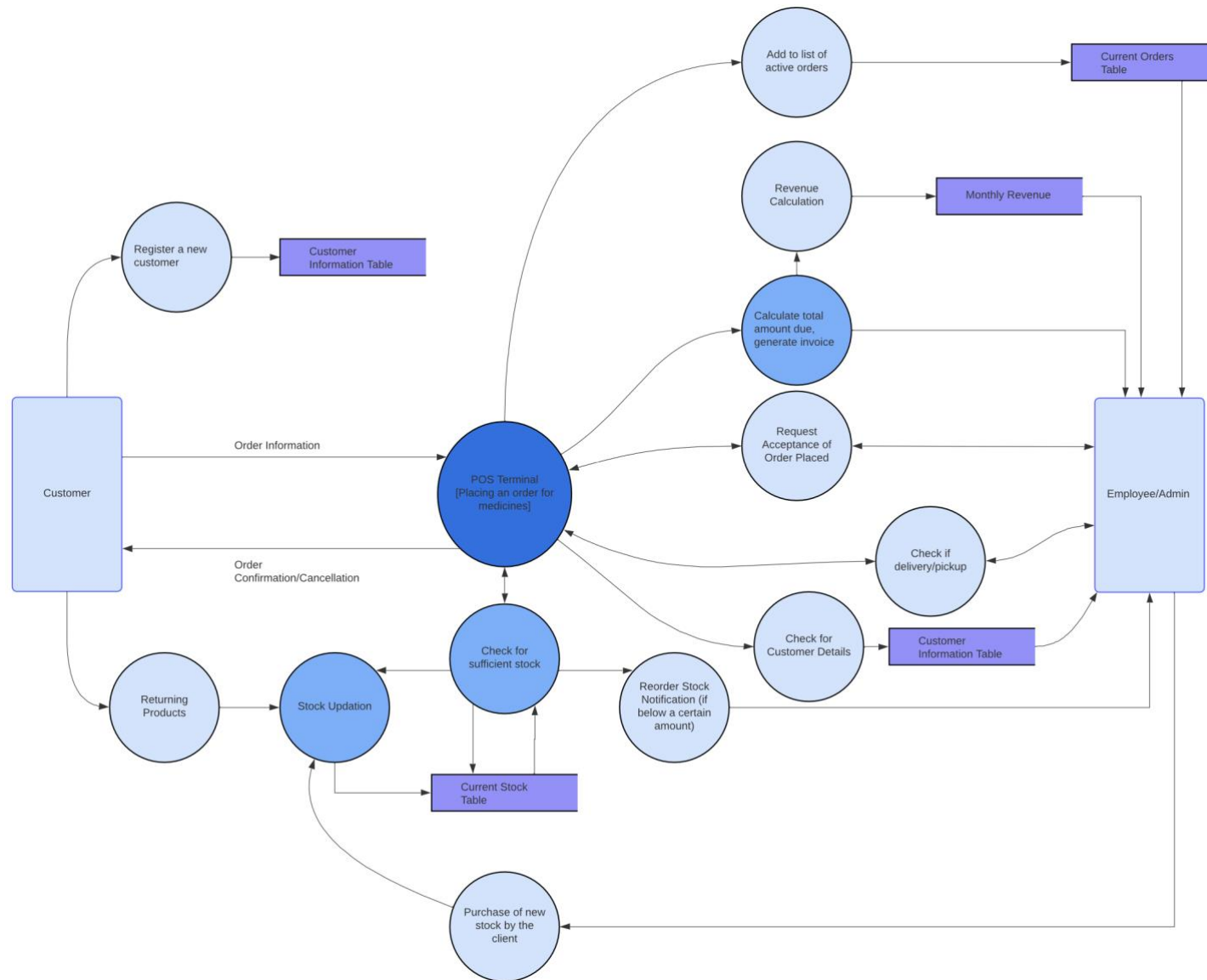


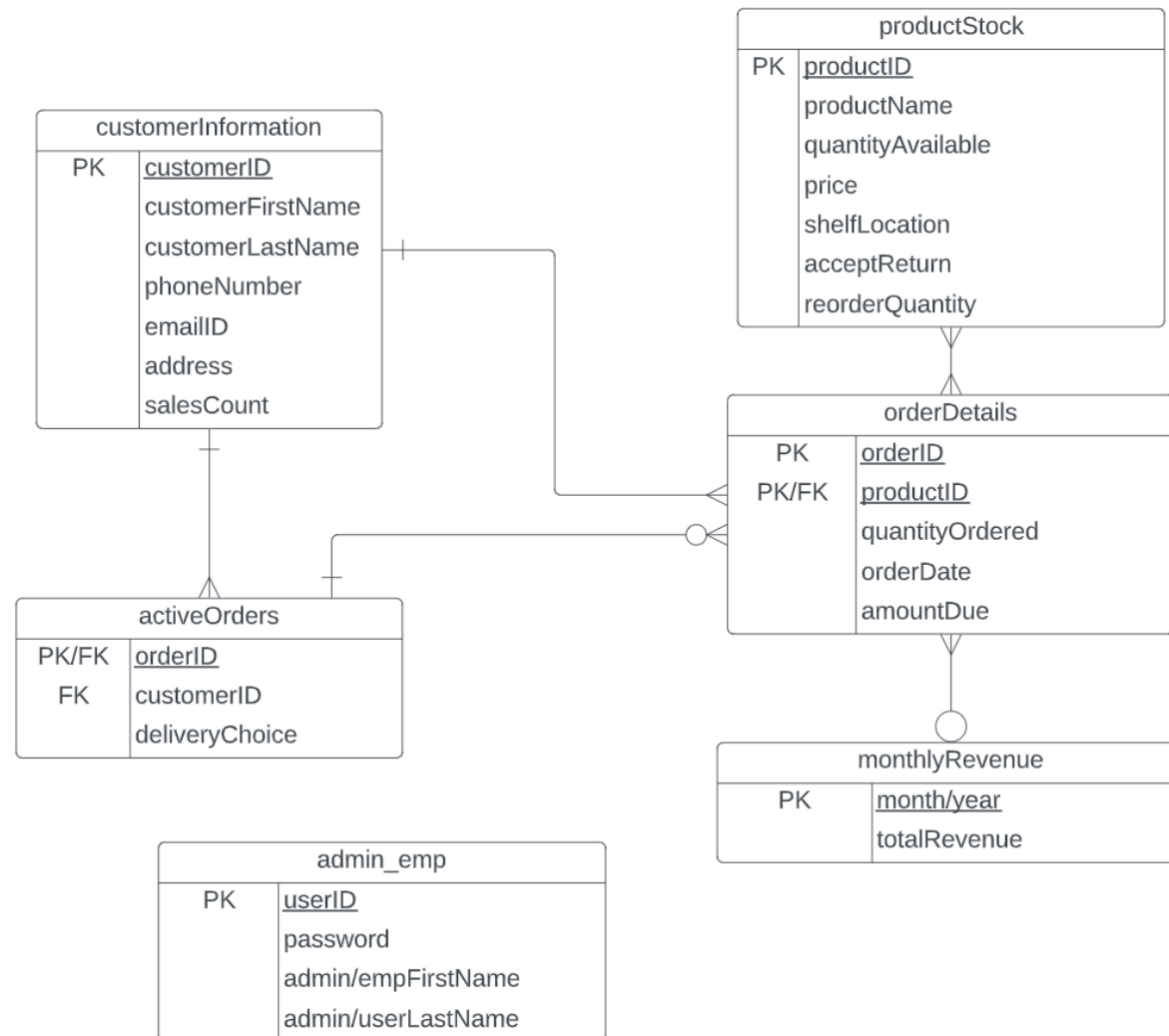
Criterion B: Design

Overall Structure of the product:

Data Flow Diagram



Entity Relationship Diagram



Internal Structure:

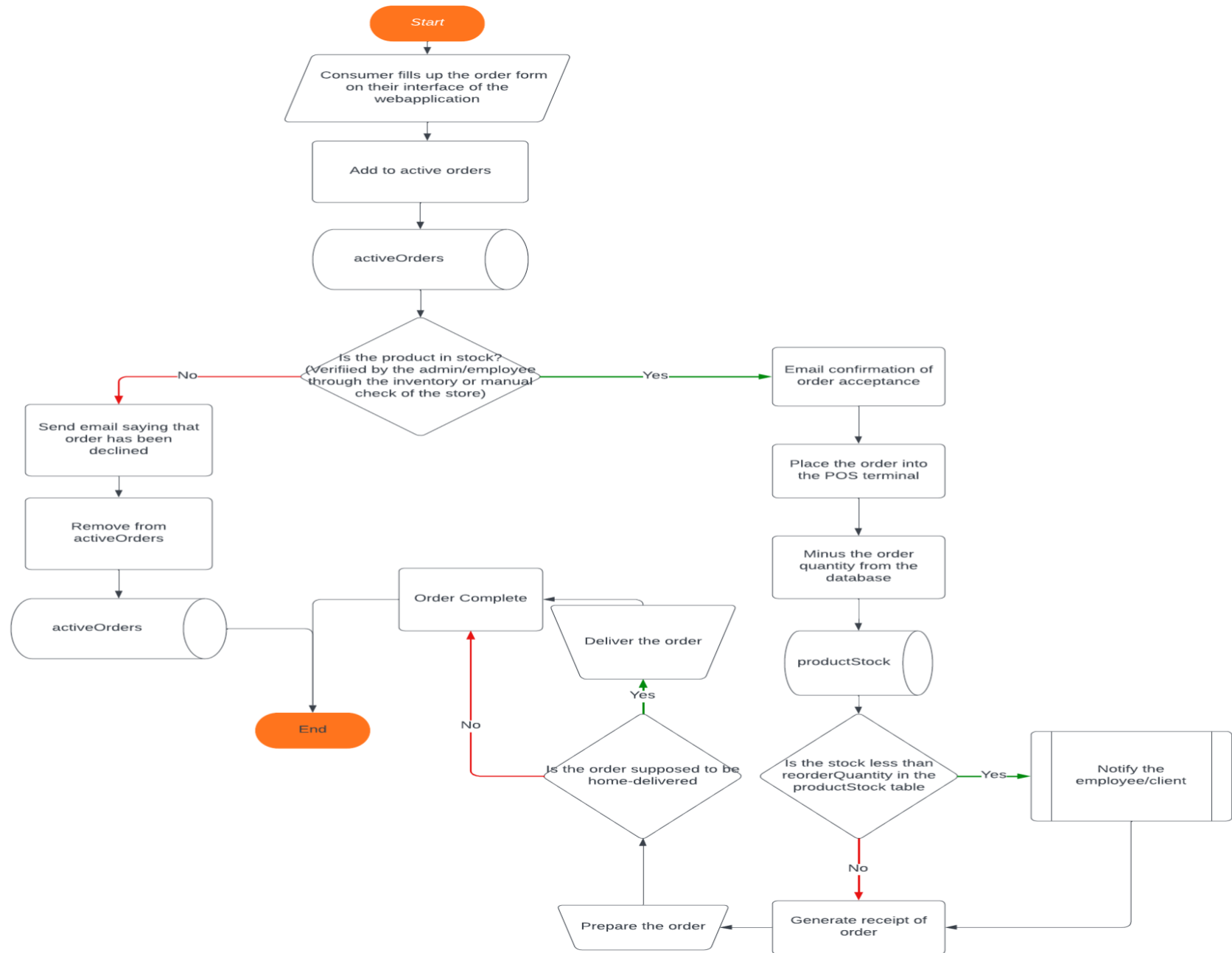
Algorithm for placing an order

1. Consumer places their order on their interface of the web application and attach their prescription.
2. Admin/Employee receives the request for an order. They check if the items available are in stock and accept or decline the orders based on this.
3. Assuming the order is accepted, the consumer will receive a notification via their email. *
4. The order is prepared at the store for delivery/pickup.
5. Admin/Employee enters the items into the POS to generate the receipt of the order.
6. If the order is placed for home-delivery, it will be sent to the consumer's address via the delivery-boy, else step 7. The delivery boy will collect the amount specified in the receipt. Go to step 8.
7. If the order is placed for pickup, it will be kept at the store. Once the consumer arrives to pick up their order from the pharmacy, they will pay their and then.
8. This process is repeated endlessly until the orders stop coming in or are declined.

* In case the order is declined, the following steps are followed:

3. The consumer receives a notification via email that their order has been declined. They may then choose to contact the store to enquire further about this.

Flowchart for placing an order



Proposed Tables

Table Name: customerInformation

The purpose of this table is to store the information of each customer who has bought from the pharmacy.

