

CS521- OBJECT ORIENTED ANALYSIS AND DESIGN

TOPIC: FOODSTORE

DATE: 06/30/2016

FIRST NAME	LAST NAME
GAUTHAM	KANNAN
GOPIKAA	MANIKANDAN
NIKHIL	BIRUR
PAVITHRA	KUTTIYANDI CHANDRAKASU
SHIVANI	SHINDE

APPENDIX

APPENDIX: A

A.1 PHASE I

A.1.1 OVERVIEW.....	Page 3
A.1.2 FEATURE LIST.....	Page 4
A.1.3 REQUIREMENTS	Page 5
A.1.4 DOMAIN DICTIONARY	Page 7
A.1.5 ACTOR DICTIONARY.....	Page 11
A.1.6 USE CASES	Page 13
A.1.7 USE CASE DIAGRAM.....	Page 14

APPENDIX: B

B.1 PHASE II

B.1.1 ANALYSIS MODEL	Page 15
B.1.2 DESIGN MODEL	
B.1.2.1 CLASS DIAGRAM.....	Page 20
B.1.2.2 OBJECT DATA MODEL.....	Page 21
B.1.3 SEQUENCE DIAGRAM.....	Page 22
B.1.4 DESIGN PATTERNS	
B.1.4.1 COMMAND PATTERN.....	Page 25
B.1.4.2 MODEL-VIEW-ARCHITECTURE PATTERN.....	Page 25
B.1.4.3 STATE PATTERN.....	Page 26

APPENDIX: C

C.1 PHASE III

C.1.1 IMPLEMENTATION.....	Page 26
C.1.2 SCREENSHOTS	Page 27

APPENDIX: A

A.1 PHASE I

A.1.1 OVERVIEW

Growing interest among the consumers to point and click on their manner through any aspects of everyday existence has fueled the internet economic system to increase offerings and sell merchandise online, even in areas that were the sole domain of conventional organizations, which includes grocery stores and pharmacies. Organizations that sell customer products online are called "e-tailors" and when it comes to the retailers who sell groceries online, they are called "e-grocers".

A number of reasons such as better charges, large choice, comfort, and time saving have encouraged the consumers to buy products online, including groceries. One of the main aims of e-grocers is to become a single-source solution for the busy consumer today. Besides selling products, e-grocers are also responsible for constant innovation. For Example, Netgrocer website has a unique feature which adds products to the customer's cart, based on the recipe uploaded by the consumer. Customization of weekly buying lists and lists of often ordered merchandise in addition to personalized buying aisles are extra advantages of purchasing on line versus in conventional shops. Storing weekly orders on the internet site makes repeat orders easy.

In this project, we have created an application similar to Netgrocer which enables users to order groceries or other food items online, make their payments online and in case they face any issue, they can report the issue online. This project is developed in Java Swings and MySQL. This project report includes design and implementation of our application.

A.1.2 FEATURE LIST

- User Account Registration
- Login for any user / employee
- Display Products on the webpage
- Product Specification (Detailed description about the product)
- Product Search (Enabling user to search the require item)
- Checkout (Once products are selected i.e. added to the cart)
- Provide Support (Any assistance required i.e. report issues)
- Product Management
- Verify Credit card before the user account is created.
- Authenticate transaction, once the user proceeds with the order.
- Manage databases (all the user data/transaction data/product data is maintained)
- Pick up goods once they are ready for delivery.
- Deliver those goods
- Pick up returned goods from the customers (incase the customer's requests for it)
- After pickup, deliver the returned goods to the dealers.
- Handle reported issues.

A.1.3 REQUIREMENTS

- **For a new user:** First Name, Last Name, Contact Details, Email, Credit Card details
- **For Login (by Customer/ Support team/ Dealer/ Database Admin/ Delivery Admin/ Technician):** Username, Password, Email address (Incase, the user forgets his/her username or password)
- **The customer must be able to:**
 - ☐ View displayed list of products.
 - ☐ Search of specific products
 - Category based
 - Brand based
 - Pricewise
 - Date based.
 - ☐ View the cart.
 - Add/Delete/Update items to cart
 - ☐ View recommended products.
 - ☐ Checkout and make payment (Choose shipping address and verify payment details)
 - ☐ Manage Account and Get help (Customer support)
- **The Support team must:**
 - ☐ Answer to customer issue in the form of chat or email.
 - ☐ Create tickets and escalate to respective Admin based on the issue, if necessary.

○ **The Dealer must:**

- ☐ Add new product
- ☐ Replenish the quantity of existing products
- ☐ Remove the old stock.

○ **The Bank must:**

- ☐ Verify the Credit card of the new customer as soon as he registers
- ☐ Add a new card
- ☐ Verify the card of an existing customer when he edits the card details
- ☐ Authenticate the payment when a customer checks out and places the final order.

○ **The Database Admin must:**

- ☐ Make any required changes to the database
- ☐ Manage the database.

○ **The Delivery Admin must:**

- ☐ Pick up goods from the dealer when a user orders for a product that is sold by the dealer.
- ☐ Deliver goods to the customer after picking the goods from the dealer.
- ☐ Collecting the goods from the customer which he/she wants to return back to the Food Store
- ☐ Deliver back the goods to the dealer that has been returned by the customer.

○ **The Technician must:**

- ☐ Work on the reported issues
- ☐ Update the status to the ticket
- ☐ Close the completed tickets.

Name	Type	Description
------	------	-------------

Account Registration	Process	First time users register and create their accounts by providing their details like First Name, Last Name, Password, contact details, email and card details.
Add items to cart	Process	Users can perform this process of adding products to cart.
Answer User issues	Process	Technician performs this process of answering users' issue.
Authenticate Transaction	Process	Bank performs this process of authenticating transaction by verifying the card details provided by the user.
Bank	Support Role	Verify credit card information and authenticate transaction.
Bank	Function	Execute the payment transaction.
Checkout	Process	Users can perform this process of updating/deleting products in cart and placing the order by choosing address and payment methods.
Create Ticket	Process	Technician performs this process based on the users' issues.
Database Admin	Role	Update and manage the database.
<u>Database Admin Information</u>	<u>Object</u>	<u>Database Admin information like username and password for login is stored here.</u>

<u>Dealer</u>	<u>Role</u>	<ul style="list-style-type: none"> • <u>Add new products.</u> • <u>Replenish the quantity of existing products.</u> • <u>Remove old stock.</u>
<u>Dealer Information</u>	<u>Object</u>	<u>Dealer's information like username and password for login is stored here.</u>
<u>Deliver Products</u>	<u>Process</u>	<u>Delivery Admin performs this process of delivering the products ordered to the users.</u>
<u>Deliver Returned Products</u>	<u>Process</u>	<u>Delivery Admin performs this process of delivering the returned products back to the dealer.</u>
<u>Delivery Admin</u>	<u>Role</u>	<ul style="list-style-type: none"> • <u>Pickup products</u> • <u>Deliver the products</u> • <u>Pickup returned products</u> • <u>Deliver returned products.</u>
<u>Delivery Admin Information</u>	<u>Object</u>	<u>Delivery Admin information like username and password for login is stored here.</u>
<u>Edit cart</u>	<u>Process</u>	<u>User performs this process of editing the products in the cart or their quantity.</u>
<u>Edit information</u>	<u>Process</u>	<u>User performs this process of editing personal info or card details.</u>

<u>Handle Reported Issues</u>	<u>Process</u>	<ul style="list-style-type: none"> • <u>Work on the reported issues.</u> • <u>Update the status of incident tickets.</u> • <u>Close incident tickets.</u>
<u>Login</u>	<u>Process</u>	<u>Process of logging in by providing the username and password. Users, Dealer, Delivery Admin, Database Admin, Support and Technician perform this process of logging to the portal.</u>
<u>Manage Database</u>	<u>Process</u>	<u>Database Admin performs this process of managing the database.</u>
<u>Pickup Products</u>	<u>Process</u>	<u>Delivery Admin performs this process of picking up the products ordered by the users from the dealer.</u>
<u>Pickup Returned Products</u>	<u>Process</u>	<u>Delivery Admin performs this process of picking up the products returned by the users.</u>
<u>Edit cart</u>	<u>Process</u>	<u>User performs this process of editing the products in the cart or their quantity.</u>
<u>Edit information</u>	<u>Process</u>	<u>User performs this process of editing personal info or card details.</u>
<u>Handle Reported Issues</u>	<u>Process</u>	<ul style="list-style-type: none"> • <u>Work on the reported issues.</u> • <u>Update the status of incident tickets.</u> • <u>Close incident tickets.</u>

<u>Login</u>	<u>Process</u>	<u>Process of logging in by providing the username and password. Users, Dealer, Delivery Admin, Database Admin, Support and Technician perform this process of logging to the portal.</u>
<u>Manage Database</u>	<u>Process</u>	<u>Database Admin performs this process of managing the database.</u>
<u>Pickup Products</u>	<u>Process</u>	<u>Delivery Admin performs this process of picking up the products ordered by the users from the dealer.</u>
<u>Pickup Returned Products</u>	<u>Process</u>	<u>Delivery Admin performs this process of picking up the products returned by the users.</u>

A.1.5 ACTOR DICTIONARY

<u>Name</u>	<u>Description</u>
<u>Bank</u>	<u>Verify Credit Card information.</u> <u>Authenticate and complete payment/transaction.</u>
<u>Database Admin</u>	<u>Update and manage the database.</u>
<u>Dealer</u>	<u>Add new products.</u> <u>Replenish the quantity of existing products.</u> <u>Remove old stock.</u>
<u>Delivery Admin</u>	<u>Pickup products ordered by the user from the dealer.</u> <u>Deliver the products ordered to the user.</u> <u>Pickup products returned by the user.</u> <u>Deliver the products returned to the dealer.</u>
<u>Support Staff</u>	<u>Resolve/answer to customer's queries and issues through chat or email.</u> <u>Create incident tickets and assign incident tickets to the respective technicians.</u>
<u>Technician</u>	<u>Work on the reported issues.</u> <u>Update incident tickets.</u> <u>Close incident tickets.</u>
<u>User</u>	<u>Create and register account.</u> <u>View products and their specifications.</u> <u>Search products.</u>

Add/Delete/Update items to cart.

Checkout items in cart.

Add/Edit address.

Make payment.

Choose payment methods.

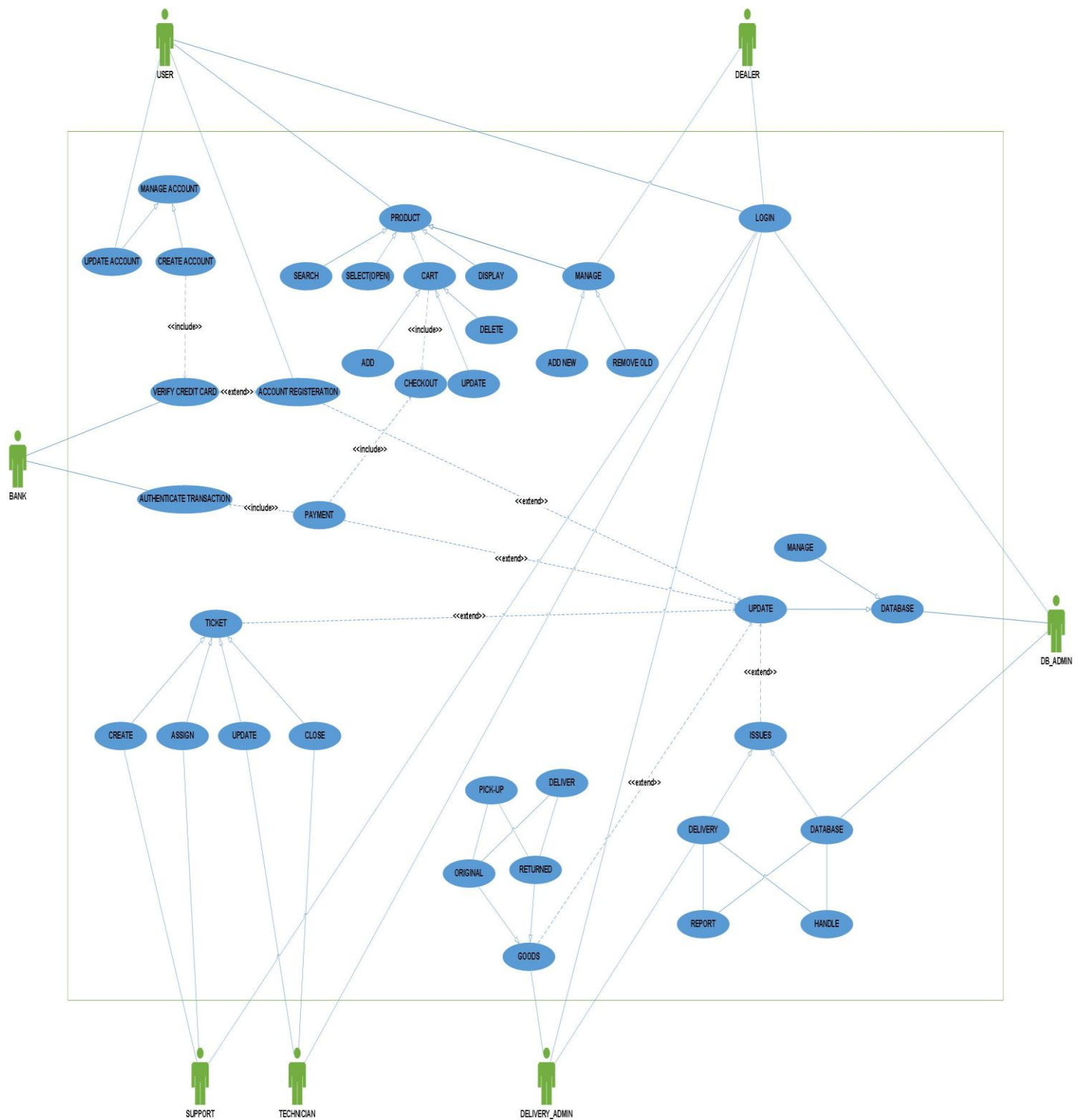
Manage and edit account.

Use technical assistance.

