

Order Management System

Behtar - Campus Technical Assessment

LMIH Technologies Pvt Ltd

Developer: Sriram Namilakonda

System Manual

This manual provides the detailed information about the order management system

INDEX

Serial Number	Topic	Page Number
1	Introduction to the system	3
2	Technology Stack	4
3	Database overview	5
4	UI overview	7
5	API overview	16
6	Procedure for the system setup	19
7	Flow diagram	20

Introduction to the system

The order management system is a light-weight system for an e-commerce website. The application can be used by both customers and sellers. These two flows are separated from each other in the application.

Customer Functionality

- The customer can view all the items present in the website.
- The customer can view his/her own order history.
- The customer can purchase an item through the online means i.e., using the wallet or net banking or through cash on delivery option. If the customer wallet does not have sufficient money to make a purchase and payment option is chosen to wallet, then the payment gateway automatically gets converted to cash on delivery.
- The customer can choose the seller from list of available based on previous delivery experiences.
- After placing the order, an order number is given to the customer for further reference.

Note: In order to view the products present in the website the customer does not need to login

Seller Functionality

After logging into the system, all the orders by the customers corresponding to that particular seller are displayed and seller can change the status of an order to Dispatched, Delivered or Processing. By default, when the order gets placed by the customer, the status of the order is 'Processing'. The customer must choose a seller while placing the order.

Technology Stack

Database: MySQL

UI development

- ReactJS
- open-source libraries used: react-router-dom, @material-ui/core, @material-ui/lab, @material-ui/icons
- node version>12 npm version>6. (node package manager)
- IDE used for UI development: Visual studio code

API development

- Java
- Framework: Spring
- IDE used for API development: IntelliJ IDEA
- Tool to test the API: Postman
- Spring dependencies: Spring Data JPA, Spring Web, Lombok (for annotation processing) , MySQL web driver
- Unit testing: Mockito framework

Database

The database is created in the local system using MySQL server. The database scripts and the database connection credentials are shared in the database folder attached along with the manual

Database Name: OrderManagementSystem

Tables:

```
mysql> use OrderManagementSystem;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_OrderManagementSystem |
+-----+
| customers
| items
| orders
| sellers
+-----+
4 rows in set (0.01 sec)
```

Metadata for the tables:

1. customers

```
mysql> describe customers;
+-----+-----+-----+-----+-----+-----+
| Field        | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| customer_id  | varchar(255) | NO   | PRI | NULL    |       |
| customer_name | varchar(255) | NO   |     | NULL    |       |
| customer_password | varchar(255) | NO   |     | NULL    |       |
| customer_wallet | float     | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

2. items

```
mysql> describe items;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| item_id | varchar(255) | NO | PRI | NULL | 
| item_name | varchar(255) | NO | | NULL | 
| item_image_url | varchar(1000) | NO | | NULL | 
| item_price | float | NO | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

3. orders

```
mysql> describe orders;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_id | varchar(255) | NO | | NULL | 
| seller_id | varchar(255) | NO | | NULL | 
| order_status | varchar(255) | NO | | NULL | 
| payment_mode | varchar(255) | NO | | NULL | 
| order_value | float | NO | | NULL | 
| item_id | varchar(255) | NO | | NULL | 
| order_id | int | NO | PRI | NULL | 
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

4. sellers

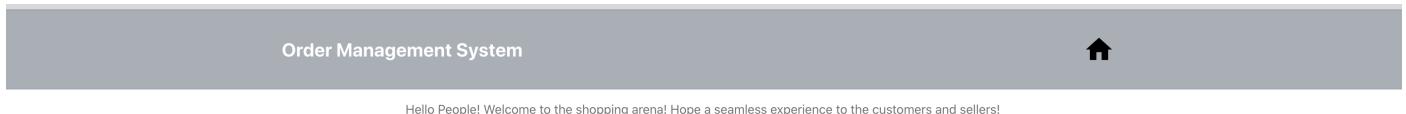
```
mysql> describe sellers;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| seller_id | varchar(255) | NO | PRI | NULL | 
| seller_location | varchar(255) | NO | | NULL | 
| seller_name | varchar(255) | NO | | NULL | 
| seller_password | varchar(255) | NO | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

