TALLINNA TEHNIKAÜLIKOOL

School of Information Technologies

Lasse Thor Lepik 223042IADB

TeaTime Inventory Management and Selling Application

Web applications in C# project

Supervisor: Andres Käver

Tallinn 2024

Author’s declaration of originality

I hereby certify that I am the sole author of this report. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Lasse Thor Lepik

21.02.2024

Table of Contents

Abstract

TeaTime is a comprehensive inventory management and sales application and it is aimed towards businesses operating within the tea industry. The app aims to streamline the process of managing tea products, from inventory control to point-of-sale transactions, thus enhancing operational efficiency and customer satisfaction. The motivation behind the project is to provide tea retailers and wholesalers with a user-friendly platform that simplifies the complexities of inventory management and sales operations. By leveraging TeaTime, businesses can focus more on product quality and customer service.

1. Analysis

The tea industry, with its vast array of products ranging from loose leaf teas to pre-packaged blends, presents unique challenges in inventory management and sales operations. The need for a specialized application arises from the diverse nature of tea products, which require detailed tracking of varieties, blends, expiration dates, and storage conditions. Additionally, the global tea market's growth demands scalable solutions that can adapt to varying business sizes and types. TeaTime addresses these challenges by offering a tailored solution that incorporates the nuances of tea product management, providing real-time inventory tracking, sales analytics, and customer management features. The application's design focuses on user experience, ensuring that businesses can efficiently manage their operations with minimal training and technical expertise. For ease of use and portability the solution should mainly target mobile devices, but also be useable on desktops. At first the project should target businesses and allow them to manage their inventory. Later it should be developed into a webapp, that is also friendly to customers: allowing them to form orders and pay online without seeing the business-focused management interfaces.

Expected Project Size

The TeaTime application will be developed with a minimum of six entities, excluding many-to-many relationship tables, logging, translations, and identity tables such as User or Role. Project should be able to run basic operations like reading and modifying database tables, perform and output specific calculations and provide a good user experience in a webapp. If project requirements are successfully finished in time, it may be useful to further integrate the application with already existing services and messaging bots.

2. Functionalities

Inventory Management

- **Admin Role:** Admin users can add new products to the inventory, including name, price, and tastes with their respective prices. Admins can also modify existing product details.

Financial Reporting

- **Bookkeeper Role:** In addition to worker capabilities, bookkeepers have access to financial data, including total sales, discounts applied, and net earnings.

Sales Processing

- **Worker Role:** Workers can select inventory items to process sales, including choosing the tea, its taste, and the selling price. They can also apply discounts to the sale price.

User Permissions

- **Root:** Able to perform all actions, including adding and removing admins.
- **Admin:** Can add new products, modify inventory, and access all reporting and sales processing features.
- **Bookkeeper:** Can view financial data, process sales, apply discounts, and access sales summaries.
- **Worker:** Can process sales and view real-time inventory updates.

Real-time Updates

System will display current inventory quantities, allowing workers to see real-time stock levels.

Tracking Profit Generated by Individual Sellers

The solution will keep track of total profit a user has generated during their lifetime. This way businesses can adjust their sales practices or offer extra rewards to sellers for reaching individual profit targets.

Daily Database Backups

To ensure data integrity and facilitate disaster recovery, the system must automatically create daily backups of the PostgreSQL database. These backups should be stored in the "/backups" folder on the server hosting the database and can be downloaded to external devices. Implementing an automated backup procedure will safeguard against data loss due to hardware failure, data corruption, or other unforeseen incidents.

End-of-Day Summary

The service will provide a summary of daily activities, including:

- Total sales and specific teas sold.
- Discounts applied, with explanations for each.
- Total amount of discounts for the day.
- Total money earned by the end of the day.

3. Development Approach

The TeaTime application will be developed using the MVC (Model-View-Controller) architecture, providing a clear separation of concerns, which facilitates maintenance and scalability. The domain models will be created in code, ensuring that the business logic is accurately represented and easily manageable. CRUD (Create, Read, Update, Delete) controllers will be generated for all entities excluding identity-related tables. These controllers will serve as the backbone of the application's operations, enabling the management of products, categories, suppliers, customers, orders, and order items through a web interface.