Field Name	Datatype	Description	Other details
customerID	integer	Unique ID assigned to each customer when they register on the web application's customer interface.	Primary key. Uses the auto-increment feature on MySQL
customerFirstName	varchar	Stores the first name of the customer	Up to 50 characters
customerLastName	varchar	Stores the last name of the customer	Up to 50 characters
phoneNumber	integer	Stores the phone number of the customer	Requires 10 digits to be entered and values greater than 0
emailID	varchar	Stores the email ID of the customer	Up to 100 characters
address	varchar	Stores the home address of the customer for delivery purposes	Up to 150 characters
salesCount	integer	Stores the number of times the customer has bought from the shop	Uses auto-increment feature on MySQL

Table Name: orderDetails This table stores the details of every order made in the pharmacy. It gives a view of every product ordered under a particular order ID.			
Field Name	Data type	Description	Order details
orderID	integer	Stores the order ID for the order which has been placed	Primary key*. Uses the auto-increment feature on MySQL.
productID	integer	Stores the ID of the product which has been ordered.	Primary key* and Foreign Key. All product IDs must be present in the productStock table first as that is the parent table for this field.
quantityOrdered	integer	Stores the quantity of the product which has been ordered by the customer	Has to be greater than 0
orderDate	date/time	Stores the date of the order	Must be current date or past date
amountDue	currency/money	Stores the amount which the customer needs to pay the pharmacy for a particular quantity of a certain product	Has to be greater than 0

* = Composite Primary Key

Table Name: productStock			
This table stores all information relating to the stocks of products available in the pharmacy.			
Field Name	Datatype	Description	Other details
productID	integer	Unique ID assigned to each product when a new product is added to the table	Primary key. Uses the auto-increment feature on MySQL
productName	varchar	Stores the name of the product in the pharmacy	Up to 50 characters
quantityAvailable	integer	Stores the stock available in the pharmacy of each certain product	Has to be greater than 0
price	currency/money	Stores the price for every product sold in the table	Has to be greater than 0
shelfLocation	integer	Stores the shelf number for each product so it is easier for the admin/employee to locate it in the pharmacy	Has to be greater than 0
acceptReturn	boolean	Stores values which indicate to an employee whether a certain product can be returned or not	(Yes) denotes that the product can be returned whereas (No) denotes it cannot be returned
reorderQuantity	integer	Stores the value of the reorder quantity of each product at which the employee/admin needs to be notified about reordering a product	Has to be greater than 0

Table Name: monthlyRevenue

This table stores the calculated values for monthly revenues. It helps the owner keep track of how his business is doing.

Field Name	Datatype	Description	Other details
month/year	date/time	Stores the month and year for which the revenue is calculated	Primary key
totalRevenue	currency/money	Stores the total revenue for each month	The sum of the values of all orders in a month is calculated and stored in this field

Table Name: activeOrders

This table shows the employee/admin all the live orders which need to be prepared and delivered/picked up.

Field Name	Datatype	Description	Other details
orderID	integer	Stores the order ID for the order which has been placed	Primary key and Foreign key. This field has a relationship with the orderID field in the orderDetails database
customerID	integer	Unique ID assigned to each customer when they register on the web application's customer interface.	Foreign key sharing a relationship with the field customerID in the customerInformation table.
deliveryChoice	boolean	Stores the home-delivery or pickup choice for each active order	(Yes) denotes home-delivery whereas (No) denotes pickup

Table Name: admin_emp

This table stores all the login ids and passwords for the admin/employee. It also stores their first and last names so it can be seen which user ID belongs to whom.

Field Name	Datatype	Description	Other details
userID	integer	Creates and stores a unique ID for each admin/employee when added to the database. This is used to log them into the web application	Primary key. Uses the auto-increment feature on MySQL
password	varchar	Stores the password of the admin/employee to login to the web application	Requires minimum 8 characters
admin/empFirstName	varchar	Stores the first name of the admin/employee	Up to 50 characters
admin/empLastName	varchar	Stores the last name of the admin/employee	Up to 50 characters