UI Overview

Port number: 3000

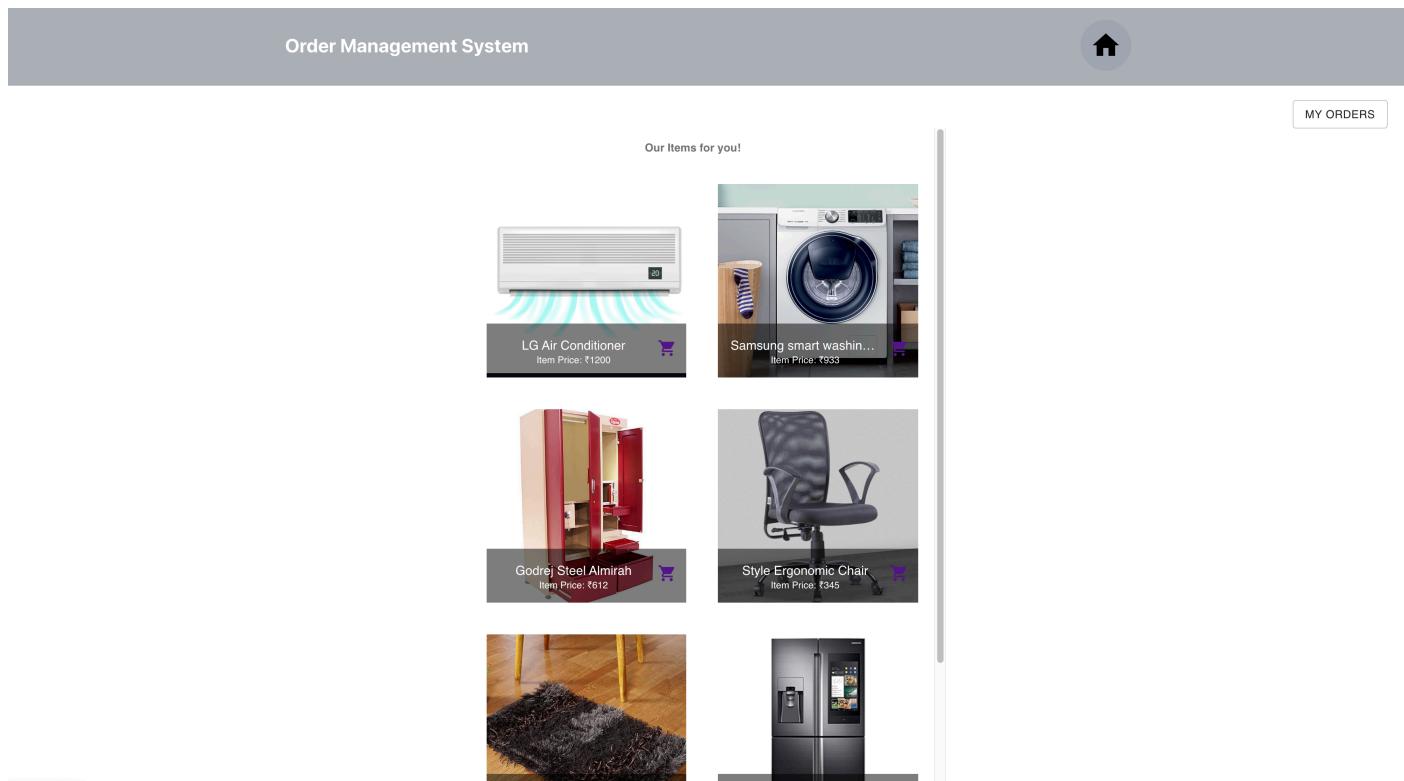
UI endpoint: <http://localhost:3000/>

1. Home Page



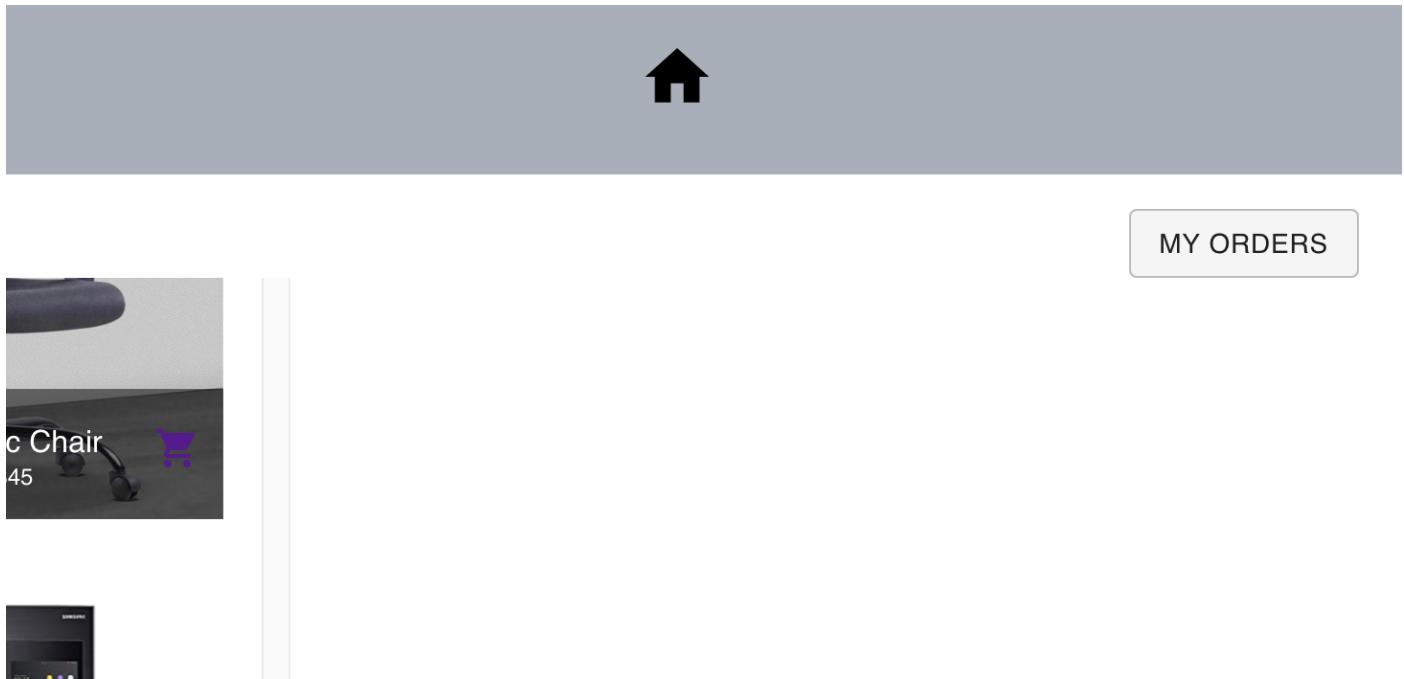
This is the home page of the system where the user can login based on the role (Customer or Seller)

2. Items Display



If the role chosen at home page to be customer, the products present in the website are displayed in this manner

3. Home Icon and My Orders button



The home icon present on the navigation pane redirects us to home page from anywhere in the application and upon clicking “My Orders” button, the system asks the customer to login and displays the customer profile (which also contains the order history of the customers)