A.1.6 USE CASES

<u>Names</u>	<u>Names</u>
Manage Account -Create Account -Update Account	Goods -Original -Returned Pick-Up Deliver
Product -Search -Select -Display -Manage -Add new -Remove old -Cart -Add -Update -Delete -Checkout	Issues -Delivery -Database Handle Report Database -Manage -Update
Verify credit card	Login
Ticket -Create -Assign -Update -Close	Account Registration Authenticate transaction Payment

A.1.7 USE CASE DIAGRAM

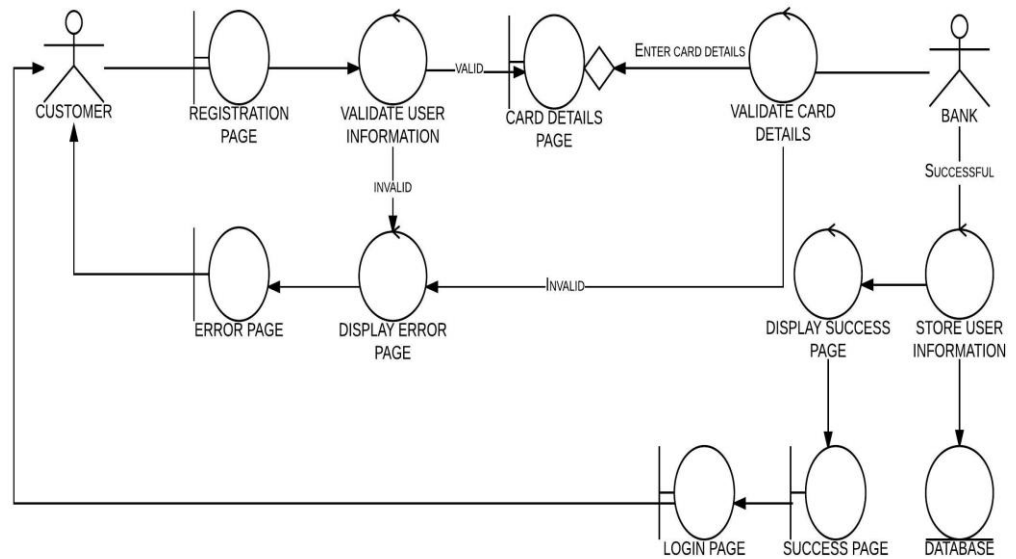


APPENDIX: B

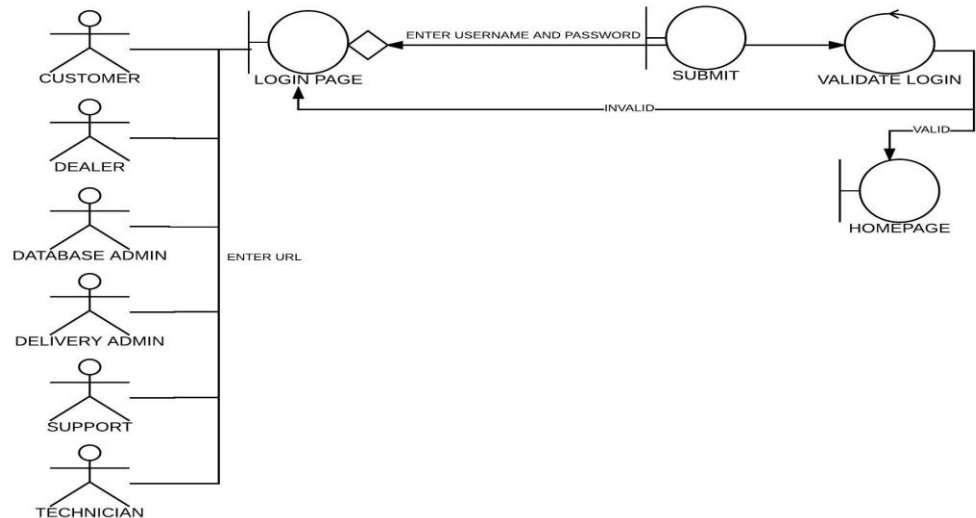
B.1 PHASE II

B.1.1.1: ANALYSIS MODEL

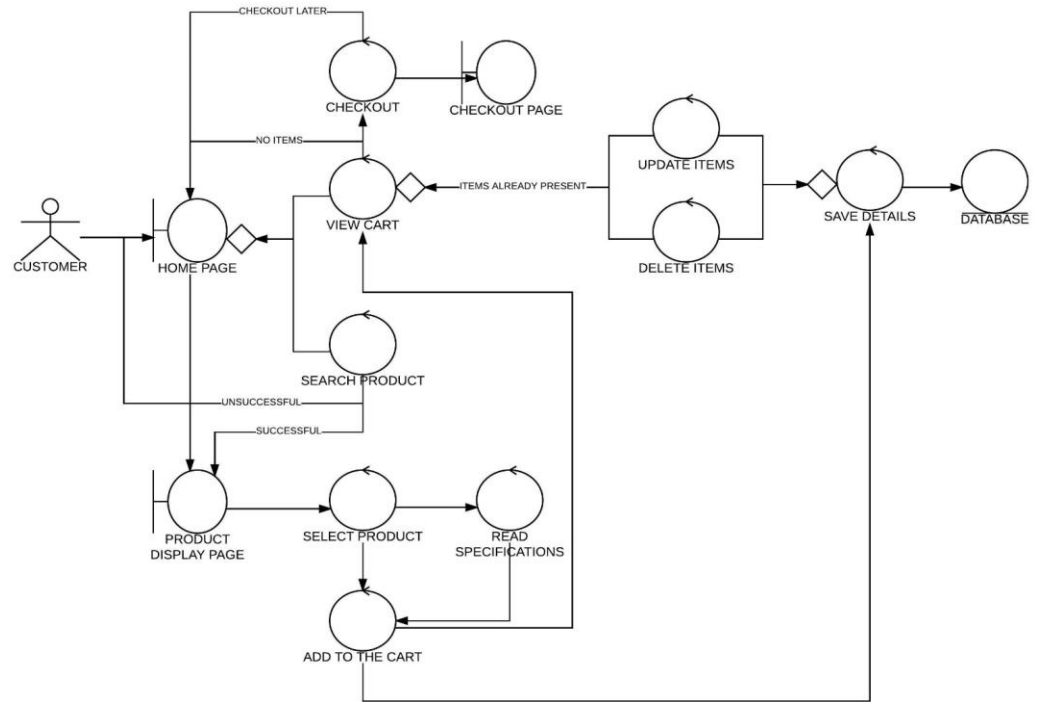
REGISTRATION:



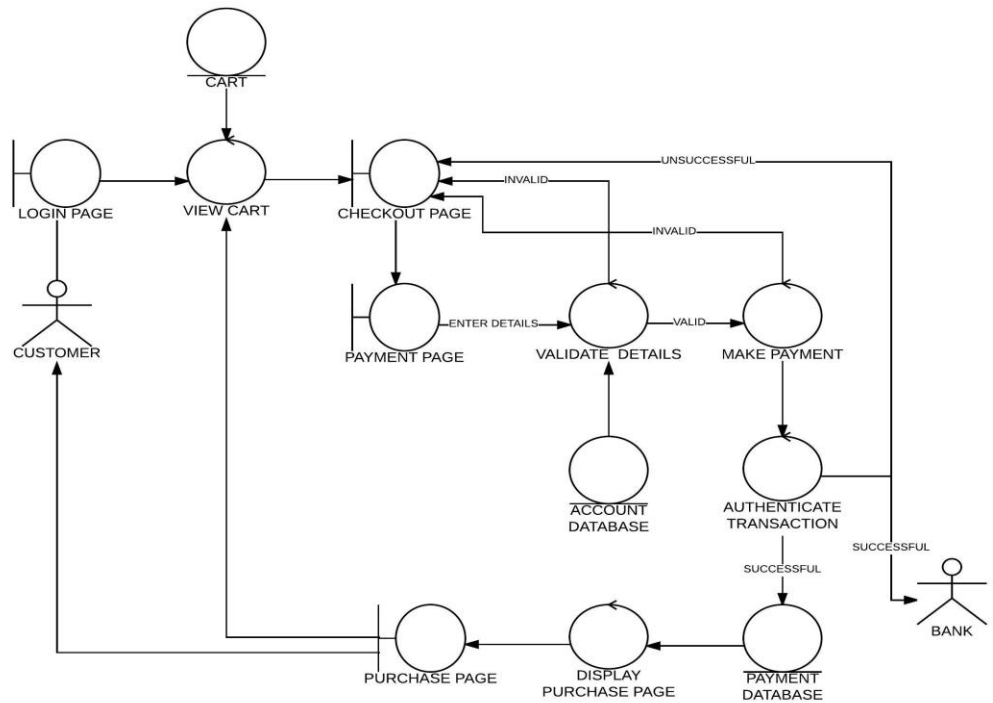
LOGIN:



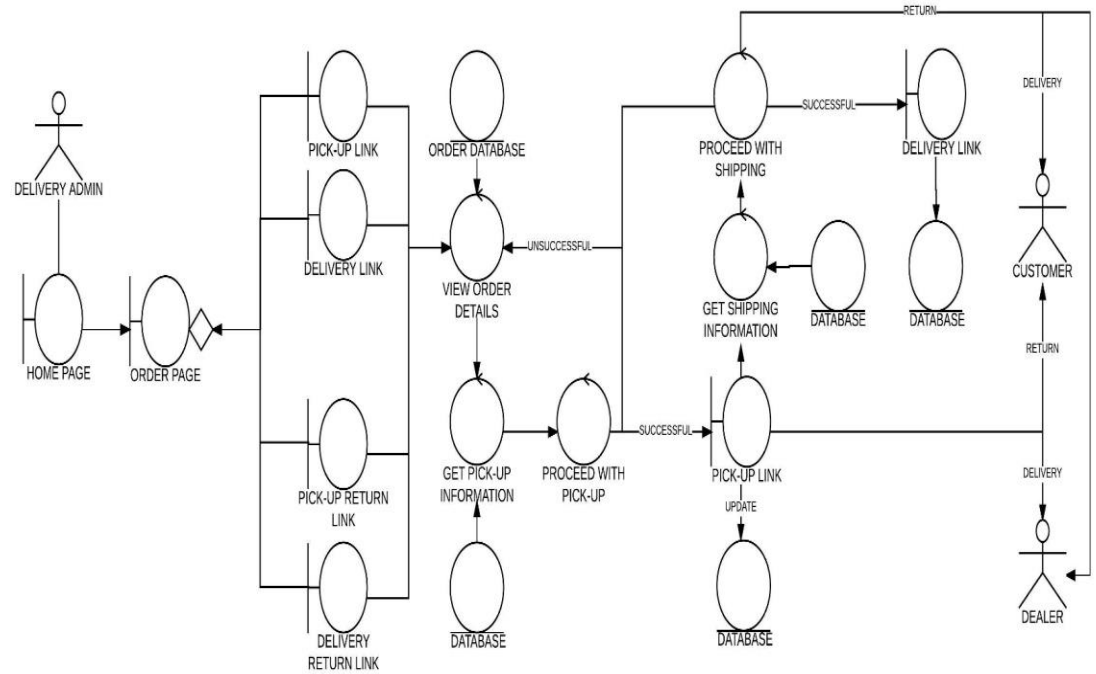
PRODUCT:



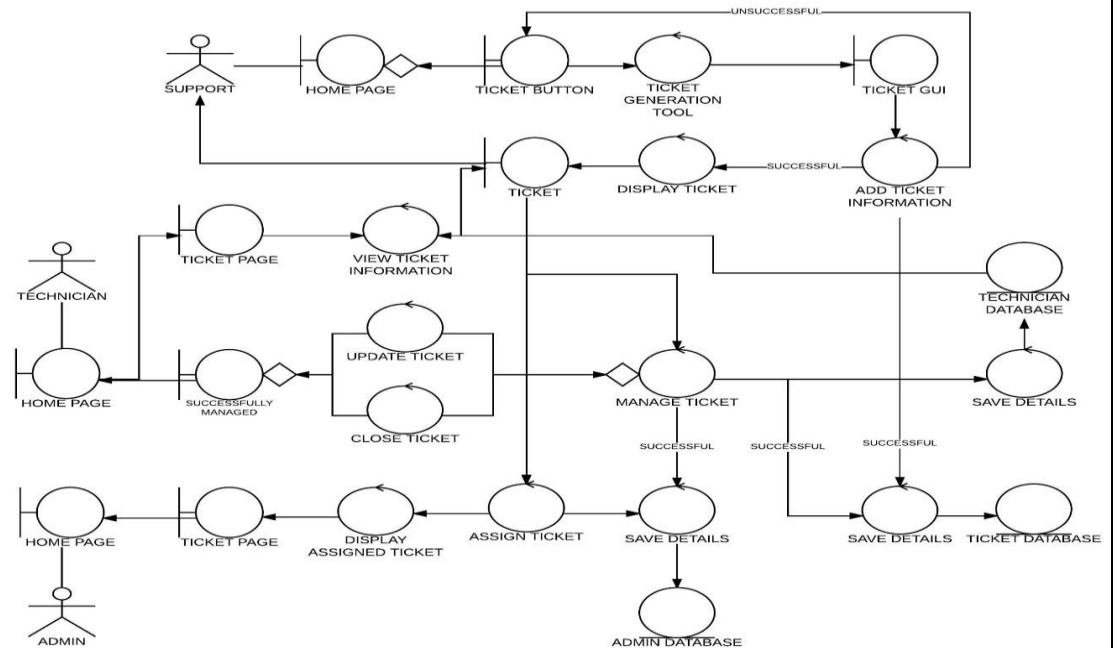
PAYMENT:



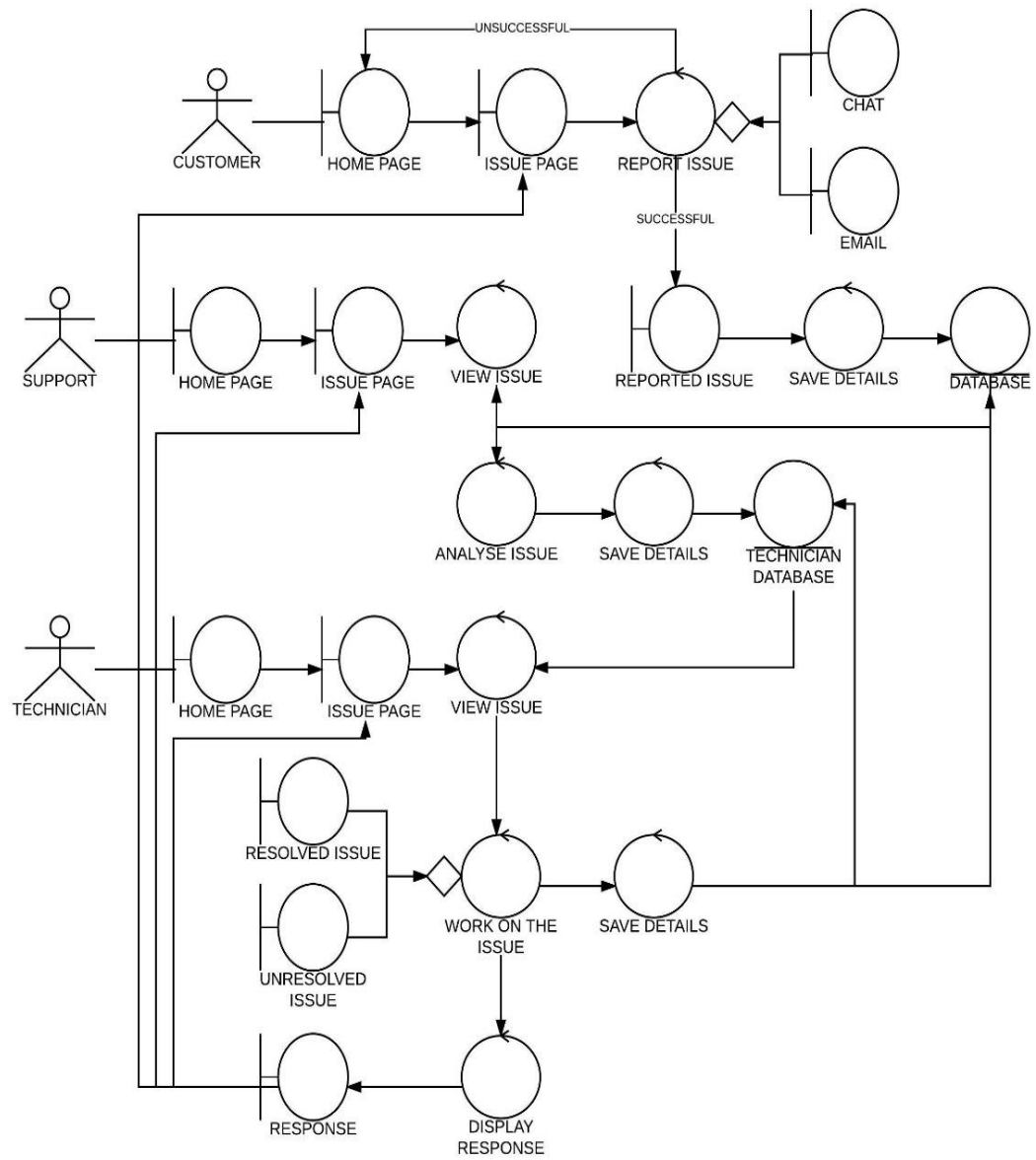
DELIVERY:



TICKET:

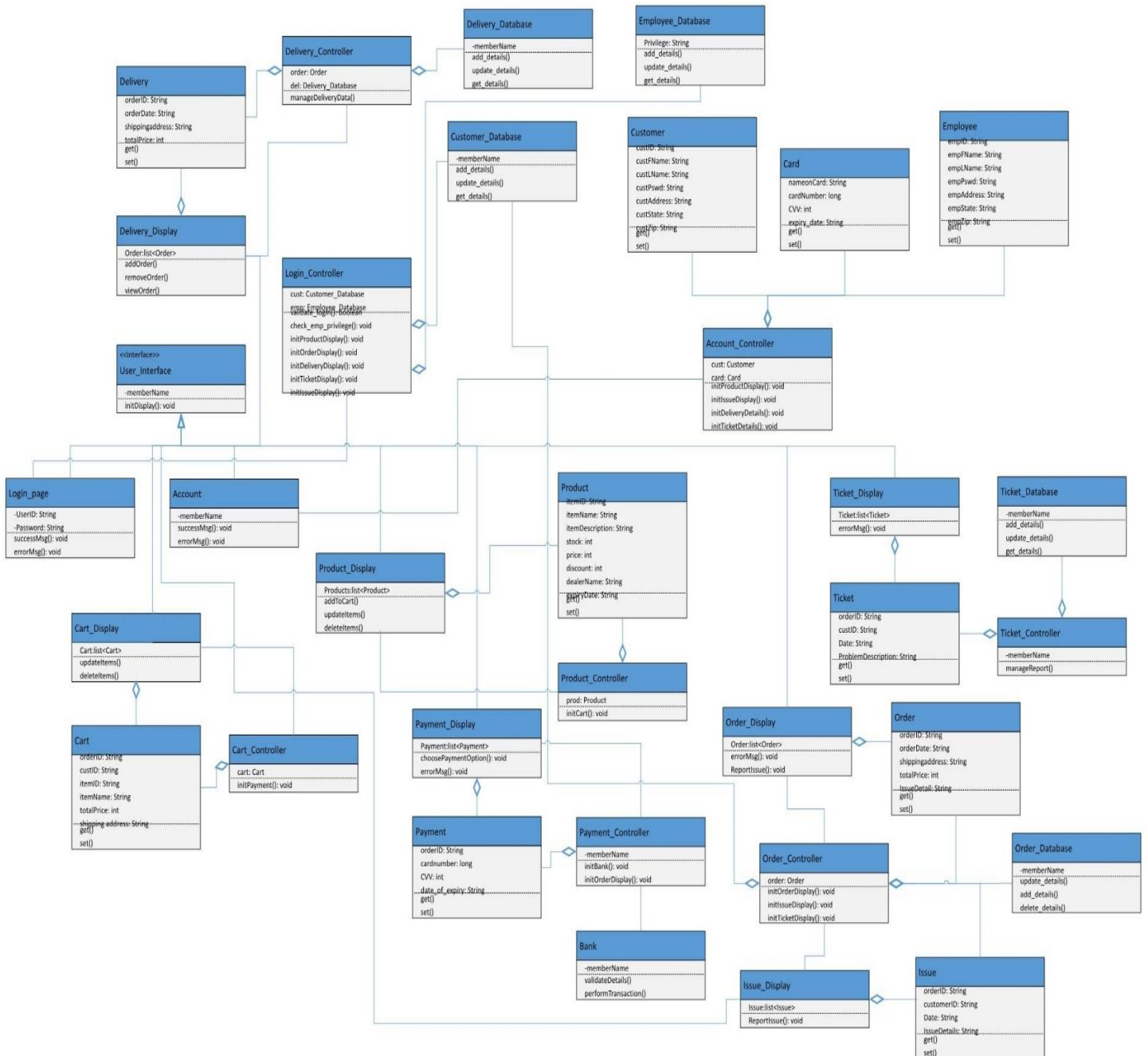


ISSUE:

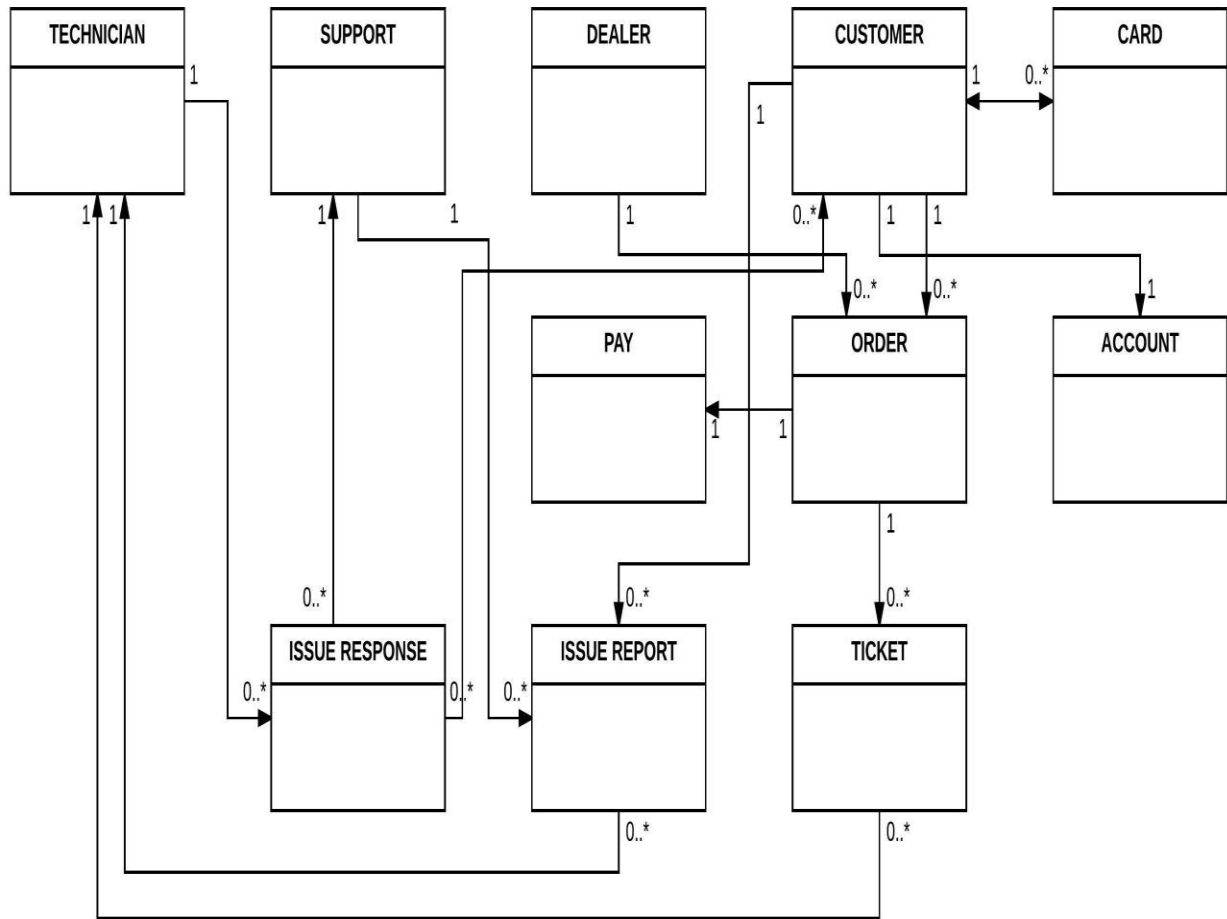


B.1.2: DESIGN MODEL

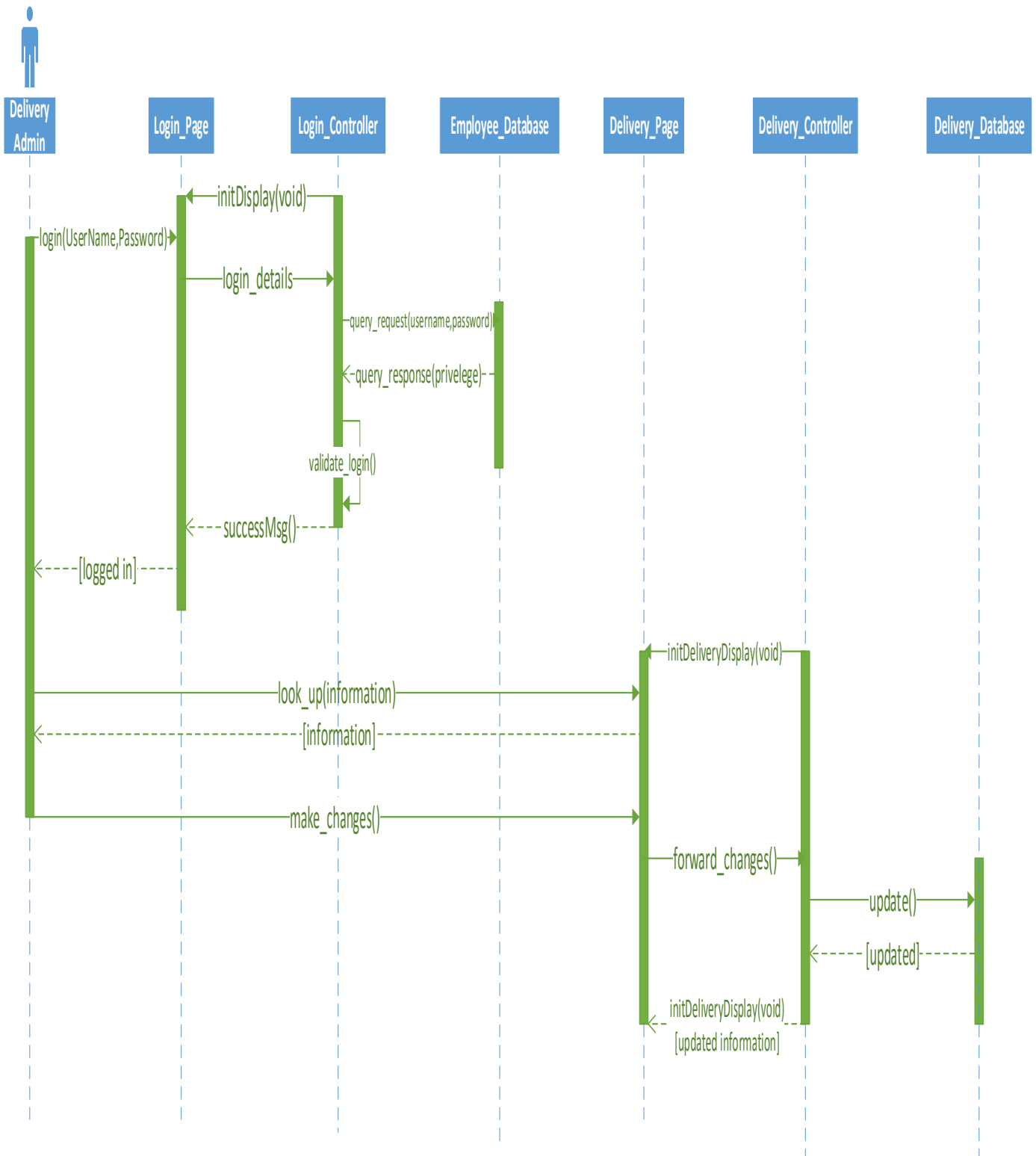
B.1.2.1 CLASS DIAGRAM

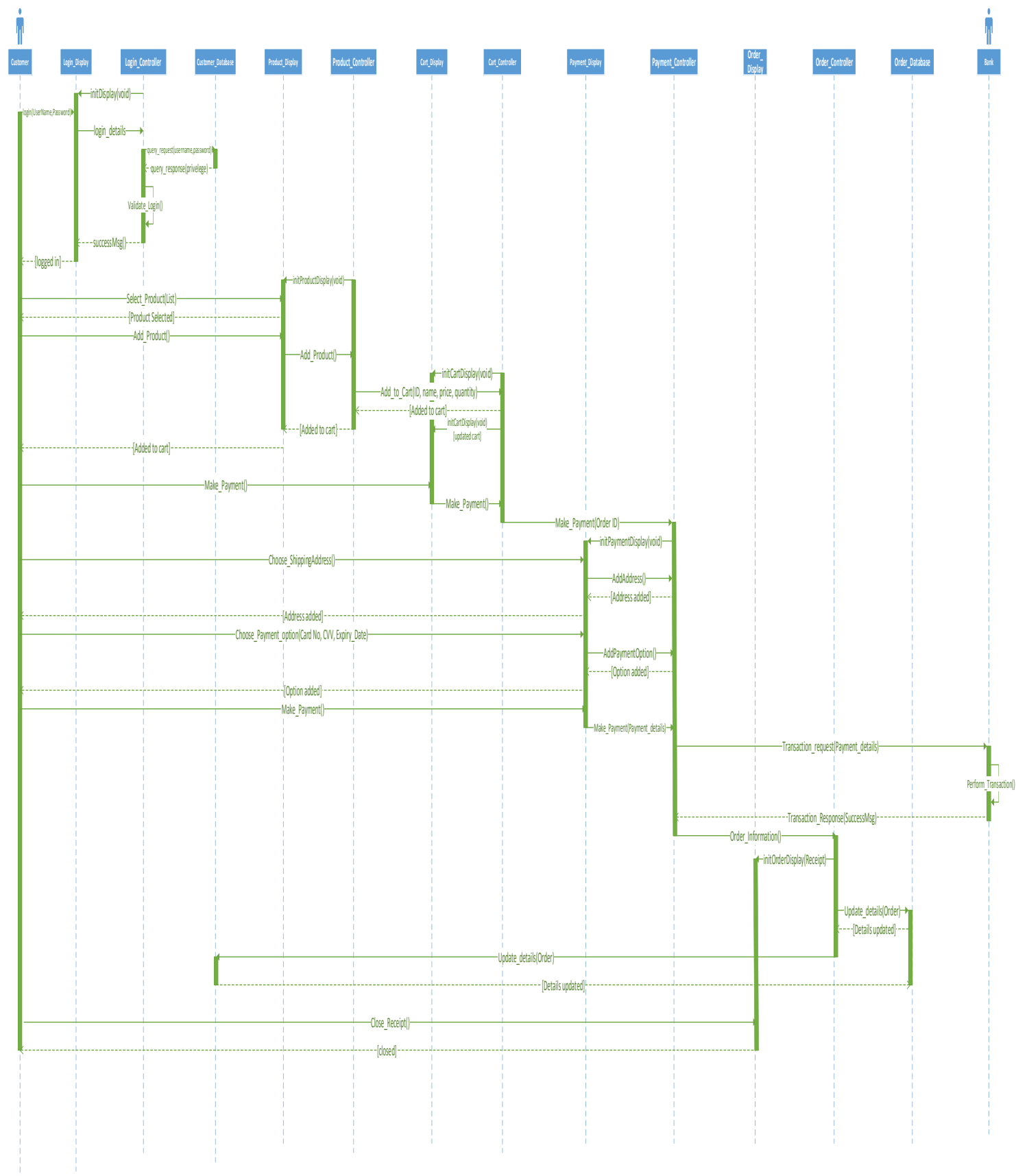


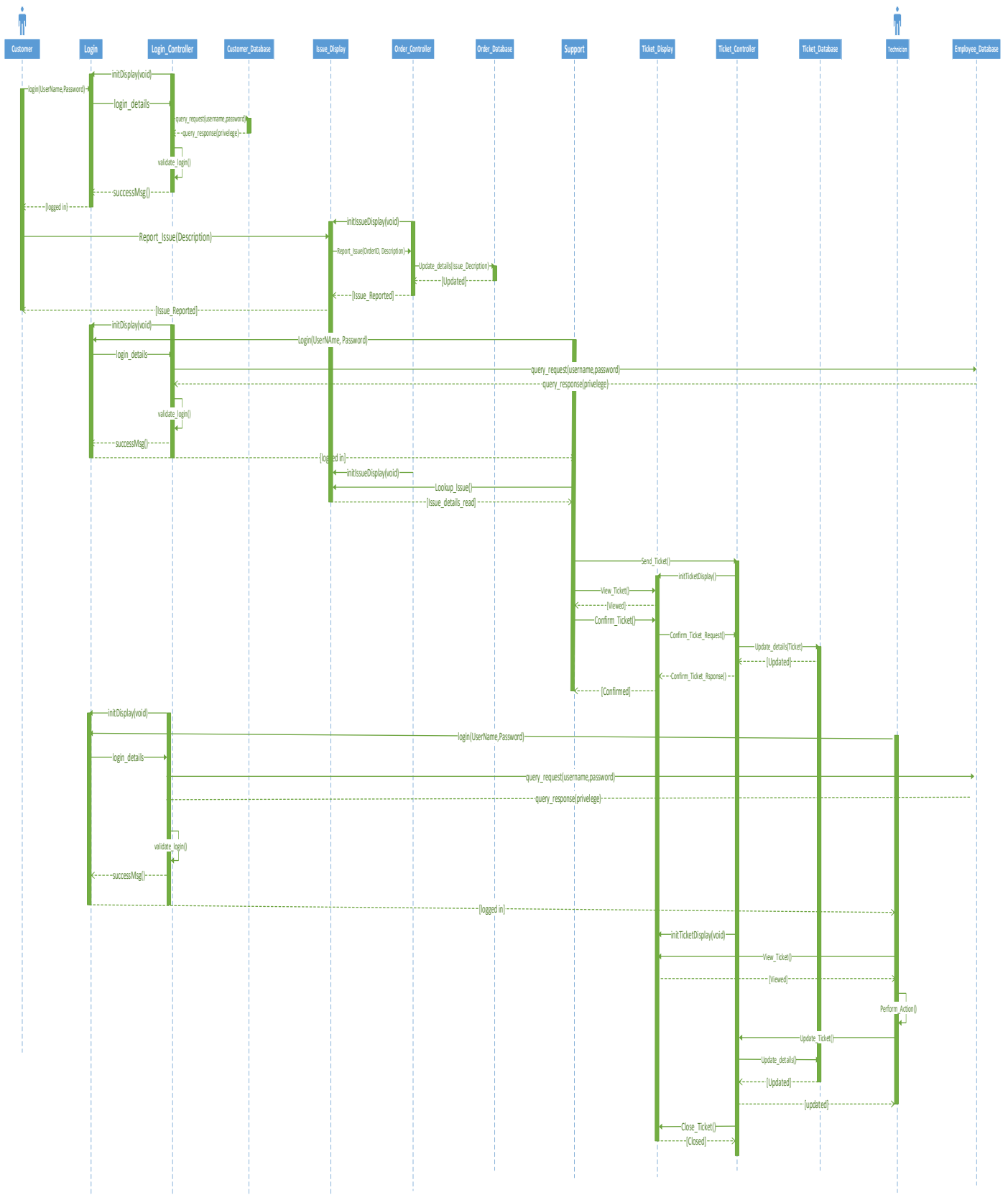
B.1.2.2 OBJECT DATA MODEL



B.1.3: SEQUENCE DIAGRAM

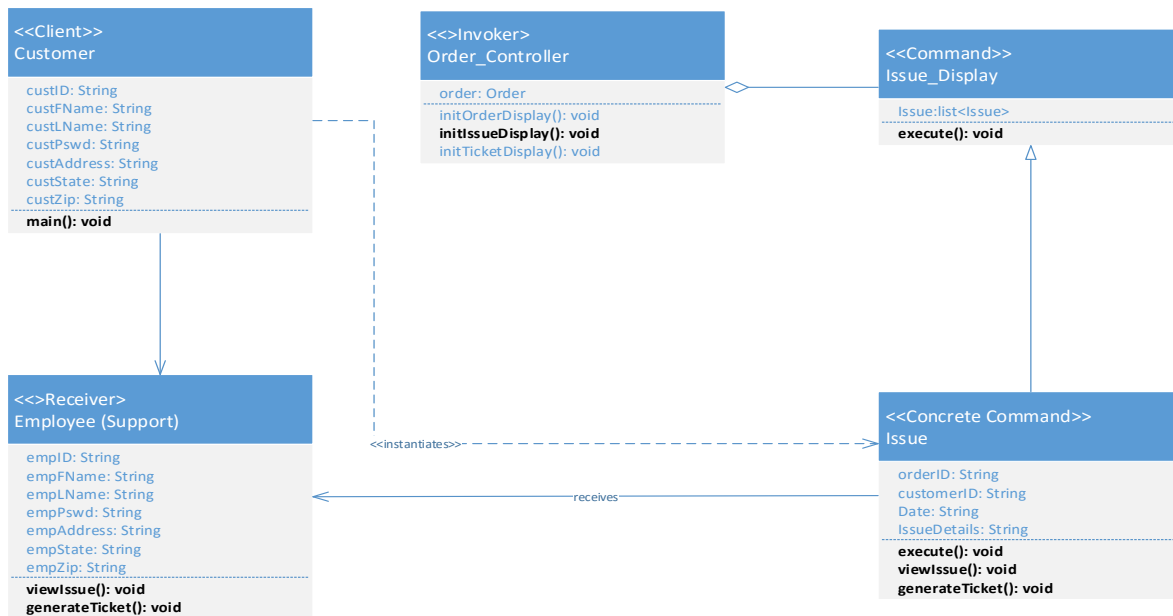




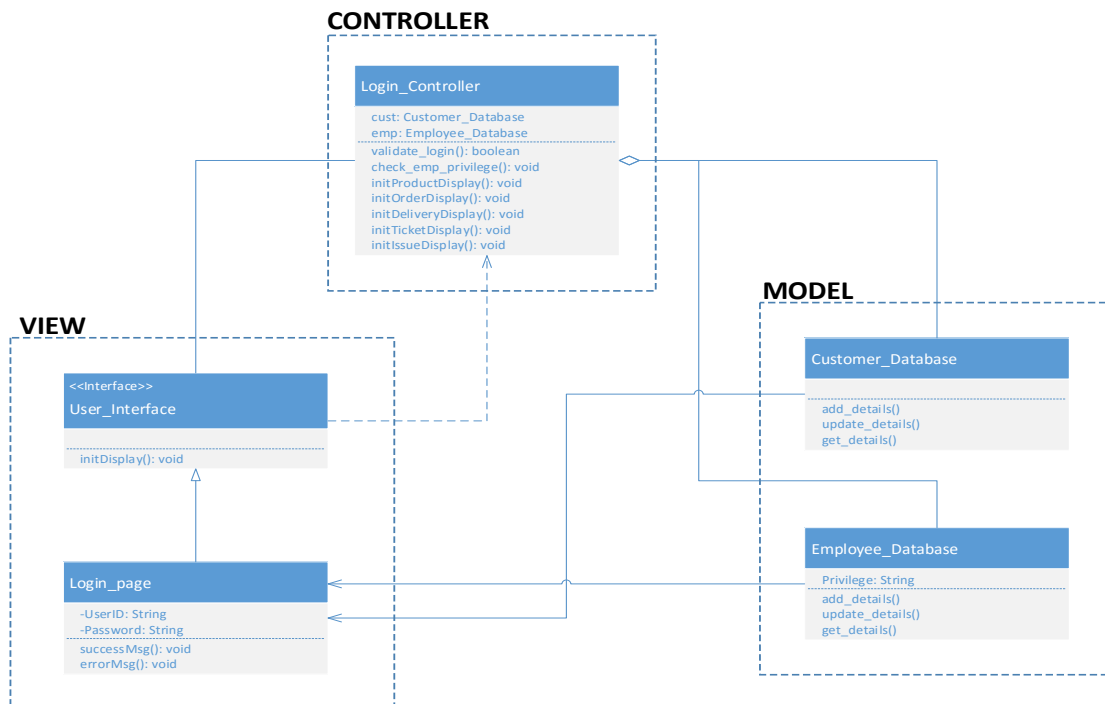


B.1.4 DESIGN PATTERNS

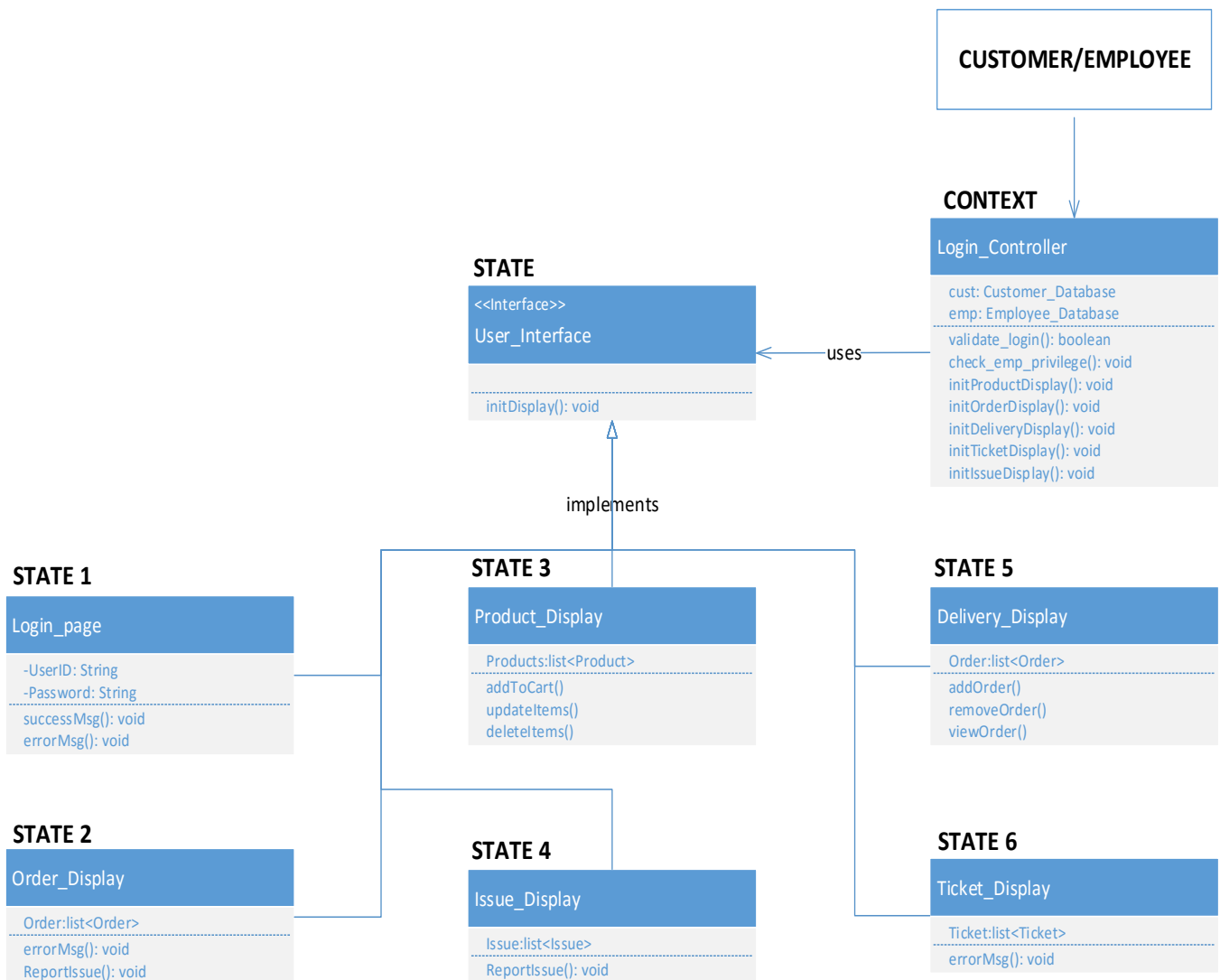
B.1.4.1 COMMAND PATTERN



B.1.4.2 MODEL-VIEW-CONTROLLER PATTERN



B.1.1.4.3 STATE PATTERN



APPENDIX: C

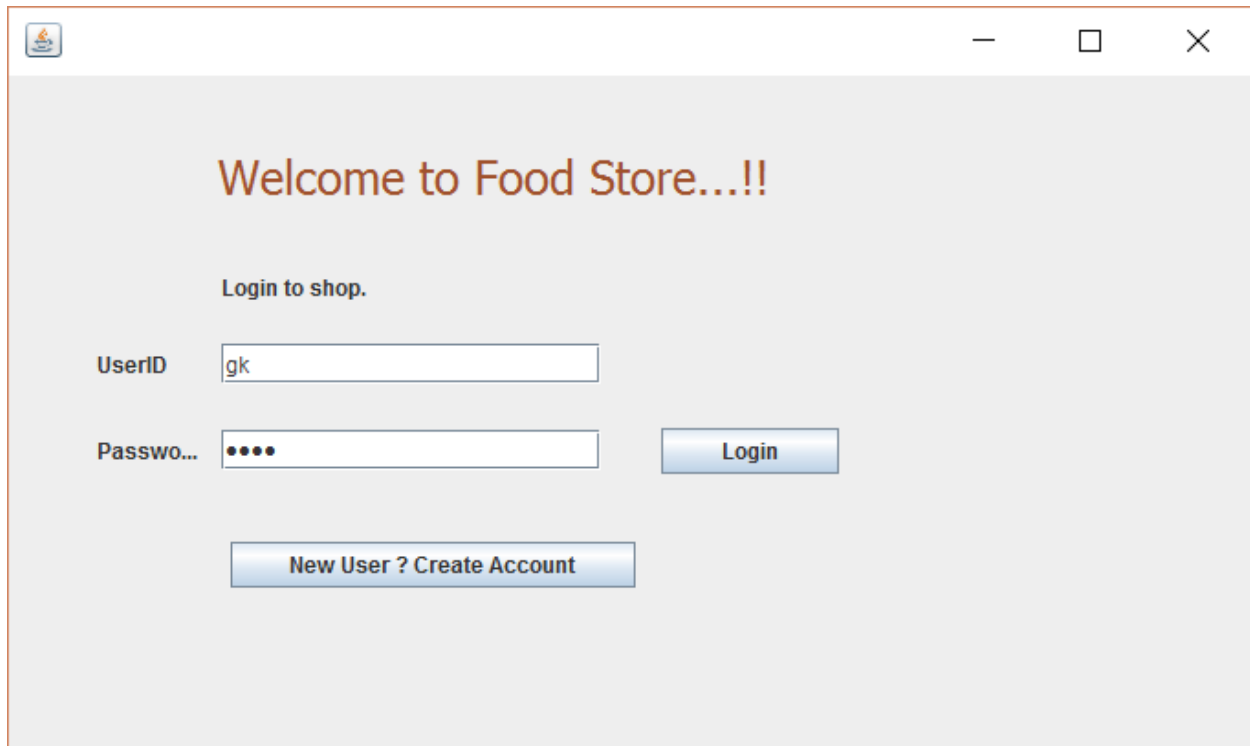
C.1 PHASE III

C.1.1 IMPLEMENTATION

(Attachment)

C.1.2 SCREENSHOTS

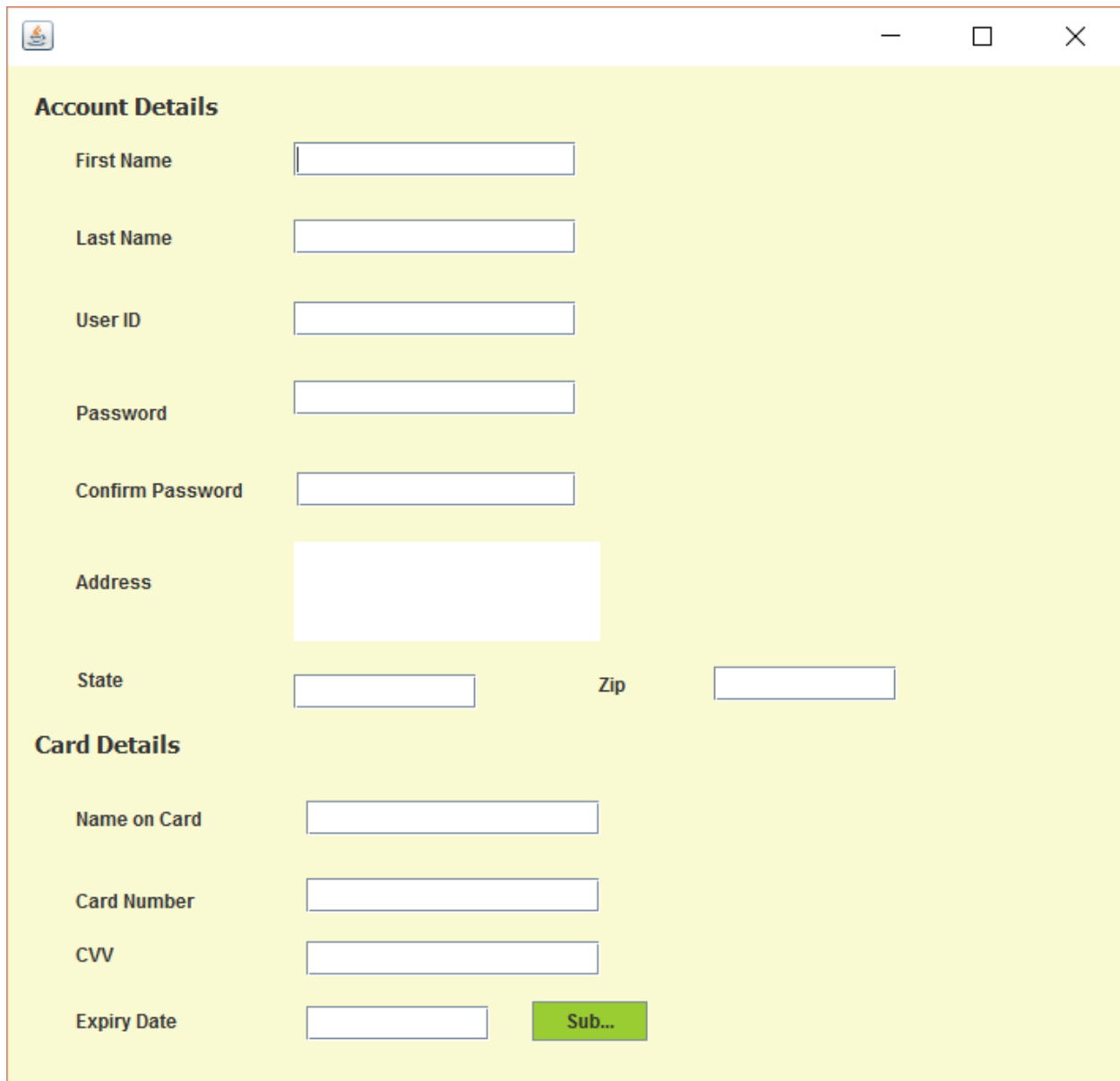
1. Login Screen



First screen for the entire application. Once the user is registered, sign in is going to be very easy. User has to give the registered userID and Password to sign in to access the application.

Login Screen has 2 strategies one from customers and other for employees. The Employee table holds the list of employees registered and the privileges that they have.

2. New Customer Details – (Registration Screen)