Technology Stack

Backend: ASP.NET Core with Entity Framework REST backend (Azure/Aws)

Frontend: Unconfirmed

Database: PostgreSQL

Project may or may not use Docker depending on development progress.

4. Entities

Users

- **Purpose**: Stores user information.
- **Fields**:
    - **id**: Primary key, unique identifier for each user.
    - **userid**: Unique big integer ID from a messaging app.
    - **role**: User's role within the application.
    - **sales**: Number of sales performed by the user.
    - **profit**: Total lifetime profit of the user.

Product

- **Purpose**: Catalogs products.
- **Fields**:
    - **id**: Primary key, unique identifier for each product.
    - **name**: Product name.
    - **description**: Description of the product.

Flavor

- **Purpose**: Tracks different taste variants of products.
- **Fields**:
    - **id**: Primary key, unique identifier for each product variant.
    - **name**: Name of the taste.
    - **productId**: Foreign key linking to the **Product** table.
    - **quantity**: Available quantity of this flavor.
    - **price**: Base price of the flavor.

Order

- **Purpose**: Tracks orders.
- **Fields**:
  - **id**: Primary key, unique identifier for each order.
  - **userId**: Foreign key linking to the **Users** table, identifying the seller.
  - **sellingPrice**: Total price at which the order is sold.
  - **discountApplied**: Discount applied to the order.
  - **saleDate**: Timestamp of the sale.
  - **status**: Status of the order (e.g., completed, pending).

Order_Item

- **Purpose**: Details items within an order.
- **Fields**:
  - **id**: Primary key, unique identifier for each order item.
  - **tasteId**: Foreign key linking to the **Flavor** table, identifying the taste.
  - **quantity**: Number of items.
  - **sellingPrice**: Selling price per item.
  - **discountApplied**: Discount applied per item.
  - **saleId**: Associates the item with a sale.
  - **order**: Foreign key linking to the **Order** table.

Financial_Report

- **Purpose**: Records financial summaries.
- **Fields**:
  - **id**: Primary key, unique identifier for each report.
  - **reportDate**: Timestamp of the report.
  - **totalSales**: Total sales value.
  - **totalDiscounts**: Total discounts given.
  - **netEarnings**: Net earnings after discounts.