4. Customers Profile

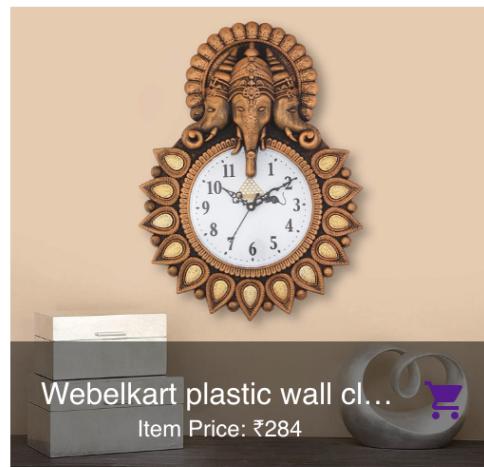
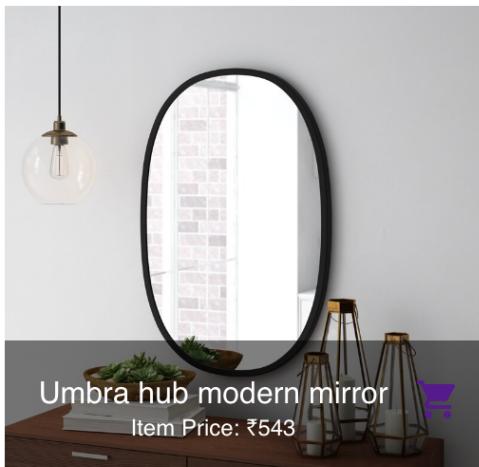
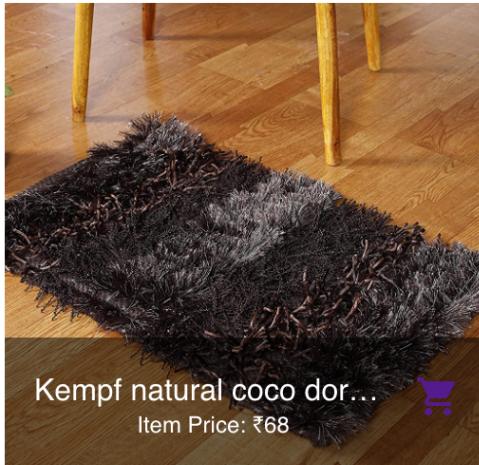
The screenshot shows the 'Customer Profile' section of the Order Management System. At the top, it displays the customer's name, 'Sriram Namilakonda', and their current wallet balance, '₹ 724'. Below this, there is a button labeled 'Add Money to Wallet:' with four options: ₹100, ₹500, ₹1000, and ₹5000. A large 'SHOP NOW!' button is prominently displayed at the bottom.

Order History

Order ID	Customer ID	Order Value	Payment Status	Order Status
1961834	ram	₹543	Paid through Wallet	DELIVERED
4786586	ram	₹933	Paid through Wallet	PROCESSING
8274218	ram	₹1100	Paid through Card	PROCESSING

The customer Profile window contains details such as Customer Name, Money Present in the customer wallet, the order history of the customer and the customer can even add money to the wallet by clicking on any of the displayed buttons that indicates the denominations. The “Shop Now” button redirects the customer to items display window

5. Select an Item to purchase



Clicking the cart icon attached to an item indicates that the customer is ready to purchase the corresponding item. Therefore, upon clicking the icon, the system asks the customer to login and initiate the payment process in order to place the order for the item.

6. Customer login form



Order Management System Home

Customer Login Form

Customer Id:

Password:

SUBMIT

This is the customer login form through which customer specific actions can be performed through the system

7. Customer login form failure



Order Management System Home

Customer Login Form

Customer Id:

raman

Password:

.....

SUBMIT

! Customer Login Failure
The customer with id: raman is invalid!

If the customer enters invalid credentials (username or password), then the alert message is displayed as depicted

8. Customer payment Window

The screenshot shows the 'Customer Payment Window' of the Order Management System. At the top, it displays the customer name 'Sriram Namilakonda' and the current wallet balance '₹ 724'. Below this, there is a section titled 'Add Money to Wallet:' with four buttons for adding ₹100, ₹500, ₹1000, or ₹5000. The item being purchased is 'Theoren fabric sofaset' at a price of ₹ 1100. A dropdown menu for 'Choose a Payment Option' is open, showing 'Pay through Wallet' and 'Net Banking' as choices. A note at the bottom states: 'Note: If the form is submitted without selecting any payment option or if the payment option selected is through wallet and wallet does not have sufficient money for purchase, the payment option will be converted to cash on delivery!'

