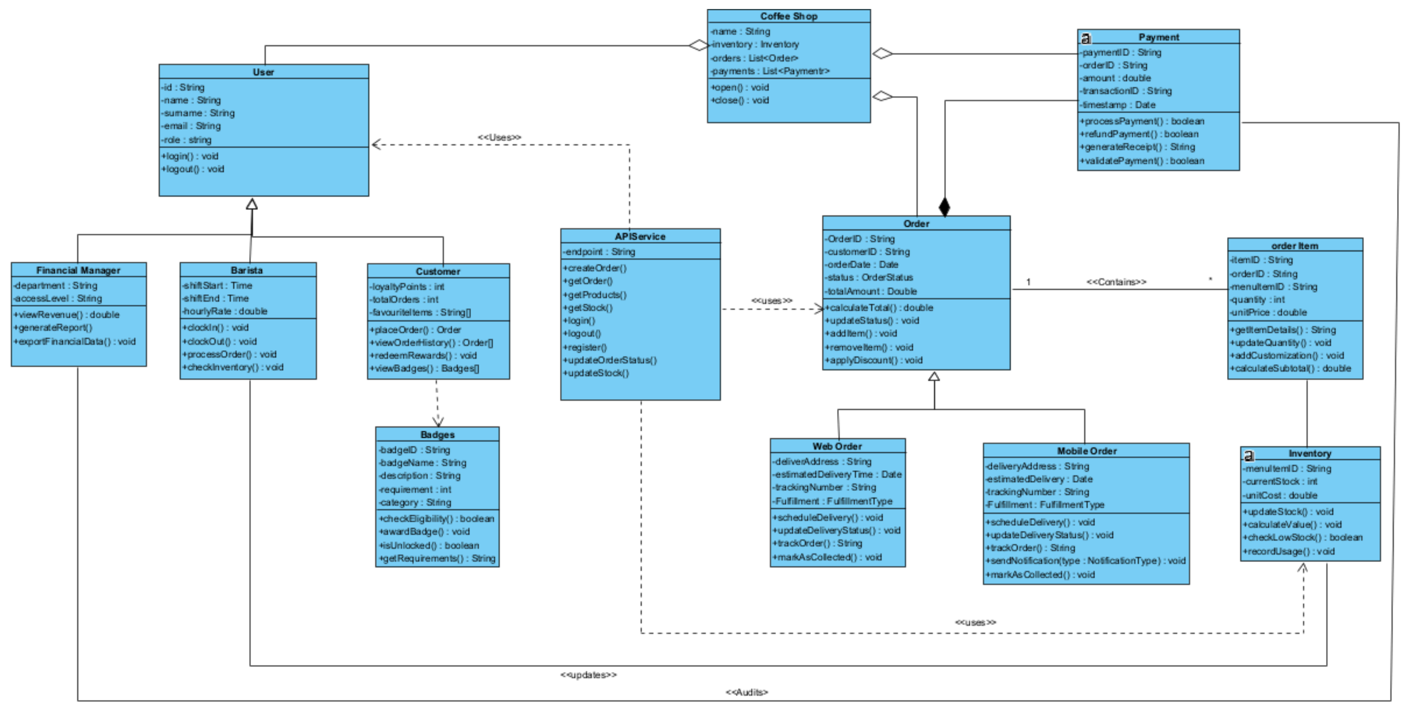
DIE KOFFIEBLIK

Coffee Shop Manager

System Requirement Specification

Small coffee shops constantly struggle with subpar or fragmented tools, even tools that do not really fit their use case. Our project aims to solve that issue by designing an integrated system that contains a point-of-sale (POS), inventory solution, multiple roles (customer, barista, manager, etc.) as well as a mobile application for online ordering.

UML Diagrams



User Characteristics

User Characteristics

- **Customer**
  - Usually a walk-in or online user with basic to moderate tech skills.
  - Uses mobile or web app to place orders, manage account, and track loyalty points.
  - Expects fast, responsive UI and clear feedback (e.g. order updates).
  - High priority on usability and convenience.

- **Barista**
  - Regular staff member operating the POS system.
  - Moderate tech proficiency.
  - Needs real-time visibility of incoming orders and inventory.
  - Interacts with system frequently during a shift.