A screenshot of a web application registration form. The form is titled "Account Details" and "Card Details". It contains several input fields for user information, including First Name, Last Name, User ID, Password, Confirm Password, Address, State, Zip, Name on Card, Card Number, CVV, and Expiry Date. A green "Sub..." button is located at the bottom right of the form.

Account Details

First Name

Last Name

User ID

Password

Confirm Password

Address

State Zip

Card Details

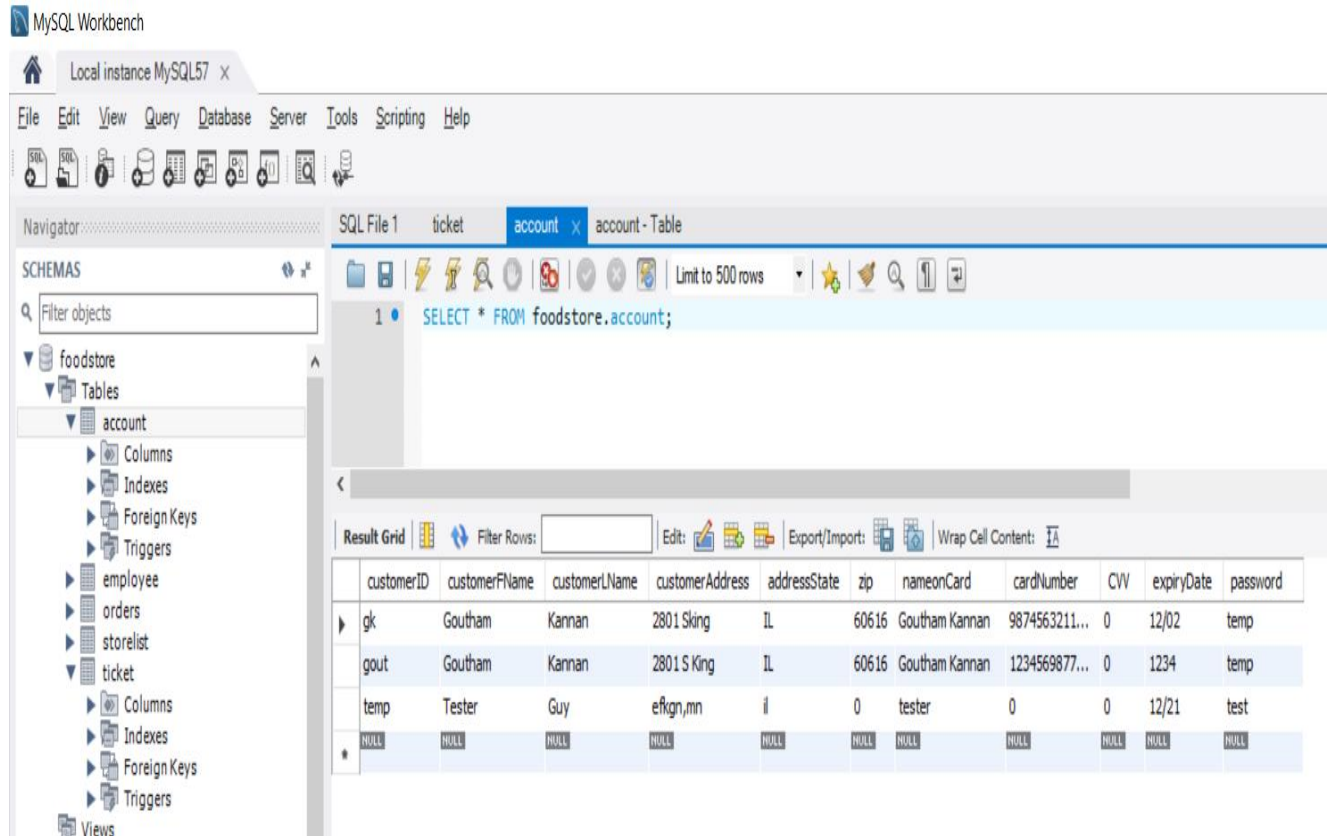
Name on Card

Card Number

CVV

Expiry Date

The details that are provided by the user in the above form will be saved in the Table accounts. The object customer which holds the customer information is converted into a row of the table. This conversion is shown in the below snap of the mysql workbench.







3. Shop Screen

By default, the retailer must have some stock in their inventory, this data is read by a Sql select query and it is associated to a class called storeList. The instance this class is bound with TabelModel of Swing.