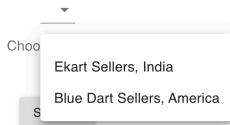
The customer payment window depicts the customer related information as well as it facilitates the addition of amount to the customer wallet if the balance is less than the order value.

9. Choosing a payment option

The screenshot shows the 'Customer Payment Window' interface again. It displays the same customer information and item details as the previous screenshot. The 'Choose a Payment Option' dropdown is open, showing 'Pay through Wallet' and 'Net Banking' as options. A note at the bottom states: 'Note: If the form is submitted without selecting any payment option or if the payment option selected is through wallet and wallet does not have sufficient money for purchase, the payment option will be converted to cash on delivery!'

The customer can choose to pay through wallet or through net banking. If the customer does not select any payment option or if the customer chose to pay through wallet and it turns out that wallet does not have sufficient amount to place the order, then the payment mode will be "Cash on Delivery". The amount will be deducted from customer's wallet for this order if the customer chose to pay through wallet.

10. Seller choice



(i) Note: If the form is submitted without selecting any payment option or if the payment option selected is through wallet and wallet does not have sufficient money for purchase, the payment option will be converted to cash on delivery!

The customer needs to choose a seller while placing the order. The seller chosen will be responsible to deliver the order to the customer

11. Order history in the customer payment window

(i) Note: If the form is submitted without selecting any payment option or if the payment option selected is through wallet and wallet does not have sufficient money for purchase, the payment option will be converted to cash on delivery!

Order History

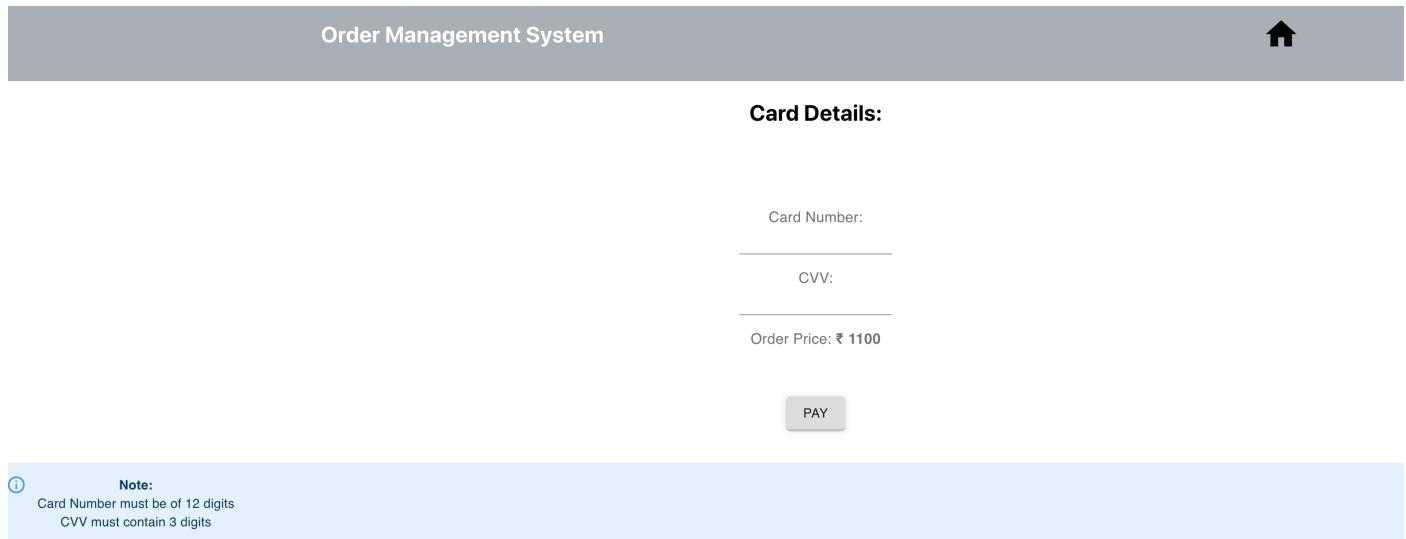
Order ID	Customer ID	Order Value	Payment Status	Order Status
1961834	ram	₹543	Paid through Wallet	DELIVERED
4786586	ram	₹933	Paid through Wallet	PROCESSING

Even in the payment window of an order, the order history of the customer is displayed. The order history contains the details such as Order Id, Customer Id, Order Value, Payment Status and Order Status.