- **Manager**
  - Oversees inventory, staff schedules, and operations.
  - Uses admin dashboard tools for real-time monitoring and decision-making.
  - Has elevated access to view/edit product data and run reports.
  - May require occasional training but has good system proficiency.

- **Financial Manager**
  - Less frequent system user, mainly involved with analytics and reporting.
  - Requires access to daily/monthly performance metrics.
  - Works with visualized data and CSV exports for financial tracking.
  - High need for data accuracy and availability.

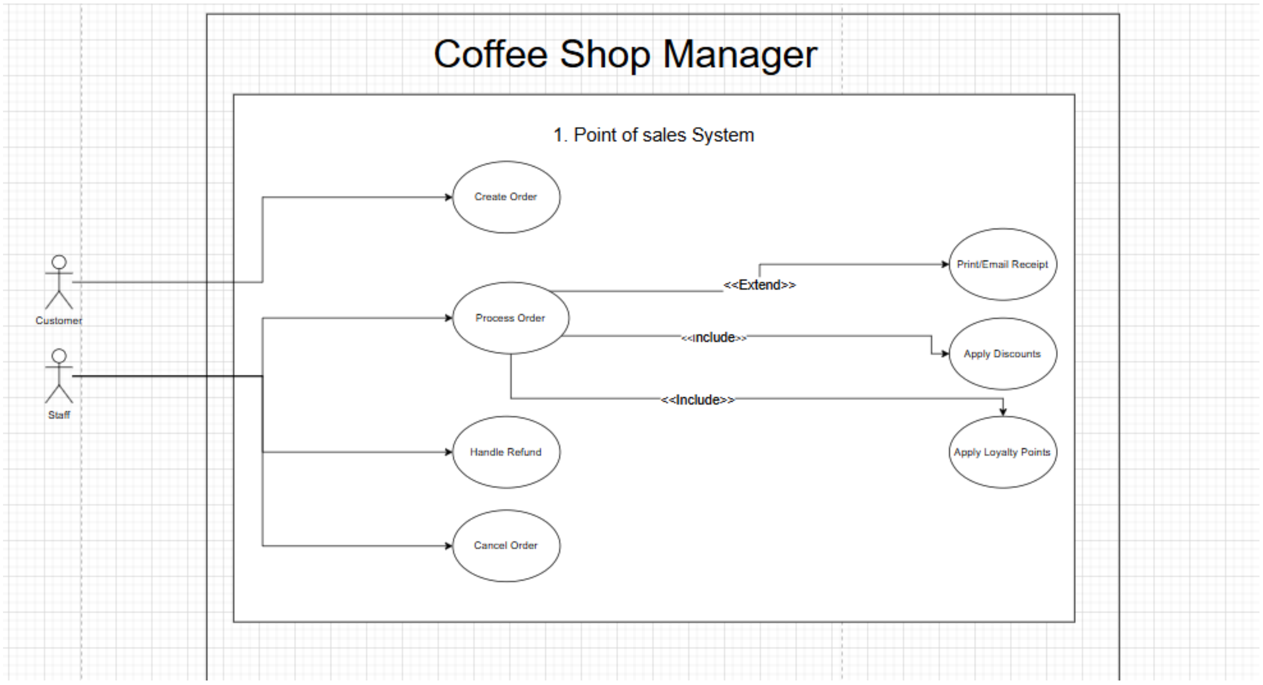
Users' stories

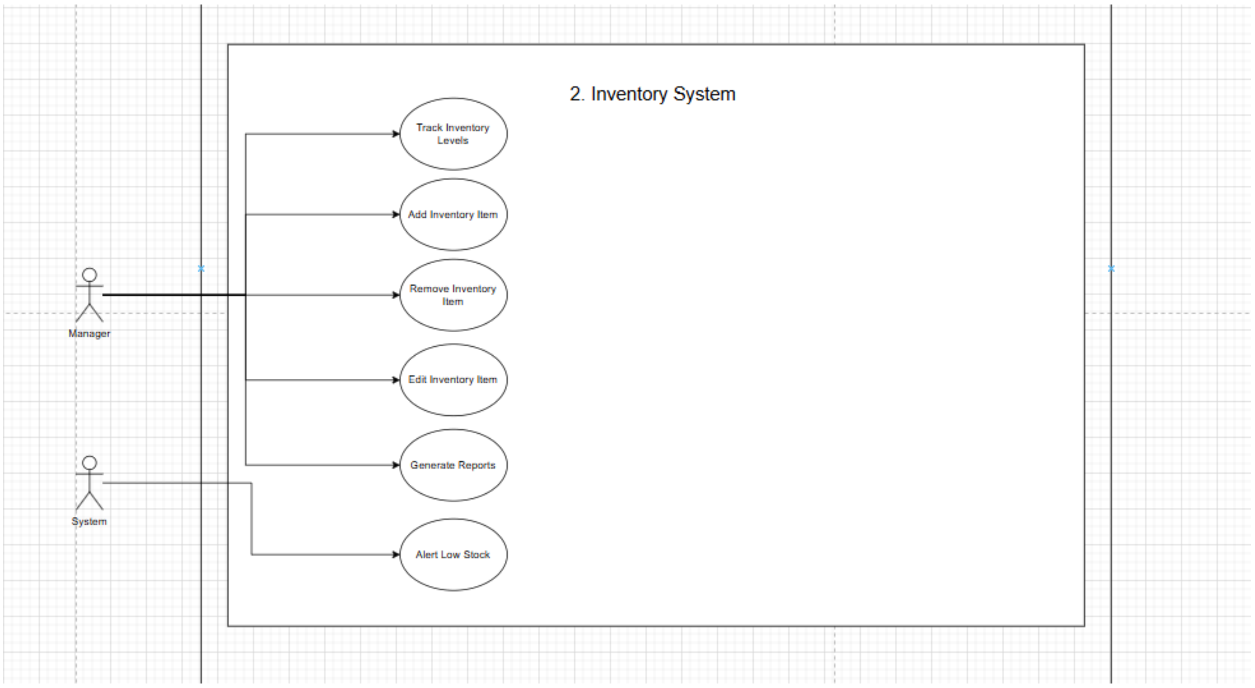