Welcome to Food Store gk

Instructions to shop: 1. Add Products to Cart ----> 2. Check out Cart ----> 3.Make Payment

Shop Checkout Payment

Product Name	Description	icon	stock	Price	Discount	Add to checkout	Select Qty
CoffeMate	Non Diary Creamer		8	4	10	<input type="checkbox"/>	1
Cookies	Home made chocochip cookie		7	2	20	<input type="checkbox"/>	1
Planter Peanuts	Sea Salted Peanuts		10	12	10	<input type="checkbox"/>	1
Oreo	Classic sandwich biscuits		2	12	10	<input type="checkbox"/>	1

Checkout

Help

MySQL Workbench

Local instance MySQL57 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SQL File 1 ticket account account - Table storelist x orders employee orders tick

1 • SELECT * FROM foodstore.storelist;

Result Grid

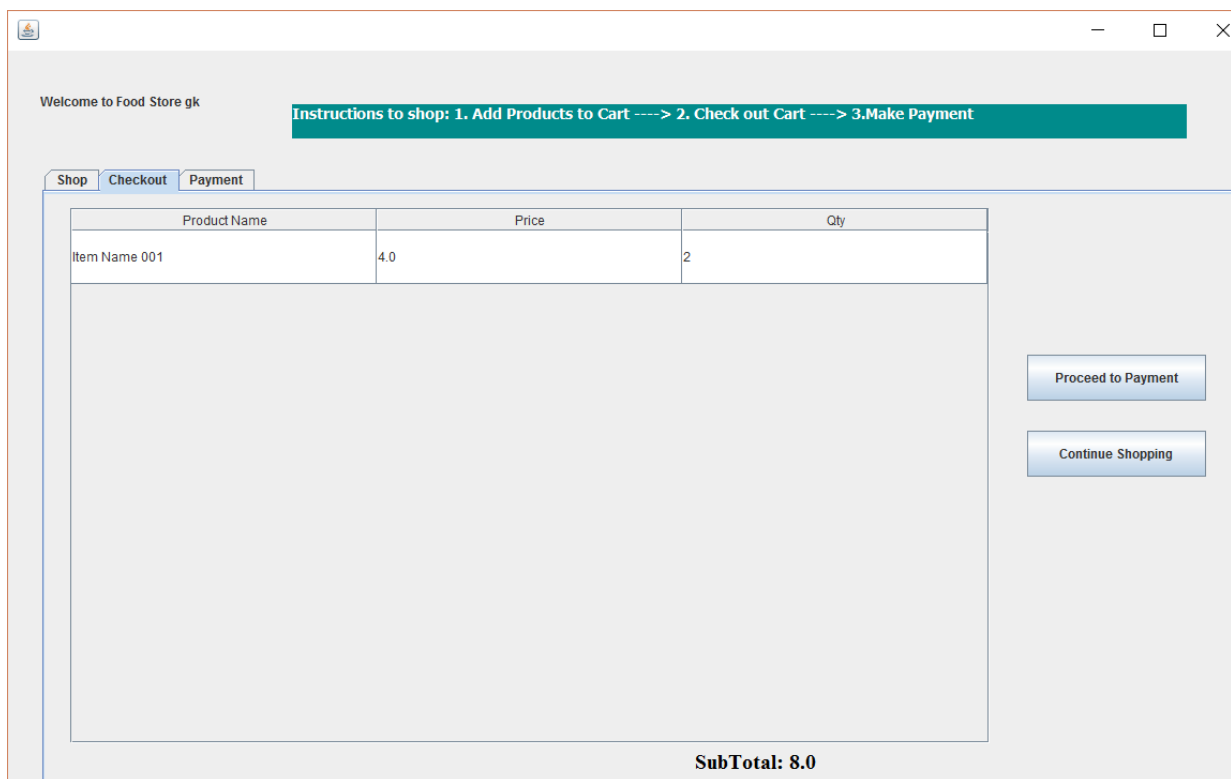
itemID	itemName	itemDescription	stock	price	discount	dealerName	isnew	icon
item_001	CoffeMate	Non Diary Creamer	8	4.99	10	John	y	BLOB
item_002	Cookies	Home made chocochip ...	7	2.99	20	Red	y	BLOB
item_003	Planter Peanuts	Sea Salted Peanuts	10	12.99	10	Snow	y	BLOB
item_004	Oreo	Classic sandwich biscuits	2	12.99	10	Ned	y	BLOB
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The Swing's default model supports only text data, in order to hold the icon, check box and make the cells editable to user, the base tablemodel is customized by inheriting Abstract Table Model. The event listeners allow the users to select the items as well as change the quantities.

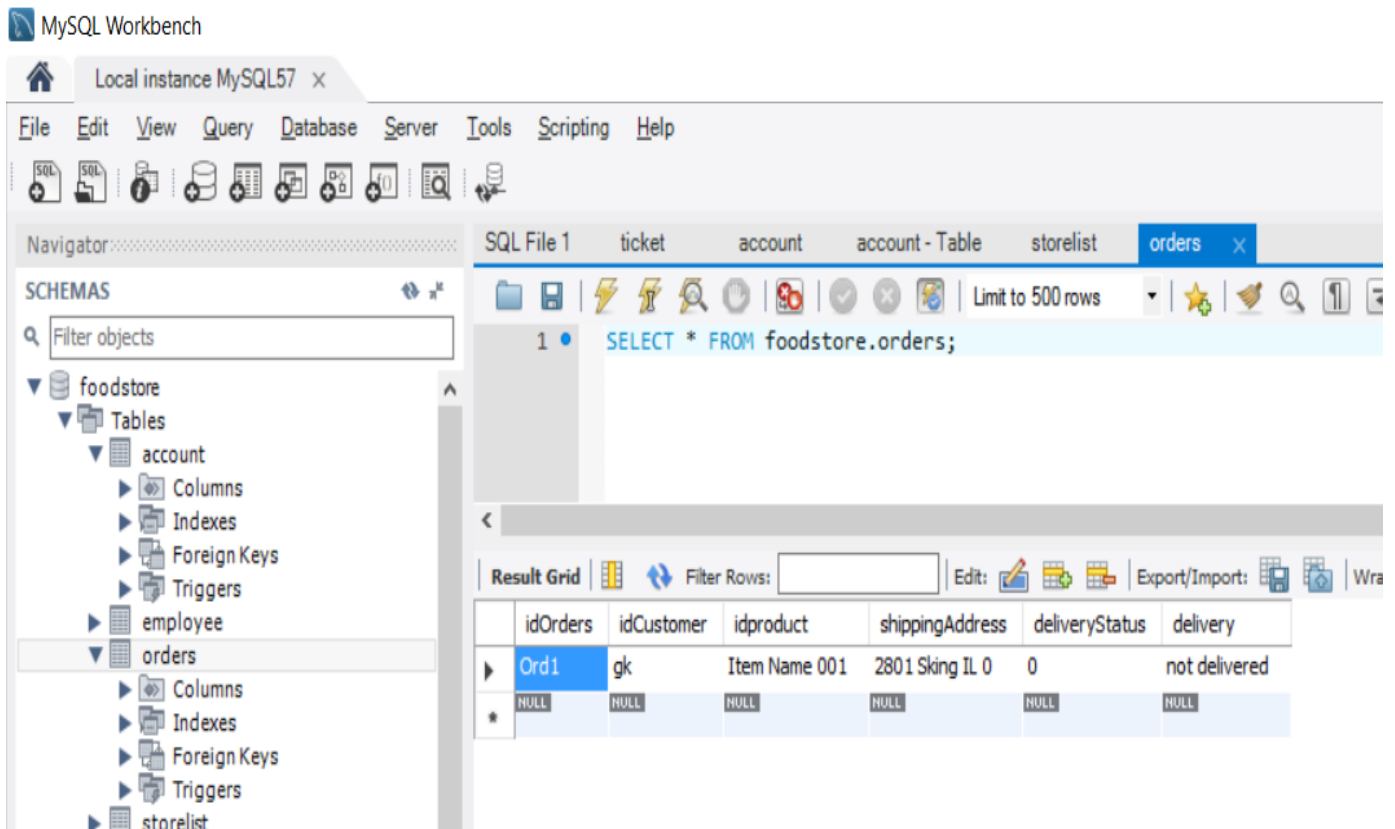
The general flow is to click on the checkout button that carry forwards the entries to the next tab, (i.e) the checkout page.

4. CheckOut Screen

Upon clicking the checkout button the next tab in JTabbedPane opens up containing the details of the selected items. It also displays the sub total of all the items in the checkout. Customer either can proceed to payment to place an order or can continue with the shopping again.



The selected order is translated into table called orders. The schema of the order table can be seen below.



5. Payment Screen

On proceeding with checkout, the next tab aids in making the payment either by card/cash. We have again used JTabbedPane for different payment options for easier access. Customer can cancel the Payment at any time before confirmation so that they will be able to review the checked out items again and continue shopping again.

Cash On Delivery –

Welcome to Food Store shan

Instructions to shop: 1. Add Products to Cart ----> 2. Check out Cart ----> 3.Make Payment

ShopCheckoutPayment

CashCredit

Amount Payable On Delivery

Extra COD charges (\$5) will be applied for this option

Shipping Address

2801 S king drive IL 0

Tip:-
Avoid extra COD charges when you pay by Credit Card or Debit Card

Amount Payable: 8.0

Make Payment

Cancel

LOGOUT

Card Payment -

Welcome to Food Store shan

Instructions to shop: 1. Add Products to Cart ----> 2. Check out Cart ----> 3.Make Payment

ShopCheckoutPayment

CashCredit

Card Number

Name on the Card

Goutham

Expiration Date

02/21

CVV

Shipping Address

2801 S king drive IL 0

Amount Payable: 8.0

Make Payment

Cancel

LOGOUT

Order Confirmation Screen:

Welcome to Food Store shan

Instructions to shop: 1. Add Products to Cart ----> 2. Check out Cart ----> 3. Make Payment

Shop Checkout **Payment**

Cash Credit

Amount Payable On Delivery

Extra COD charges (\$5) will be applied for this option

Shipping Address

2801 S king drive IL 0

Tip:-
Avoid extra COD charges when you pay by Credit Card or Debit Card

Message X

i Order Has been Placed

OK

Amount Payable: 14.0

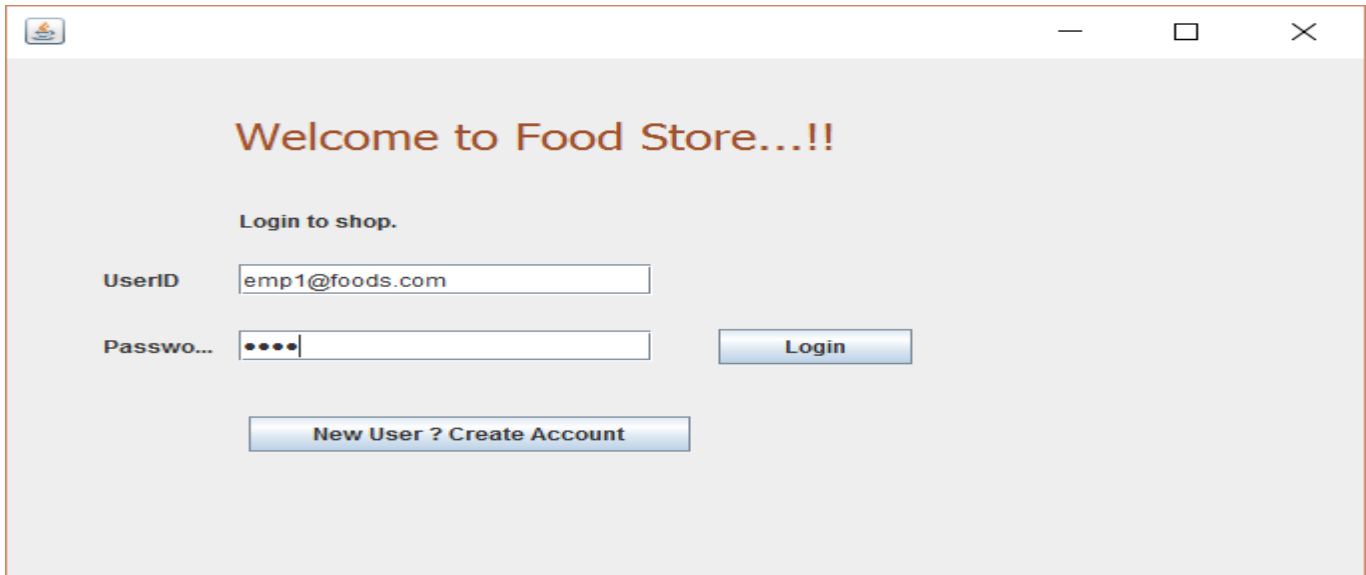
Make Payment

Cancel

LOGOUT

Once the customer confirms the payment, the order will be placed which then will be delivered to the customer's shipping address accordingly.

6. Employee Login Screen



Welcome to Food Store...!!

Login to shop.

UserID

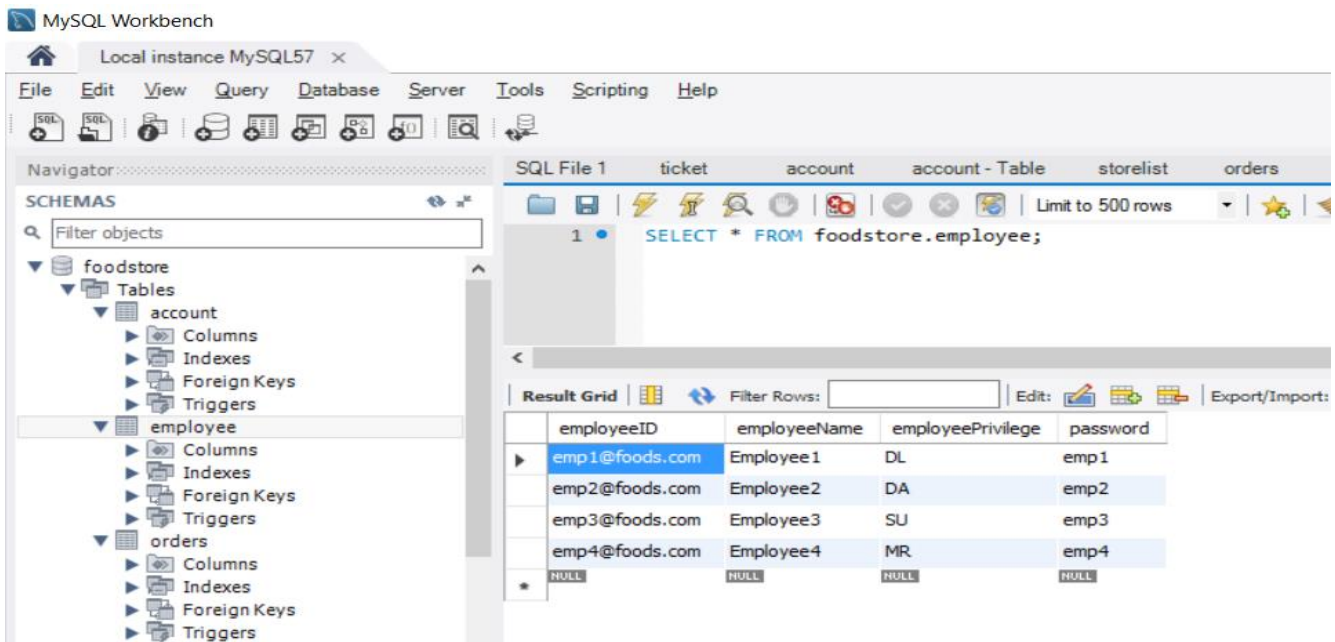
Passwo...

Login

New User ? Create Account

When employee login they use the same login interface but they have special username that ends with @foods.com.

Login Screen has 2 strategies one from customers and other for employees. The Employee table holds the list of employees registered and the privileges that they have.



MySQL Workbench

Local instance MySQL57 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

foodstore

- Tables
 - account
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - employee
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - orders
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers

SQL File 1 ticket account account - Table storelist orders

1 • SELECT * FROM foodstore.employee;

Result Grid

employeeID	employeeName	employeePrivilege	password
emp1@foods.com	Employee1	DL	emp1
emp2@foods.com	Employee2	DA	emp2
emp3@foods.com	Employee3	SU	emp3
emp4@foods.com	Employee4	MR	emp4
NULL	NULL	NULL	NULL

7. Dealer Login Screen:

When a dealer logs into the system, based on the privilege of the employee the application displays the below shown UI that allows the dealers to upload their product data like item name, id, description, icon, price, discounts.