Promotion

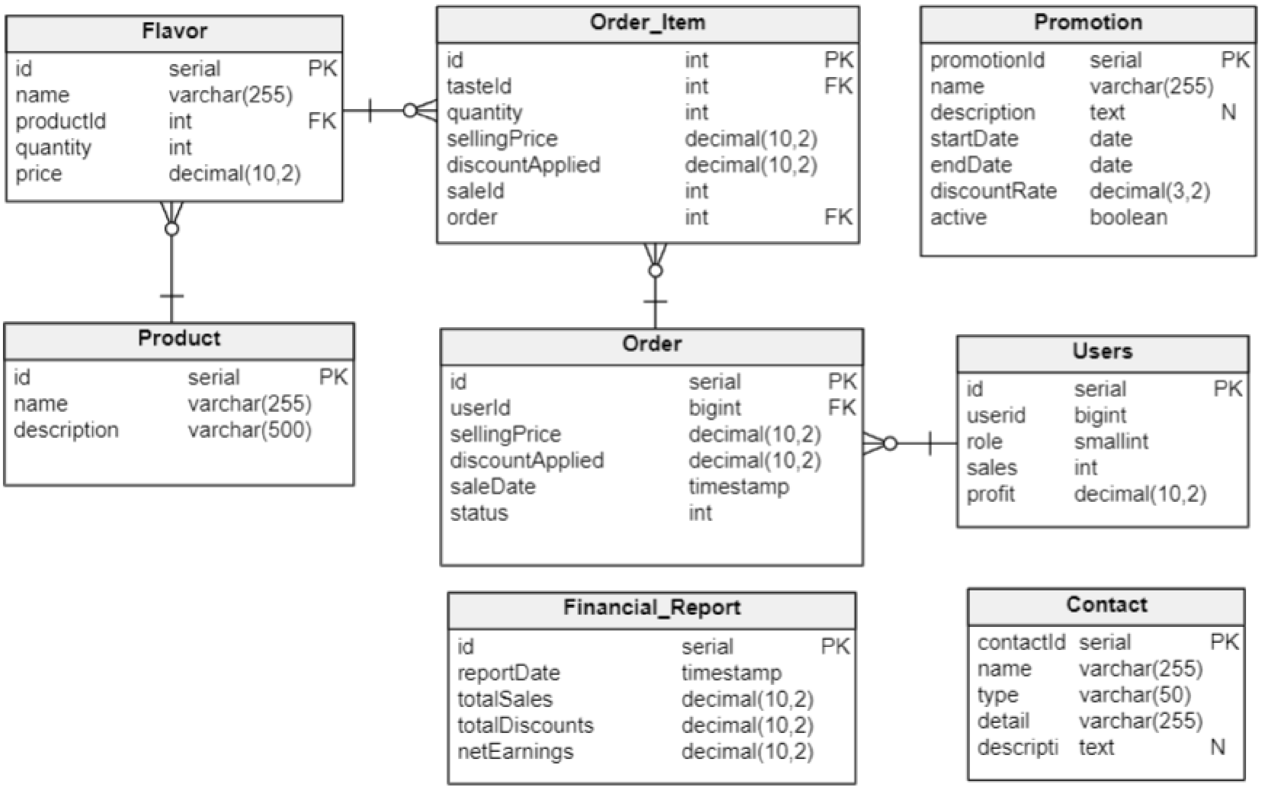
- **Purpose**: Manages promotional campaigns.
- **Fields**:
    - **promotionId**: Primary key, unique identifier for each promotion.
    - **name**: Promotion name.
    - **description**: Detailed description of the promotion.
    - **startDate**: Start date of the promotion.
    - **endDate**: End date of the promotion.
    - **discountRate**: Discount rate offered.
    - **active**: Status indicating whether the promotion is active.

Contact

- **Purpose**: Stores business contact information.
- **Fields**:
    - **contactId**: Primary key, unique identifier for each contact.
    - **name**: Contact's name.
    - **type**: Type of contact (e.g., 'Email', 'Phone').
    - **detail**: Contact details corresponding to the type.
    - **description**: Optional description of the contact.

5. ERD Schema

The full ERD schema is created using Vertabelo. This schema will detail all entities, their attributes, and relationships, providing a comprehensive overview of the database structure.



Flavor		
id	serial	PK
name	varchar(255)	
productId	int	FK
quantity	int	
price	decimal(10,2)	

Order_Item		
id	int	PK
tasteId	int	FK
quantity	int	
sellingPrice	decimal(10,2)	
discountApplied	decimal(10,2)	
saleId	int	
order	int	FK

Promotion		
promotionId	serial	PK
name	varchar(255)	
description	text	N
startDate	date	
endDate	date	
discountRate	decimal(3,2)	
active	boolean	

Product		
id	serial	PK
name	varchar(255)	
description	varchar(500)	

Order		
id	serial	PK
userId	bigint	FK
sellingPrice	decimal(10,2)	
discountApplied	decimal(10,2)	
saleDate	timestamp	
status	int	

Users		
id	serial	PK
userid	bigint	
role	smallint	
sales	int	
profit	decimal(10,2)	

Financial_Report		
id	serial	PK
reportDate	timestamp	
totalSales	decimal(10,2)	
totalDiscounts	decimal(10,2)	
netEarnings	decimal(10,2)	

Contact		
contactId	serial	PK
name	varchar(255)	
type	varchar(50)	
detail	varchar(255)	
descripti	text	N