- **Customer: "**As a customer, I want to register and create an account."
- **Customer**: "As a customer, I want to log in to my account so that I can view past orders, loyalty points and place my next order."

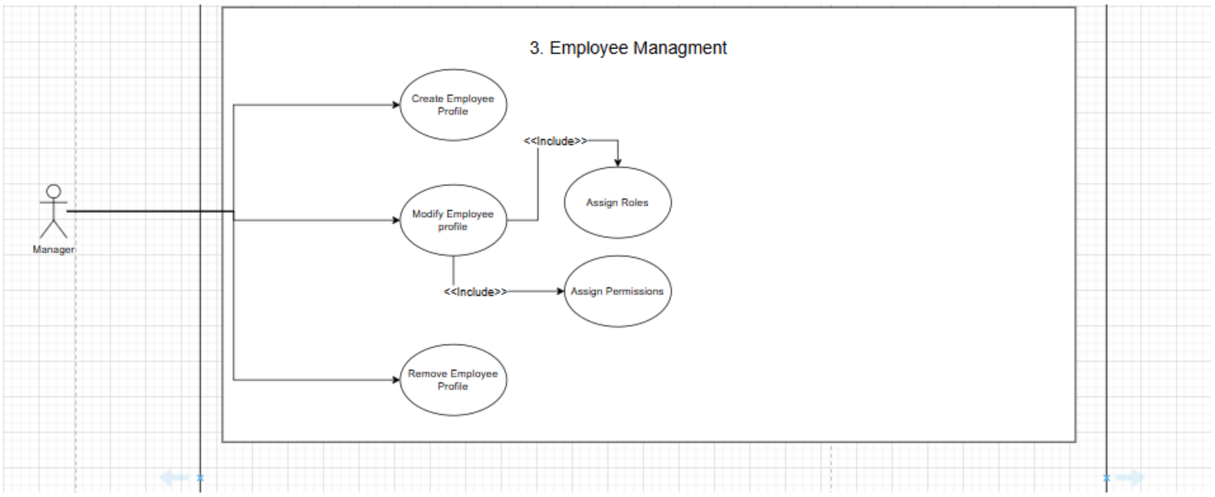
- **Customer: “**As a customer, I want to reset my password to access my account without creating a ticket or contacting support.”
- **Customer**: **“**As a customer, I want to access my account profile to update any relevant information.”
- **Customer:** “As a customer, I want to browse the full menu and descriptions so that I can decide what to order.’
- **Customer**: "As a customer, I want to order coffee via my phone so I can skip the wait in line.”