Welcome emp1@foods.com

Item ID

Price

Save Details

Item Name

Quantity

1

Description

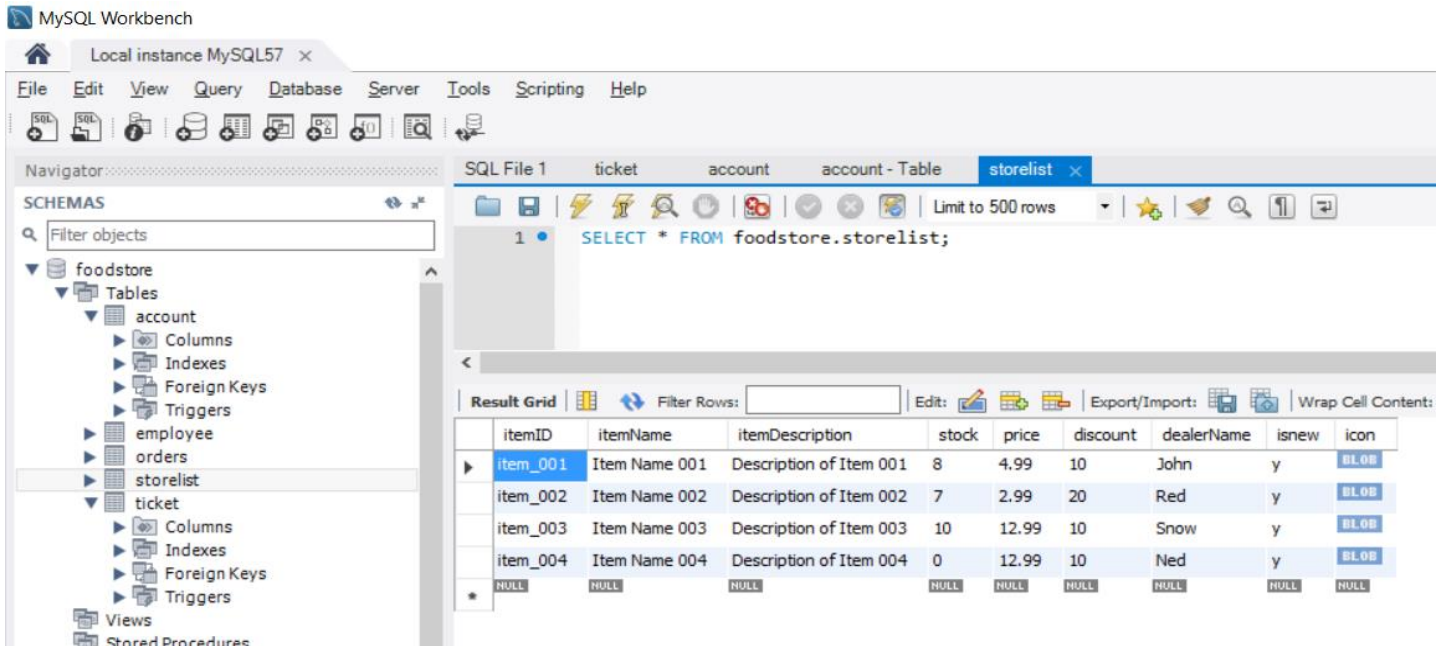
Discount

0

Icon

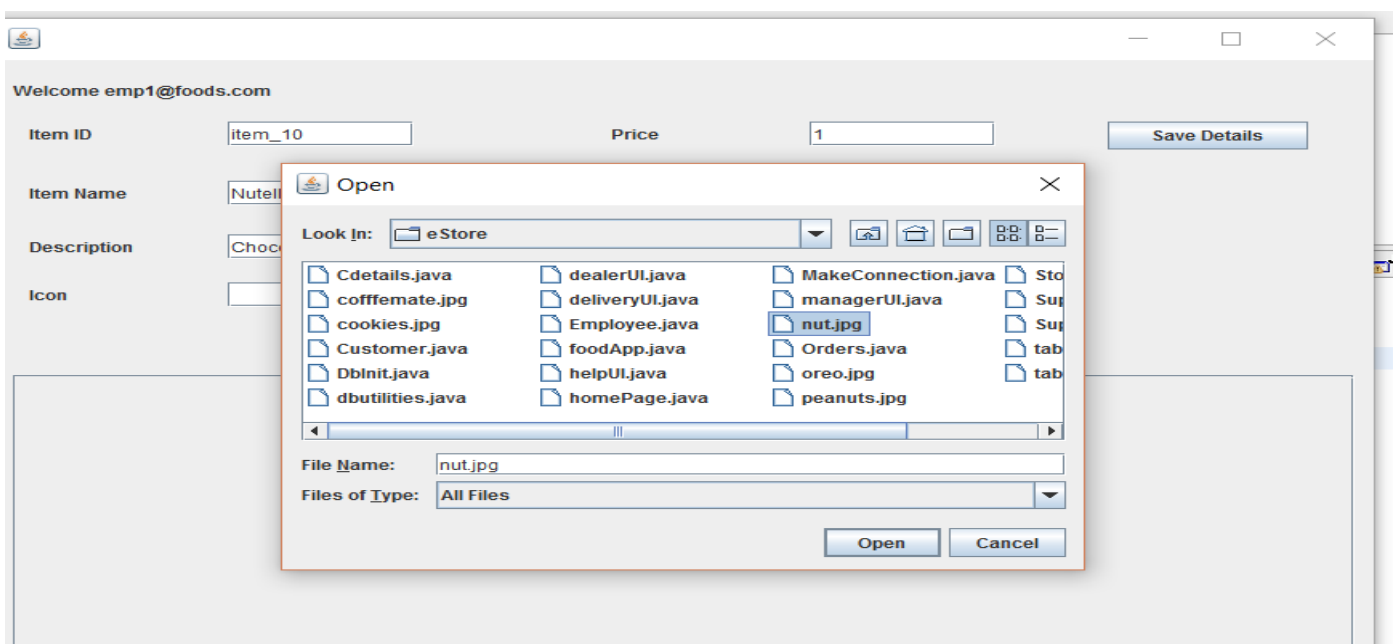
browse

itemID	itemName	stock	price	discount
item_001	Item Name 001	8	4.99	10
item_002	Item Name 002	7	2.99	20
item_003	Item Name 003	10	12.99	10
item_004	Item Name 004	0	12.99	10




8. Upload new data Screen:

It allows the dealer to browse through the files to upload the icons and dealer can enter the other details which will be stored in the orders table and new added items will be displayed to the customer when they log in.



9. Latest Stock UI Screen:

After entering the details on the text fields the dealer clicks the save details button, then the new data is inserted into the database. The updated data is displayed in the below table.

— □ ×

Welcome emp1@foods.com

Item ID

item_5

Price

3

Save Details

Item Name

Nutella

Quantity

1

Description

Chocolate Spread

Discount

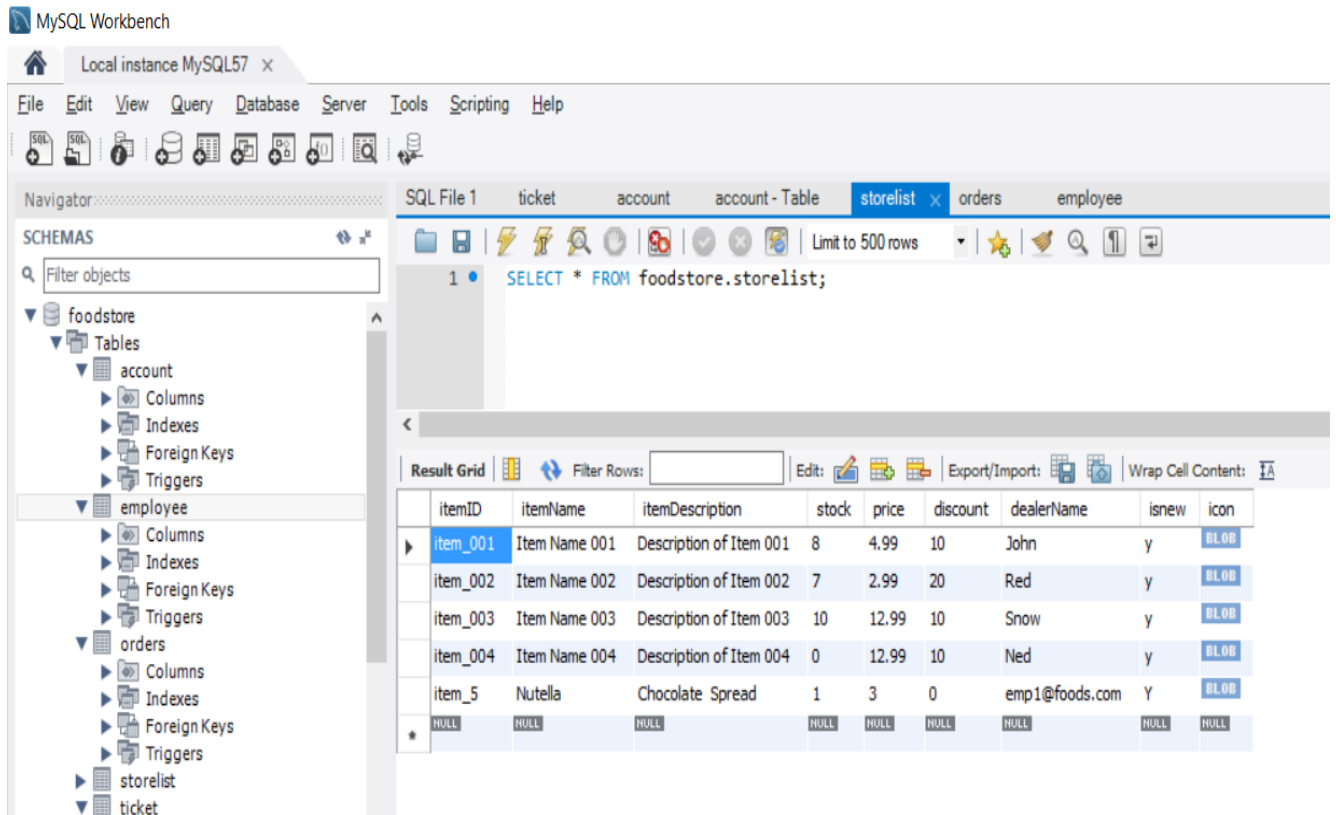
0

Icon

AD\SwingApp\src\Store\nut.jpg

browse

itemID	itemName	stock	price	discount
item_001	Item Name 001	8	4.99	10
item_002	Item Name 002	7	2.99	20
item_003	Item Name 003	10	12.99	10
item_004	Item Name 004	0	12.99	10
item_5	Nutella	1	3.0	0



10. Delivery Admin Screen:

When an employee of delivery division logs into the system, they get the Delivery UI that aids them in viewing the open delivery orders as a table. Employee can update the table with the delivery status.

— □ ×

Delivery List

Order ID	Customer Name	Product	Address	Delivered?	Delivery Status
Ord1	gk	Item Name 001	2801 SkingIL0	<input checked="" type="checkbox"/>	not delivered

Save

Cancel

All the orders that are made by the customer gets saved in the Order table with the status being not delivered once the delivery admins deliver the goods, this UI helps them in setting it to delivered. The schema for this table is shown below.

MySQL Workbench

Local instance MySQL57 x

File Edit View Query Database Server Tools Scripting Help

SQL File 1 ticket account account - Table storelist orders employee orders x

1 • SELECT * FROM foodstore.orders;

Limit to 500 rows

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	idOrders	idCustomer	idproduct	shippingAddress	deliveryStatus	delivery
▶	Ord1	gk	Item Name 001	2801 Sking IL 0	0	not delivered
*	NULL	NULL	NULL	NULL	NULL	NULL

SCHEMAS

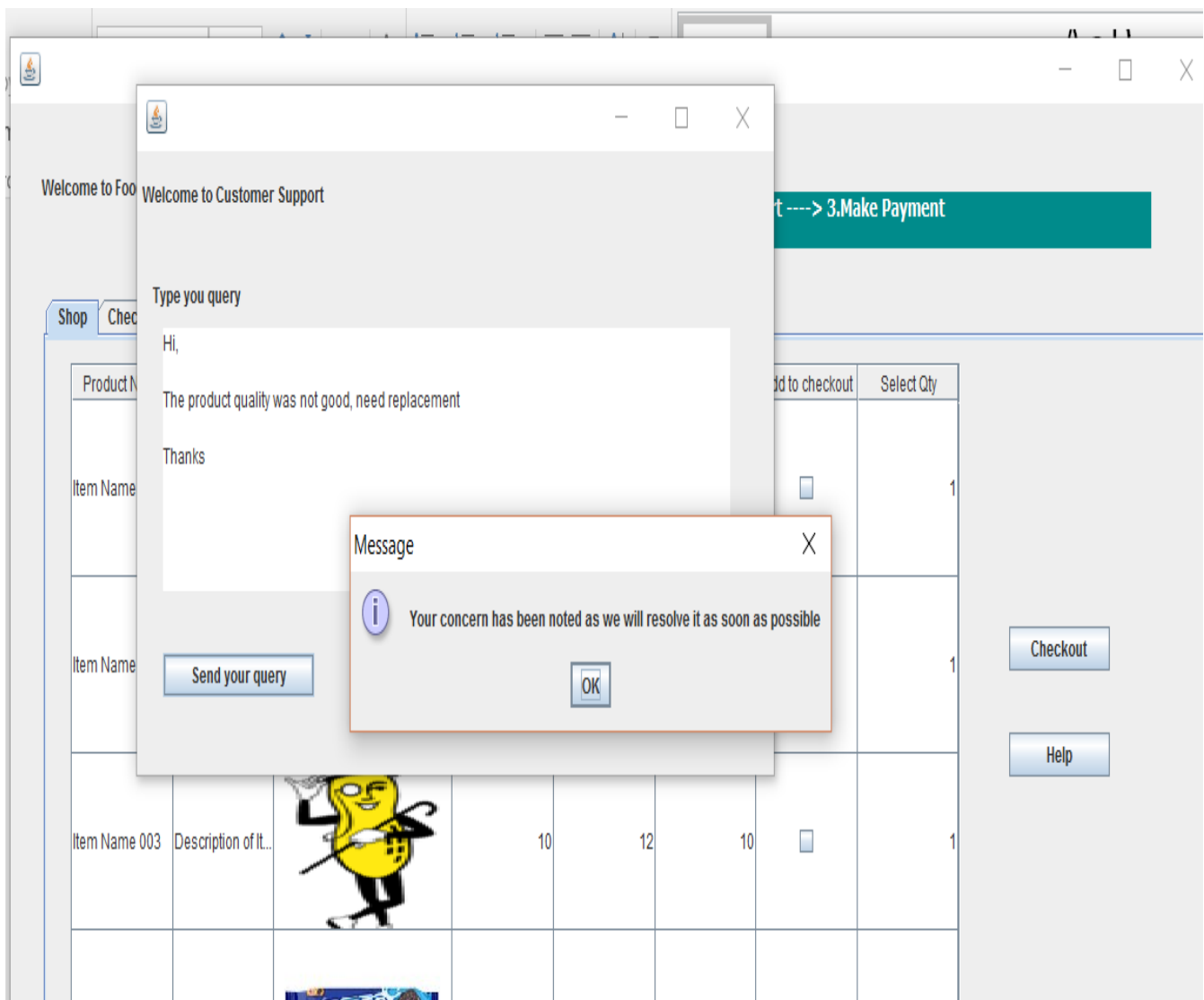
Filter objects

- foodstore
 - Tables
 - account
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - employee
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - orders