6. SQL

```
CREATE TABLE Contact (
    contactId serial  NOT NULL,
    name varchar(255)  NOT NULL,
    type varchar(50)  NOT NULL,
    detail varchar(255)  NOT NULL,
    description text  NULL,
    CONSTRAINT Contacts_pk PRIMARY KEY (contactId)
);

CREATE TABLE Financial_Report (
    id serial  NOT NULL,
    reportDate timestamp  NOT NULL,
    totalSales decimal(10,2)  NOT NULL,
    totalDiscounts decimal(10,2)  NOT NULL,
    netEarnings decimal(10,2)  NOT NULL,
    CONSTRAINT Financial_Report_pk PRIMARY KEY (id)
);

CREATE TABLE Flavor (
    id serial  NOT NULL,
    name varchar(255)  NOT NULL,
    productId int  NOT NULL,
    quantity int  NOT NULL,
    price decimal(10,2)  NOT NULL,
    CONSTRAINT Taste_pk PRIMARY KEY (id)
);
```

```
CREATE TABLE "Order" (
    id serial  NOT NULL,
    userId bigint  NOT NULL,
    sellingPrice decimal(10,2)  NOT NULL,
    discountApplied decimal(10,2)  NOT NULL,
    saleDate timestamp  NOT NULL,
    status int  NOT NULL,
    CONSTRAINT Sale_pk PRIMARY KEY (id)
);

CREATE TABLE Order_Item (
    id int  NOT NULL,
    tasteId int  NOT NULL,
    quantity int  NOT NULL,
    sellingPrice decimal(10,2)  NOT NULL,
    discountApplied decimal(10,2)  NOT NULL,
    saleId int  NOT NULL,
    "order" int  NOT NULL,
    CONSTRAINT Order_Item_pk PRIMARY KEY (id)
);

CREATE TABLE Product (
    id serial  NOT NULL,
    name varchar(255)  NOT NULL,
    description varchar(500)  NOT NULL,
    CONSTRAINT Product_pk PRIMARY KEY (id)
);
```

```
CREATE TABLE Promotion (
    promotionId serial  NOT NULL,
    name varchar(255)  NOT NULL,
    description text  NULL,
    startDate date  NOT NULL,
    endDate date  NOT NULL,
    discountRate decimal(3,2)  NOT NULL,
    active boolean  NOT NULL,
    CONSTRAINT Promotions_pk PRIMARY KEY (promotionId)
);

CREATE TABLE Users (
    id serial  NOT NULL,
    userid bigint  NOT NULL,
    role smallint  NOT NULL,
    sales int  NOT NULL,
    profit decimal(10,2)  NOT NULL,
    CONSTRAINT AK_0 UNIQUE (userid) NOT DEFERRABLE  INITIALLY
IMMEDIATE,
    CONSTRAINT Users_pk PRIMARY KEY (id)
);

ALTER TABLE Order_Item ADD CONSTRAINT item_taste
    FOREIGN KEY (tasteId)
    REFERENCES Flavor (id)
    NOT DEFERRABLE
    INITIALLY IMMEDIATE;
```

