

The hierarchy of the project:

Backend

- structures
 - Transaction C
- Analytics C
- Backend C
- Counter C
- Routes C

Frontend

- Admin
 - AdminMain C
 - MarketingController C
 - SalesController C
 - SoldProduct C
- cashflowFX
 - CashflowMain C
 - CashflowViewController C
 - CashierTransactionViewController C
 - CashierViewController C
 - CustomerViewController C
- interfaces
 - Controller I
 - Window I
- utils
 - WindowLoader C

SharedResources

- exceptions
 - MissingDataException C
 - NoSuchCustomerException C
- productCatalog
 - Product C
 - ProductCatalog C
 - Products C
- structures
 - Customer C
 - CustomerRegister C
- utils
 - HttpController C

Klasser:

backend:

- structures

Transaction: Class that represents a transaction. Contains instance variables that are involved in a transaction, such as customer, timestamp, discount, total and items.

- **Analytics:** For analyzing all sorts of sales, the sales data are in JSON format.
- **Backend:** The main method in this class uses `SpringApplication.run()` method to launch the application with Spring Boot. The `@Bean` retrieves the beans that were created or added by Spring Boot and prints them out.
- **Counter:** Class for counting things by inserting properties. Used by the analytics class.
- **Routes:** Interface for both the admin and cashier parts of the application.

frontend:

- admin

AdminMain: Sets up the first window that pops up when the user chooses to use the admin part of the application. The class contains the main method for running the marketing and sales part.

MarketingController: Controller for the marketing window. This window makes it possible for the user to choose to analyze the products' sales during a specific time. The user will be able to analyze the amount, age group and gender distribution related to the sold products. The user will also be able to analyze bonus customers' purchases.

SalesController: Controller for the sales window. In this window the user will be able to change products' prices and add discounts to them. The discounts can be made for certain customer groups or specific time spans. The discounted products will show up in a table. The user will also be able to analyze the amount of sold items per product and date.

SoldProduct: This class represents a sold product and its variables for storing analytics.

- cashflowFX

CashflowMain: Handles the application launch configuration and launch.

CashflowViewController: Controls the window for choosing the admin or cashier part of the application.

CashierTransactionViewController: This class controls all the buttons and actions that are used during a cashier transaction. This class also initializes the customer view. The products that are added in the cart table are retrieved from the backend. When a sale is finished, the sale data will be sent back to the backend.

CashierViewController: This class controls the window that pops up after the user has clicked the "Cashier" button. By pressing "New Customer Transaction", the user will open the cashier transaction view.

CustomerViewController: Controller for the customer view. This class makes it possible for the customer to see what happens in the shopping cart.

- utils:

WindowLoader: Helper class for opening new windows

sharedResources:

- exceptions:

MissingDataException: Is thrown if the customer ID is missing and a customer object was passed.

NoSuchCustomerException: Is thrown when there is no customer matching the ID.

- productCatalog:

Product: Class that represents a product in the application.

ProductCatalog: Class for creating product catalog from the API.

Products: Visibility layer

- structures:

Customer: Class that represents a customer in the application.

CustomerRegister: Class for creating a customer register from the API.

- utils:

HttpController: Handles Http calls to different backend API's.

Interfaces:

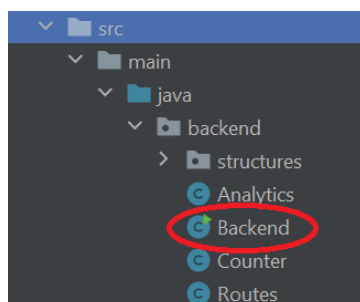
- Controller
- Window

User manual for using the application:

Starting the application

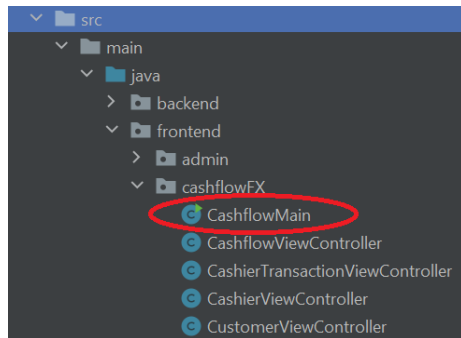
1. Run the backend part of the application.

The main method for the backend can be found here:



2. Run the frontend part of the application.

The main method for the frontend can be found here:



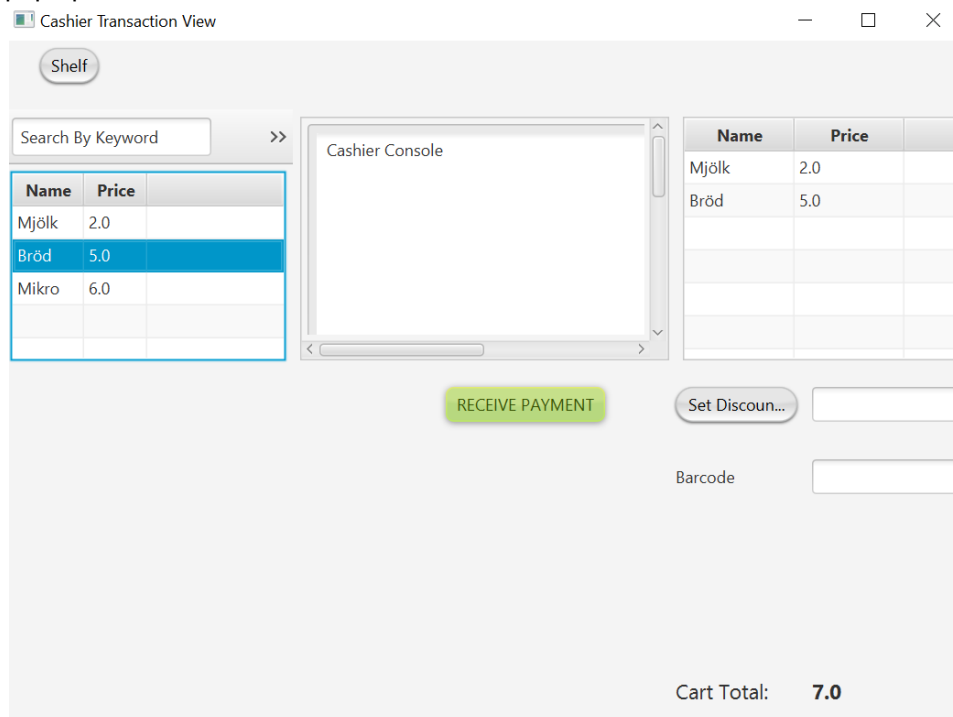
3. A window should pop up with two buttons.

Cashier: Opens up the cashier part of the application

Admin: Opens up the admin part of the application

Cashier

To make a new customer transaction, press the “New Customer Transaction” button. This window will pop up:



Cashier Console: Information about a purchase will be displayed here.

Product table: Table for all products in the catalog. The cashier can search for certain products by entering the product's name, or search for product groups by entering the keyword. Press enter for search.

Cart table: Here you can see all the products that have been added to the cart. To add products to the table, doubleclick a product in the product table.

Set Discount: Markera också To set a new discount to a product, enter the discount percentage in the text field and press “Set Discount”.

Barcode: To add a product to the cart by searching for its barcode, enter the barcode in this text field and press enter.

Cart Total: The total sum of the cart will be shown here.

Shelf: Button to be pressed if a transaction needs to be shelved. While the current transaction is shelved, the cashier can begin a new customer transaction by pressing the “New Customer Transaction” button in the previous window. To unshelve a transaction, press the “Unshelve” button.

Receive Payment: When all products are added to the cart, press this button.

Card Payment: If the customer wants to pay for all or some of the products by card, enter the sum in the textfield and then press “Card Payment” or enter. The card reader will be activated.

Reset Card Reader: Button that can be pressed by the cashier to reset the card reader.

Card Reader Status: Button that can be pressed by the cashier to check the card reader status.


Swipe Result: The cashier needs to press this button to check if a payment has been accepted or not. The result will be shown in the “Cashier Console”.

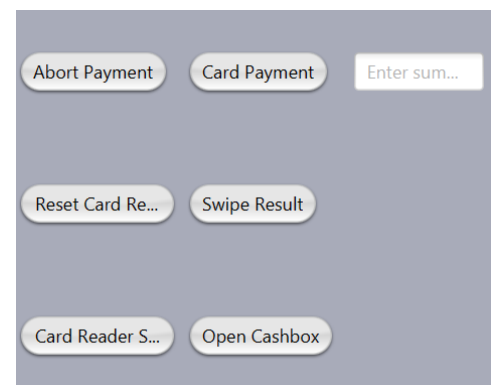
Open Cash box: If the customer wants to pay for all or some of the products with cash, press the “Open Cash box” button, enter the sum in the textfield and then press enter when the customer has paid.

Abort Payment: Button that can be pressed if the cashier wants to abort a payment.

Print Receipt: By pressing this the receipt can be printed and the receipt will be saved as a PNG.

Finish Sale: Button that pops up when all products in the cart have been paid for. By pressing this the customer transaction will close.

Abort Sale: Can be done by pressing  .



Admin

To open the admin part, press the “Admin” button in the “Select User Role” window. This will open up the marketing window:

Popular: Button to be pressed if the user wants to get information about the most and least popular products per time period.

Popular Bonus: Button to be pressed if the user wants to get information about the most and least popular products per time period, buyer age and buyer sex based on bonus customer information.

Date: The user has to choose two dates to be able to analyze purchases at that time period.

Customer: The user can analyze a customer's purchases by entering the customer number in the text field.

By pressing the “Sales” tab at the top of the window, the user will open the sales window.

A 50% discount was added here to the food category. The discount starts on the 26th of October, and ends on the 29th. The discount was added for bonus customers only.

Name	Price	Discount	Bar code	Keyword	Start Date	End Date	Bonus c
Mjöl	9.0	0.5	1234	Mat	26.10.2021	29.10.2021	true
Bröd	8.0	0.5	123	Mat	26.10.2021	29.10.2021	true

The data for products sold on a specific day will be displayed like this. The user only has to choose a specific date and press the button to display this.

Name	Bar code	Amount
Bröd	123	4
Lampa	678	2