The Payment Status can be: Paid through Wallet, Paid through Card, Cash On Delivery

The Order Status can be: PROCESSING, DELIVERED, DISPATCHED

12. Card Payment



The screenshot shows a card payment form titled "Card Details:" within a "Order Management System". It includes fields for Card Number, CVV, and Order Price (₹ 1100), a "PAY" button, and a note about digit requirements.

Note:
Card Number must be of 12 digits
CVV must contain 3 digits

If the payment mode chosen by the customer is through card, the card details form is displayed upon payment form submission from the customer payment window. If the payment mode chosen by the customer is through wallet or cash on delivery, then system displays order summary directly.

13. Order summary



The screenshot shows an order summary window titled "Order Report" within a "Order Management System". It displays a success message, order details, and a "CONTINUE SHOPPING!" button.

Hi, The Order has been placed successfully!
Please Expect the delivery of the product within 3 days!
Please note that your Order Number is: **8274218**, for further reference
Thanks for shopping with us!!

CONTINUE SHOPPING!

The order summary window acknowledges the customer for placing the order successfully. It also displays system generated Order Id to the customer for further reference. It displays the expected time of arrival to the customer (in number of days). The “Continue Shopping” button redirects the customer to items display window of the application.

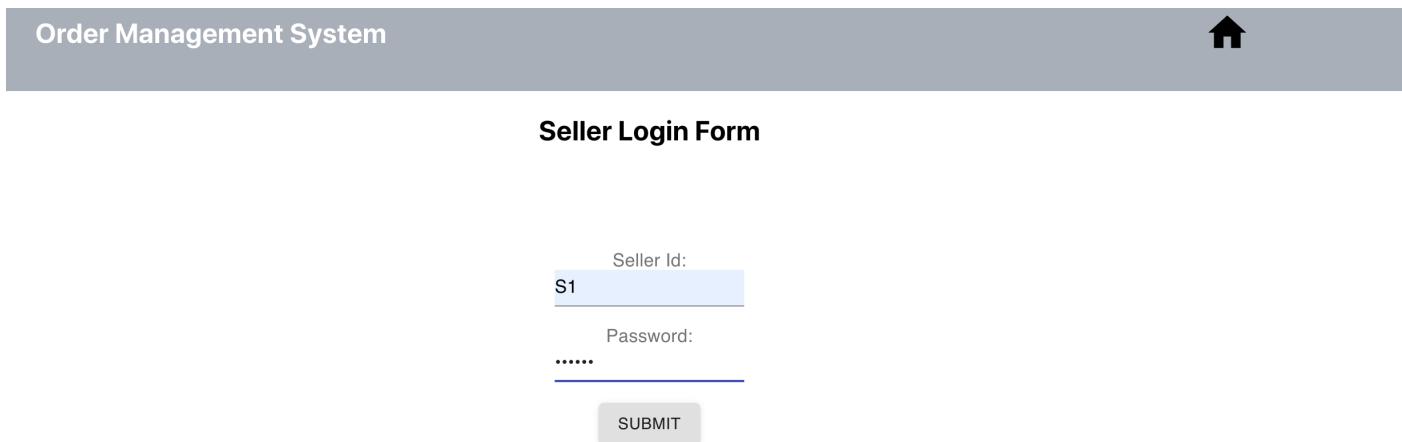
14.Seller login choice

CUSTOMER LOGIN

SELLER LOGIN

The seller can login to the system by clicking the corresponding button in the home page

15.Seller login form



The screenshot shows a "Seller Login Form" interface. At the top, there is a header bar with the text "Order Management System" on the left and a house icon on the right. Below the header, the title "Seller Login Form" is centered. The form itself has two input fields: "Seller Id:" with the value "S1" and "Password:" with the value ".....". A "SUBMIT" button is located at the bottom right of the form area.