11. Support System:

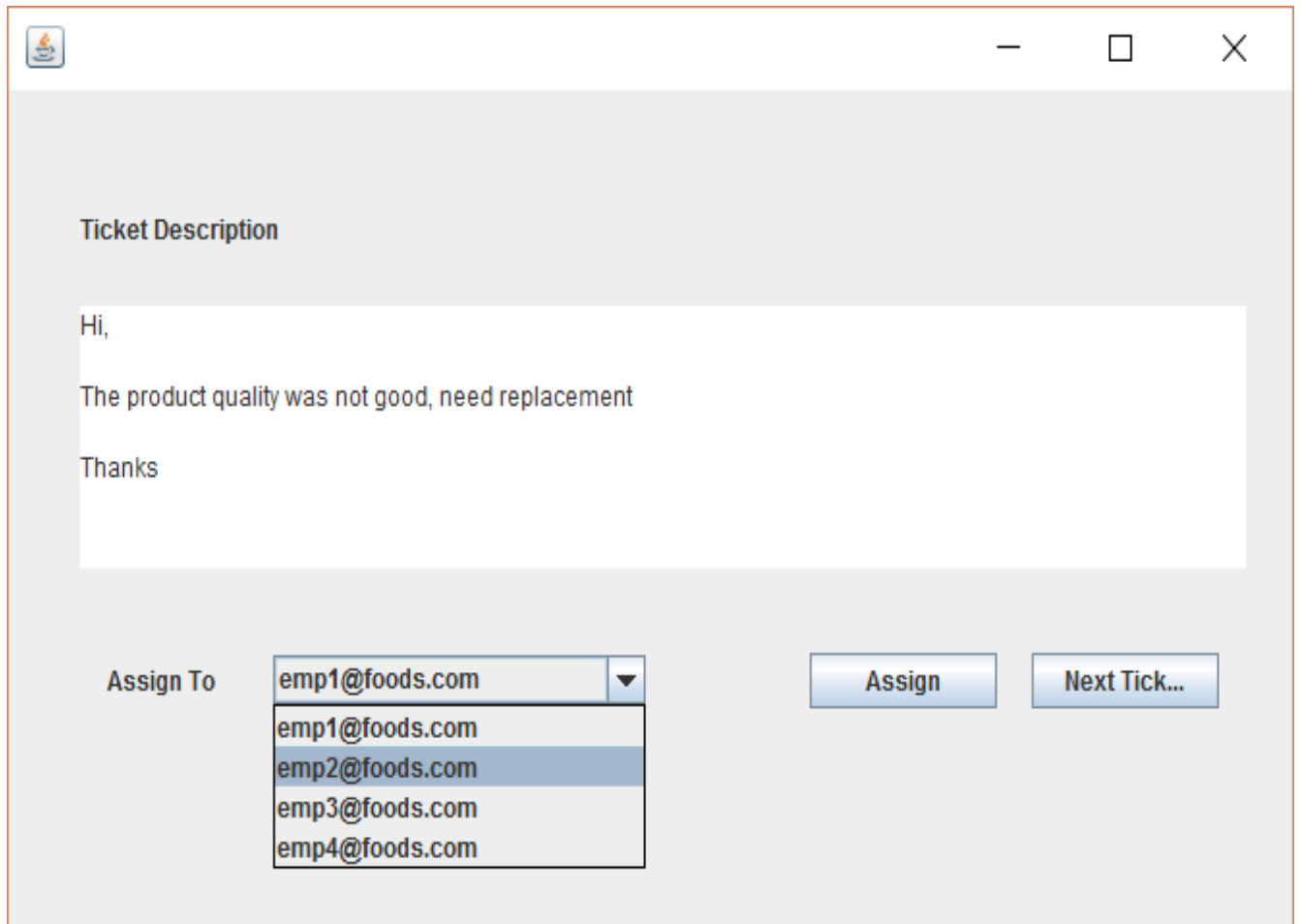
a. Customer Query Screen

Any customer can click on the “Help” button on the shop pane to open the support pane. The support UI lets the customer to enter the customer queries. This query gets updated in the database so that the technicians can work on it.



b. Manager UI to assign technician:

When a manager logs in to the system, they can view the manage portal that aids the managers to assign support tickets to the corresponding employees.



The screenshot shows a web application window with a title bar containing a logo and standard window controls (minimize, maximize, close). The main content area has a light gray background. At the top, the text "Ticket Description" is displayed. Below it, a white text box contains the message: "Hi, The product quality was not good, need replacement Thanks". At the bottom left, there is a label "Assign To" next to a dropdown menu. The dropdown menu is open, showing a list of email addresses: "emp1@foods.com", "emp2@foods.com", "emp3@foods.com", and "emp4@foods.com". The second option, "emp2@foods.com", is highlighted with a blue background. To the right of the dropdown menu are two buttons: "Assign" and "Next Tick...".

Ticket Description

Hi,
The product quality was not good, need replacement
Thanks

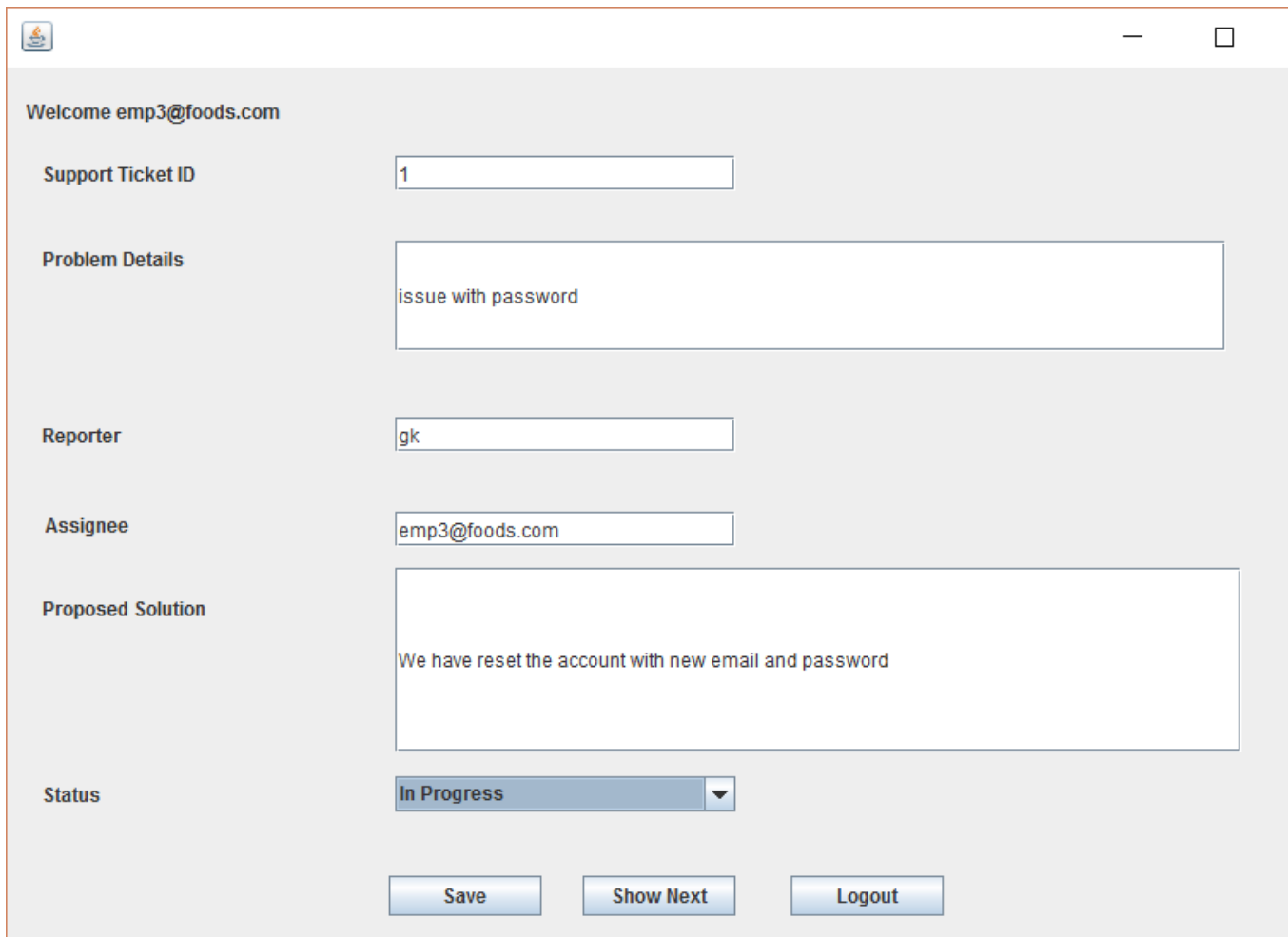
Assign To

- emp1@foods.com
- emp2@foods.com
- emp3@foods.com
- emp4@foods.com

Assign Next Tick...

c. Solution by the technician

When an employee with Support privilege logs into the system, they view the support portal that enables them to provide solution to the user queries and update the status of the tickets assigned to them. They can iterate to the next ticket using the next button.



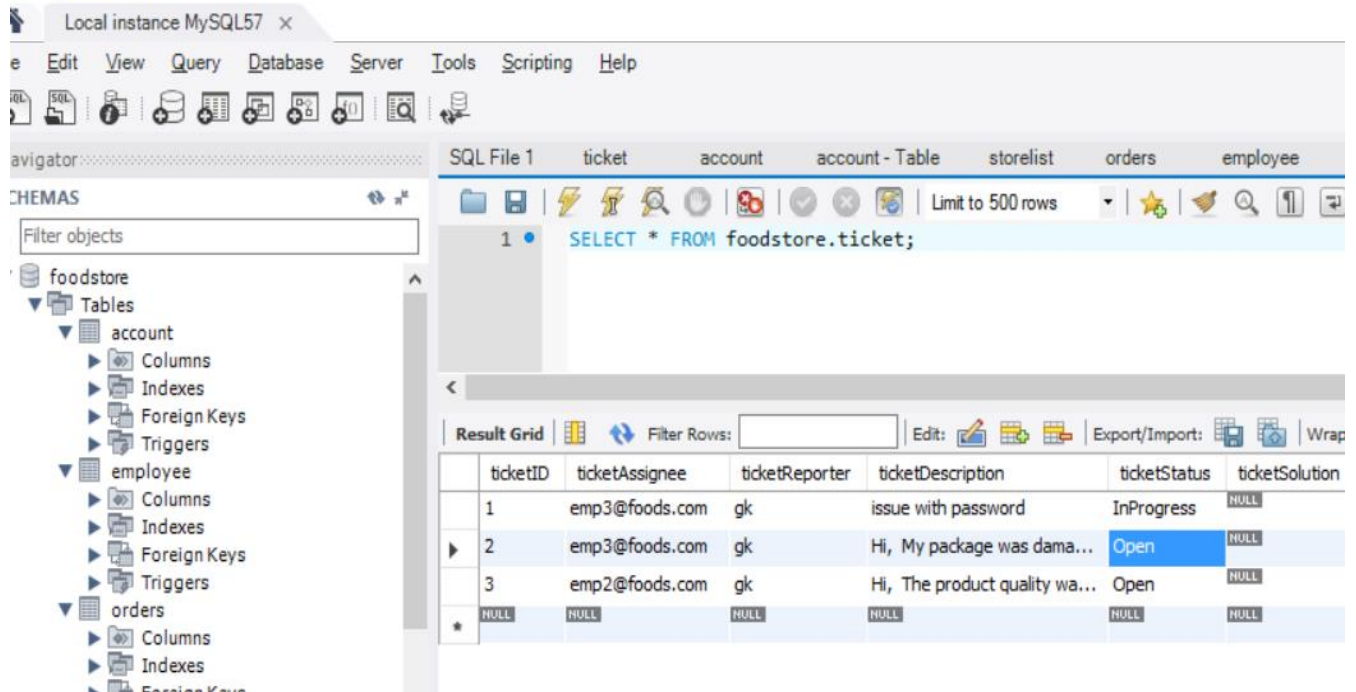
The screenshot shows a web application window with a title bar containing a logo, a minus sign, and a close button. The main content area has a light gray background and displays the following information:

- Welcome emp3@foods.com**
- Support Ticket ID**: 1
- Problem Details**: issue with password
- Reporter**: gk
- Assignee**: emp3@foods.com
- Proposed Solution**: We have reset the account with new email and password
- Status**: In Progress (with a dropdown arrow)

At the bottom of the form, there are three buttons: **Save**, **Show Next**, and **Logout**.

The data from the above UIs are consolidated under a single table called tickets which contains the details like reporter, assignee, status of the ticket, proposed solution. The details of assigned tickets are shown in the SQL workbench for cross verification.

MySQL Workbench



The screenshot displays the MySQL Workbench interface. On the left, the 'Schemas' pane shows the 'foodstore' database with tables 'account', 'employee', and 'orders'. The 'ticket' table is selected in the top tab bar. The SQL editor contains the query: `SELECT * FROM foodstore.ticket;`. The 'Result Grid' shows the following data:

ticketID	ticketAssignee	ticketReporter	ticketDescription	ticketStatus	ticketSolution
1	emp3@foods.com	gk	issue with password	InProgress	NULL
2	emp3@foods.com	gk	Hi, My package was dama...	Open	NULL
3	emp2@foods.com	gk	Hi, The product quality wa...	Open	NULL
*	NULL	NULL	NULL	NULL	NULL