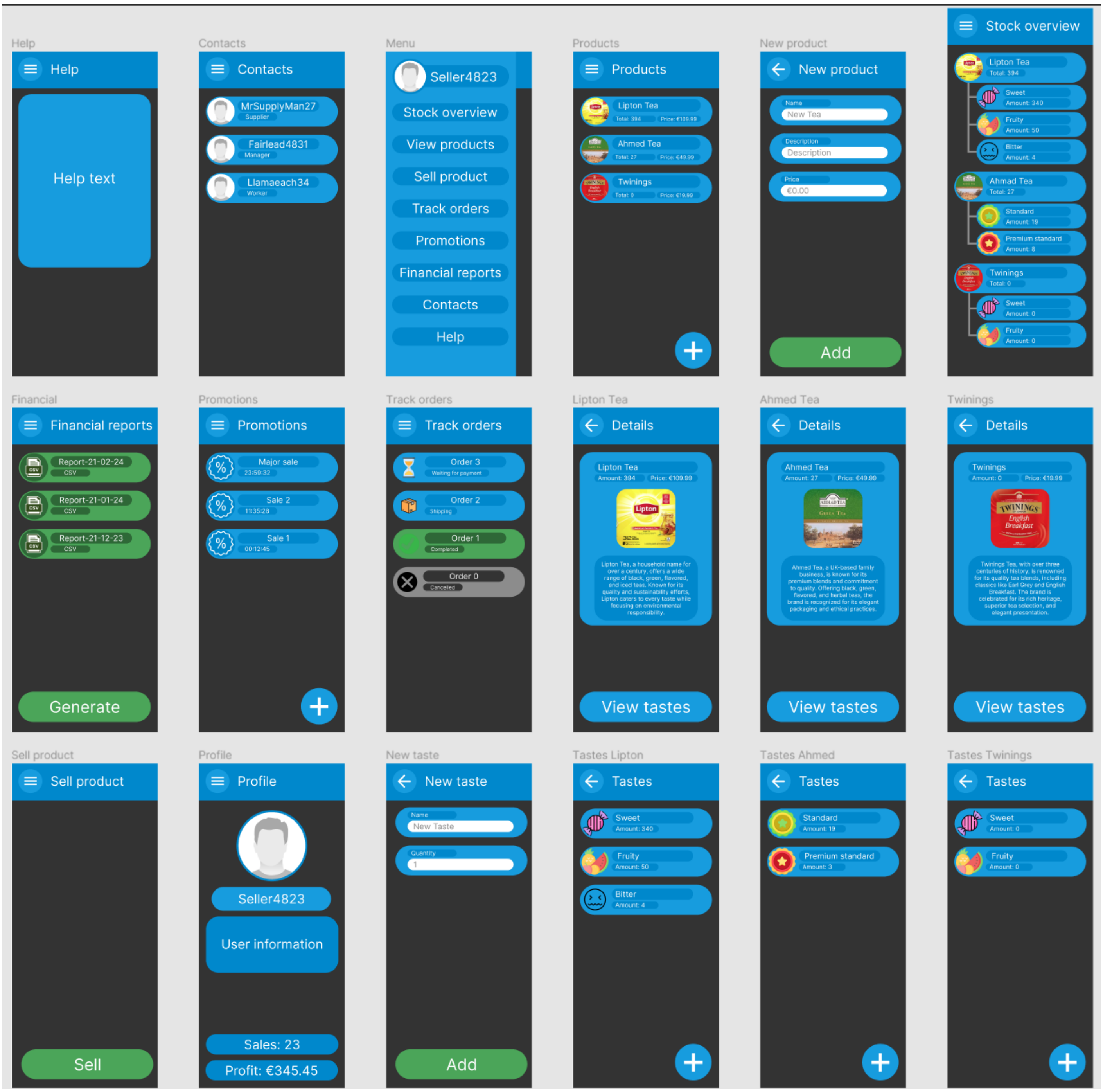
```
ALTER TABLE Order_Item ADD CONSTRAINT order_contains
    FOREIGN KEY ("order")
    REFERENCES "Order" (id)
    NOT DEFERRABLE
    INITIALLY IMMEDIATE;

ALTER TABLE "Order" ADD CONSTRAINT seller_order
    FOREIGN KEY (userId)
    REFERENCES Users (userid)
    NOT DEFERRABLE
    INITIALLY IMMEDIATE;

ALTER TABLE Flavor ADD CONSTRAINT taste_product
    FOREIGN KEY (productId)
    REFERENCES Product (id)
    NOT DEFERRABLE
    INITIALLY IMMEDIATE;
```

7. Interface

Interface of the product may be structured like this:



Lipton Tea, a household name for over a century, offers a wide range of black, green, flavored, and iced teas. Known for its quality and sustainability efforts, Lipton caters to every taste while focusing on environmental responsibility.

Ahmed Tea, a UK-based family business, is known for its premium blends and commitment to quality. Offering black, green, flavored, and herbal teas, the brand is recognized for its elegant packaging and ethical practices.

Twinings Tea, with over three centuries of history, is renowned for its quality tea blends, including classics like Earl Grey and English Breakfast. The brand is celebrated for its rich heritage, superior tea selection, and elegant presentation.

https://www.figma.com/proto/Qfrq7UVn481Mf52gPTQBSq/Untitled-(Copy)?type=design&node-id=10-138&scaling=scale-down&page-id=0%3A1&starting-point-node-id=10%3A138

The initial interface is subject to change and does not reflect the final product.

Conclusions

TeaTime aims to revolutionize the way tea businesses manage their inventory and sales, offering a dedicated solution that caters to the unique needs of the tea industry. By providing a user-friendly platform that simplifies complex operations, TeaTime will empower businesses to focus on what they do best — delivering quality tea products efficiently.