The seller login form will be displayed upon the appropriate choice made on the home page. The seller needs to provide the Id and password. If the credentials entered turns out to be invalid, the alter message is displayed alike customer login form.

16.Seller Orders

Orders List of Ekart Sellers, India					
Order ID	Customer ID	Order Value	Payment Status	Order Status	Change Status
1961834	ram	₹543	Paid through Wallet	DELIVERED	PROCESSING DELIVERED DISPATCHED
8274218	ram	₹1100	Paid through Card	PROCESSING	DELIVERED DISPATCHED

The list of orders will be displayed upon seller logging into the system successfully. The seller can change the status of the orders to PROCESSING, DELIVERED, DISPATCHED by clicking on the appropriate buttons attached to the order.

API overview

Rest endpoints and brief description:

Port number:8080

Prefix: <http://localhost:8080/>

1. customer/login/{id}/{password} (GET)

It authenticates the credentials of the customer with given “id” and “password”

2. customer/{id} (GET)

It returns the details of the customer with given “id”

3. deductWallet/{id}/{price} (PUT)

It deducts the amount by “price” from the wallet of the customer with mentioned “id”

4. addWallet/{id}/{price} (PUT)

It adds the amount by “price” to the wallet of the customer with mentioned “id”

5. getItems (GET)

It retrieves the list of all the items present in the website

6. item/{id} (GET)

It retrieves the details of the item with the given “id”

7. placeOrder/{CustomerId}/{ItemId}/{SellerId}/{Payment}/{ItemPrice} (POST)

It places the order for the customer with given “CustomerId” for the item with given “ItemId” and for the seller with given “SellerId” and the payment mode as “payment” and the order value as “ItemPrice”. It indeed returns the system generated number that indicates the corresponding order Id

8. getOrders/{sellerId} (GET)

It retrieves the list of orders for the seller with given “sellerId”

9. updateOrderStatus/{orderId}/{new Status} (PUT)

It updates the order status of the order with given “orderId” to “new Status”

10.getOrders/customer/{customerId} (GET)

It retrieves the list of orders of the customer with given “customerId”

11.seller/login/{id}/{password} (GET)

It authenticates the credentials of the seller with given “id” and “password”

12.seller/getSellers (GET)

It retrieves the list of all the sellers

13.seller/{id} (GET)

It retrieves the details of the seller with given “id”

Code coverage upon unit testing:

1. Code coverage Report

[all classes]			
Overall Coverage Summary			
Package	Class, %	Method, %	Line, %
all classes	100% (13/ 13)	98.5% (66/ 67)	97.6% (121/ 124)
Coverage Breakdown			
Package	Class, %	Method, %	Line, %
com.example.OrderManagementSystem	100% (1/ 1)	50% (1/ 2)	33.3% (1/ 3)
com.example.OrderManagementSystem.Controllers	100% (4/ 4)	100% (17/ 17)	100% (17/ 17)
com.example.OrderManagementSystem.Entities	100% (4/ 4)	100% (27/ 27)	100% (27/ 27)
com.example.OrderManagementSystem.Sevices	100% (4/ 4)	100% (21/ 21)	98.7% (76/ 77)

generated on 2020-10-24 16:06

Class Coverage: 100%

Method Coverage: 98.5%

Line Coverage: 97.6%

2. Unit tests execution snapshot

The screenshot shows the IntelliJ IDEA interface during the execution of unit tests for the `OrderManagementSystem` application. The test runner window displays the following information:

- Time taken:** 471ms
- Tests passed:** 32 of 32 tests - 471ms
- Test Suite:** OrderManagementSystem (com.example) 471ms
- Test Classes:** CustomersControllerTest, OrderManagementSystemApplicationTest, ItemsControllerTest, ItemsServiceTest, SellersServiceTest, OrdersControllerTest, CustomersServiceTest, OrdersControllerTest.
- Logs:** Shows Spring Boot startup logs and test execution details, including DEBUG and INFO levels for various components like `BootstrapUtils`, `SpringBootTestContextBootstrapper`, and `AbstractContextLoader`.