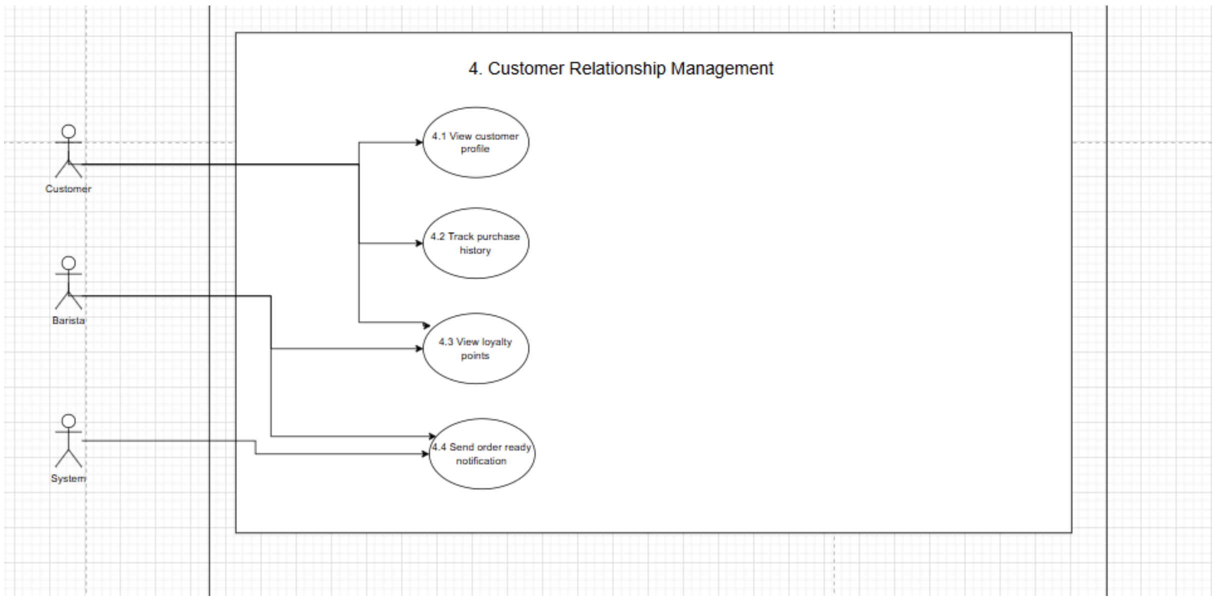
- **Barista**: "As a barista, I want to see incoming orders in a queue so I can prepare them efficiently."
- **Barista**: “ As a barista, I want to update the status of each order to provide real-time updates for customers.”
- **Barista**: “As a barista, I want the system to automatically notify customers once their order is completed and ready for collection. ”
- **Barista:** “ As a barista, I want to see real time stock levels and mark any item ‘out of stock’. ”
- **Barista: “**As a barista, I want to print the receipt for each order after successful payment.**”**
- **Barista: “**As a barista, I want to see customers' loyalty points so I can apply discounts.**”**
- **Manager**: "As a manager, I want to view inventory levels so I can restock in time."
- **Manager:** “As a manager, I want to view sales reports at the end of each day to analyze performance. “
- **Financial Manager**: "As a financial manager, I want to view daily sales reports so I can track revenue."
- **Financial manager:** “As a financial manager, I want to view sales reports to track performance. ”
- **Financial Manager:** “As a financial manager, I want to track expenses to maximise profit.”
- **Financial Manager**: “As a financial manager, I want to track labor costs for each employee.”