Procedure for system setup

- Install the complete technology stack specified in the local system
- Execute the scripts present in ‘database` folder attached to the manual in MySQL workbench
- Ensure the presence of Java version 8 in the system
- Ensure all the dependencies present in pom.xml file of backend spring boot application have been imported successfully
- Run the spring boot application
- Ensure the installation of node package manager
- Open the front-end application present in folder “routing” in any of the code editor, run the command called “npm install” to install all the required UI related libraries to run the application. Ensure the presence of @material-ui/core, @material-ui/lab, @material-ui/icons and “react-router-dom” libraries
- Start the ReactJs application using “npm start”. A browser window opens in the machine with port number “3000” indicating the instance of application execution.

The “npm run build” command is executed in order to create the optimized build of the application for production. It is present in the build folder of source path in front end application.

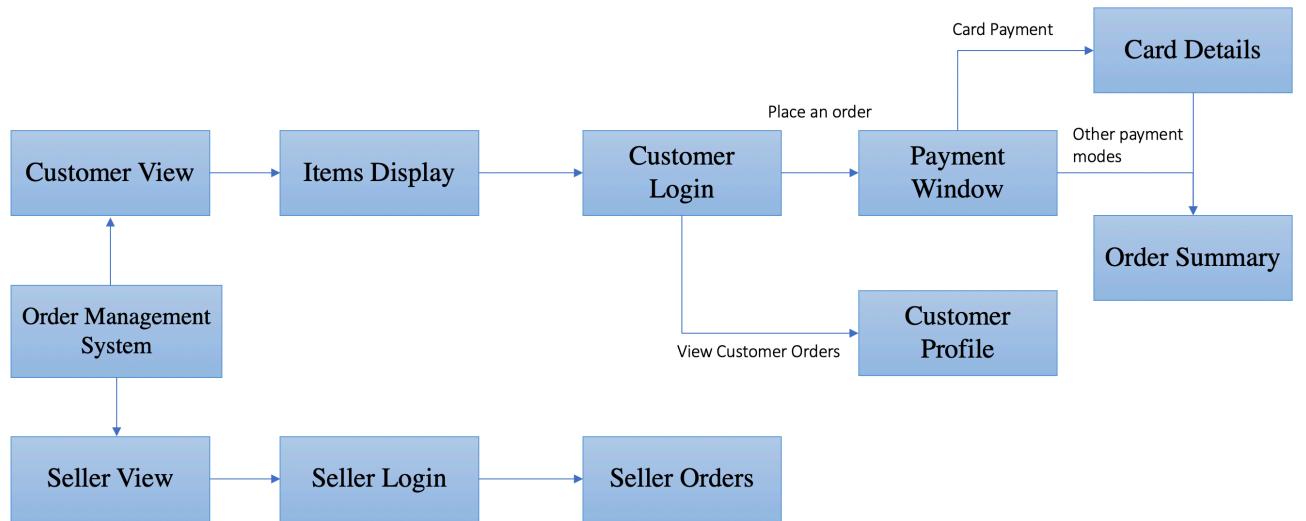
Folders attached

routing: This folder contains the front-end code (ReactJs application)

OrderManagementSystem: This folder contains the back-end code (Springboot application)

Database: This folder contains all the MySQL database scripts to be executed and the text file contains the required credentials to establish database connection

Flow diagram



Order Management System

Sriram Namilakonda

Registration Number: 11707779

University: Lovely Professional University

Email: sriramnamilakonda1@gmail.com

Github Profile: <https://github.com/namilakondaram>

Phone: +91 9849737755