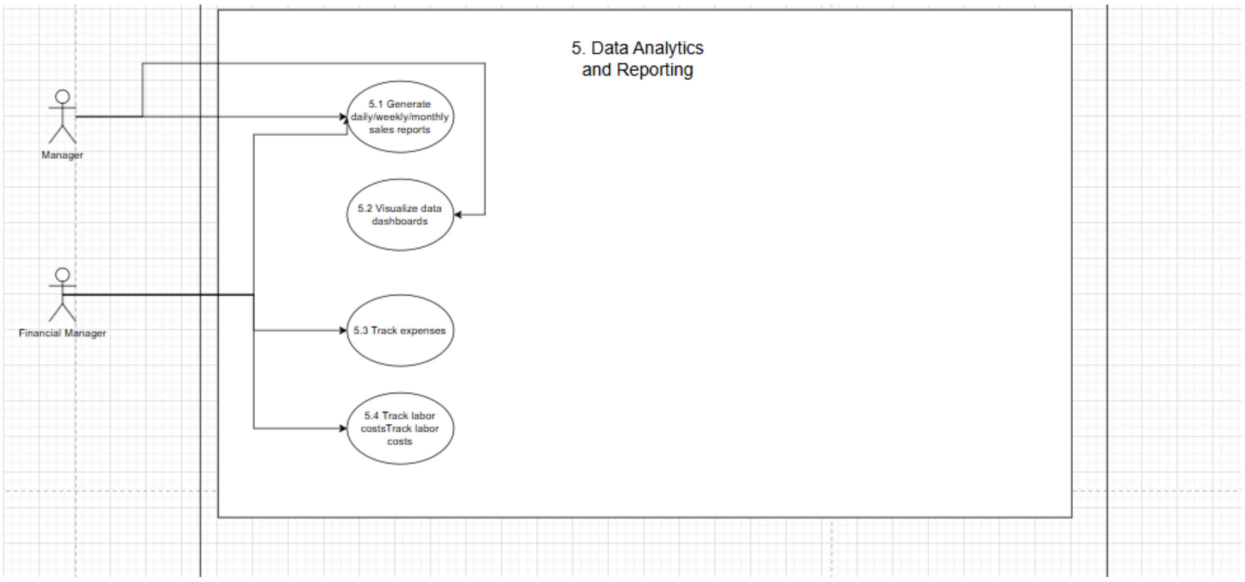
Use Case Diagram

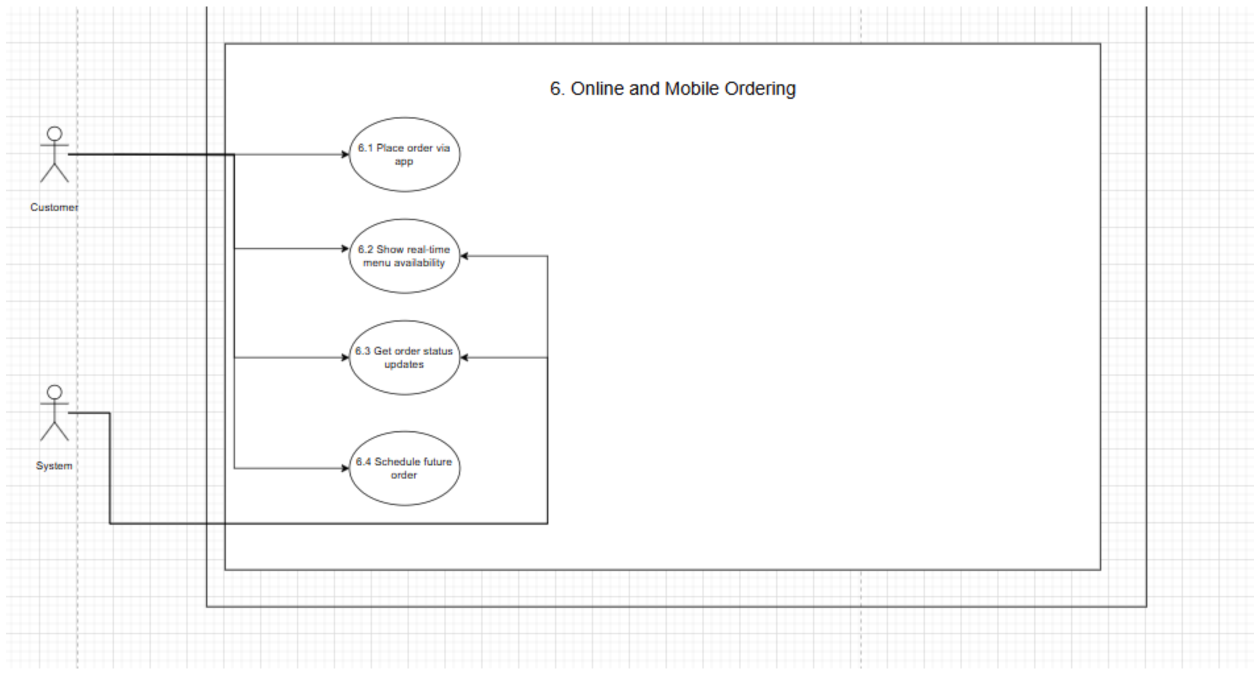





3. Employee Managment
Create Employee Profile
<<Include>>
Assign Roles
Modify Employee profile
Manager
<<Include>>
Assign Permissions
Remove Employee Profile


4. Customer Relationship Management
Customer
4.1 View customer profile
4.2 Track purchase history
Barista
4.3 View loyalty points
System
4.4 Send order ready notification


5. Data Analytics and Reporting
Manager
5.1 Generate daily/weekly/monthly sales reports
5.2 Visualize data dashboards
Financial Manager
5.3 Track expenses
5.4 Track labor costsTrack labor costs


6. Online and Mobile Ordering
Customer
System
6.1 Place order via app
6.2 Show real-time menu availability
6.3 Get order status updates
6.4 Schedule future order

Functional Requirements

1. Point of Sale System (POS)
    1.1. Allow staff to create and process orders.
    1.2. Apply promotional discounts and loyalty points at checkout.
    1.3. Print or email customer receipts.
    1.4. Handle refunds and cancellations.
2. Inventory Management
    2.1. Track inventory levels of ingredients and products in real time.
    2.2. Alert managers when stock drops below a defined threshold.
    2.3. Allow managers to add, edit, or remove inventory items.
    2.4. Generate inventory usage and wastage reports.
    2.5. Support stock intake and update quantities manually or via bulk upload.
3. Employee Management
    3.1. Support creation, modification, and removal of employee profiles.
    3.2. Allow managers to assign roles and permissions to employees.
4. Customer Relationship Management
    4.1. Store customer profiles and preferences.
    4.2. Track and display purchase history.
    4.3. Display customer loyalty points.
    4.4. Notify customers when the order is ready for collection.
5. Data Analytics and Reporting

5.1. Provide daily, monthly, or weekly sales reports.
5.2. Visualise data through dashboards.

6. Online and Mobile Ordering
   6.1. Allow customers to place orders via web or mobile app.
   6.2. Show real-time menu availability based on inventory.
   6.3. Provide real-time order status updates.
   6.4. Allow customers to schedule orders.

Quality Requirements

Security : Use JWT, hashed passwords, and role-based access control.
Maintainability : Modular codebase with clearly documented APIs.
Usability : All customer-facing pages shall follow **responsive design** and be usable on mobile devices.
Portability : The web app shall function correctly on the latest versions of Chrome, Edge, Safari, and Firefox.
Testability : All critical modules will have 90% or higher unit test coverage.
Logging : All user logins, failed login attempts, and order placements shall be logged with timestamps and user roles.

Architectural Requirements

Layered Architecture (3-tier) :

- **Presentation Layer**: Next.js (web), React Native (mobile)
- **Business Logic Layer**: Node.js / Express APIs
- **Data Layer**: PostgreSQL (accessed via Drizzle ORM)

Design Patterns

- Singleton : Database connection
- Factory : User/Order object creation
- Observer : Notification systems

- Strategy : Different discount/reward strategies based on customer types

Constraints

- Must use **PostgreSQL** for persistence
- Must use **React Native** for mobile app (not Flutter, etc.)
- System must run on **Node.js** and be deployable via **Docker**
- Only open-source libraries may be used
- Mobile and web clients must share the same backend API

Technology Requirements

Component	Technology
Web Frontend	Next.js (React, TypeScript)
Mobile App	React Native
Backend API	Node.js + Express
Database	PostgreSQL + Drizzle ORM
Authentication	JWT + Role-based access control
Styling	Tailwind CSS
DevOps / CI/CD	GitHub Actions, Docker
Testing	Jest, React Testing Library