Proposed Forms

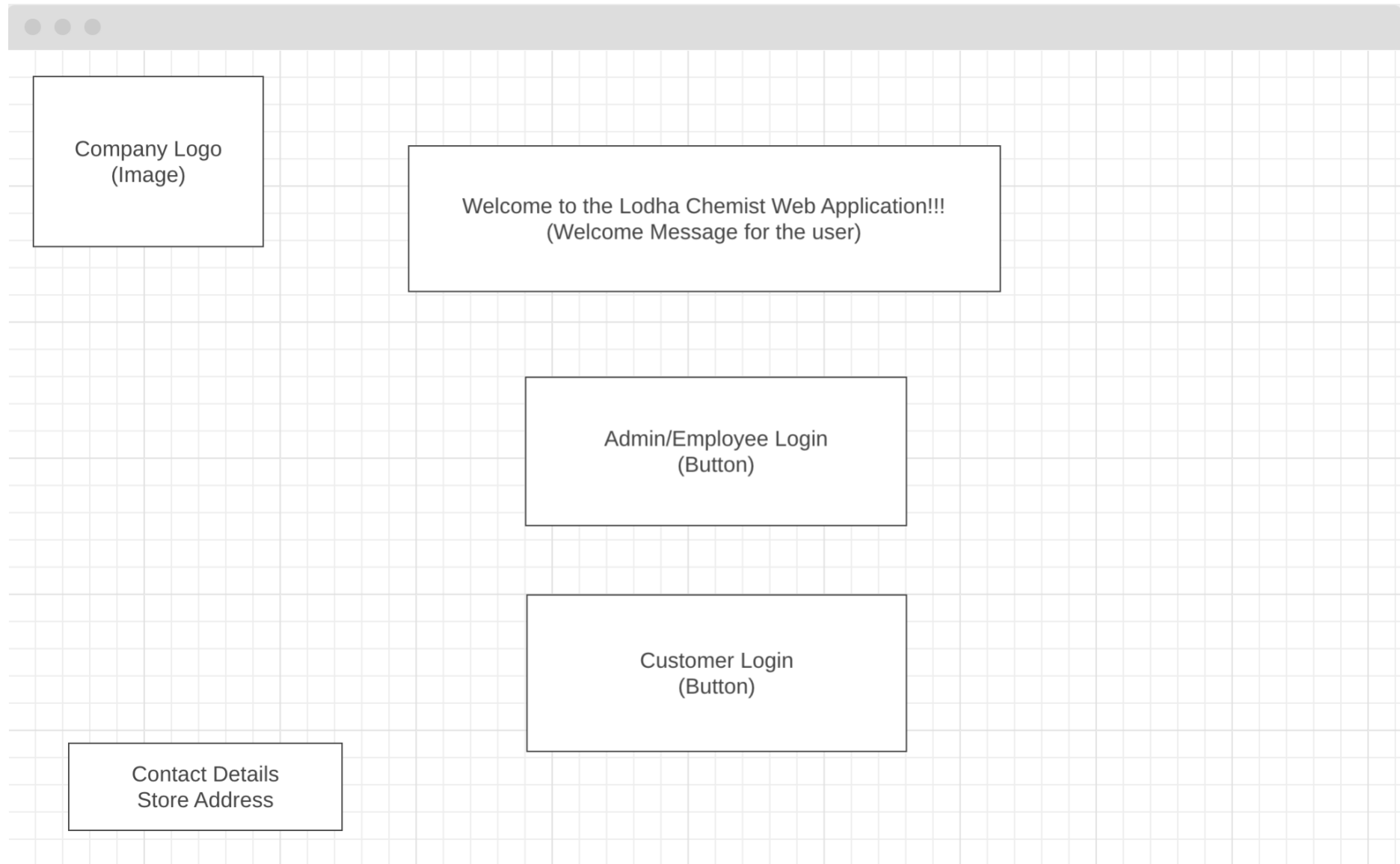
Form Name	Outline
Home Page	This form allows the user logging onto the website to choose the way he would like to login – as a user or as a admin/employee
Admin/Employee Login Page	This form allows the admin/employee to enter their userID and password to login to the admin/employee section of the web application. They can access this page after clicking the admin/employee button on the home page
Customer Login Page	This form allows the customer to login by entering their phone number and password after clicking the customer login button on the Home page
newCustomerRegistration	This form allows new customers to register and use the web application. It can be accessed via a button on the customer login page
customerCart	The customer is redirected to this form after logging into their account. It allows them to place the orders for their product and add a prescription as well
changeCustomerDetails	This form allows the customer to change his details such as phone number, password, address, etc and update them in the database.
addNewProduct	This form allows the employee/admin to add a new product to the inventory
updateProduct	This form allows the employee/admin to update the information of a certain product in the inventory
deleteProduct	This form allows the employee/admin to delete a certain product from the inventory

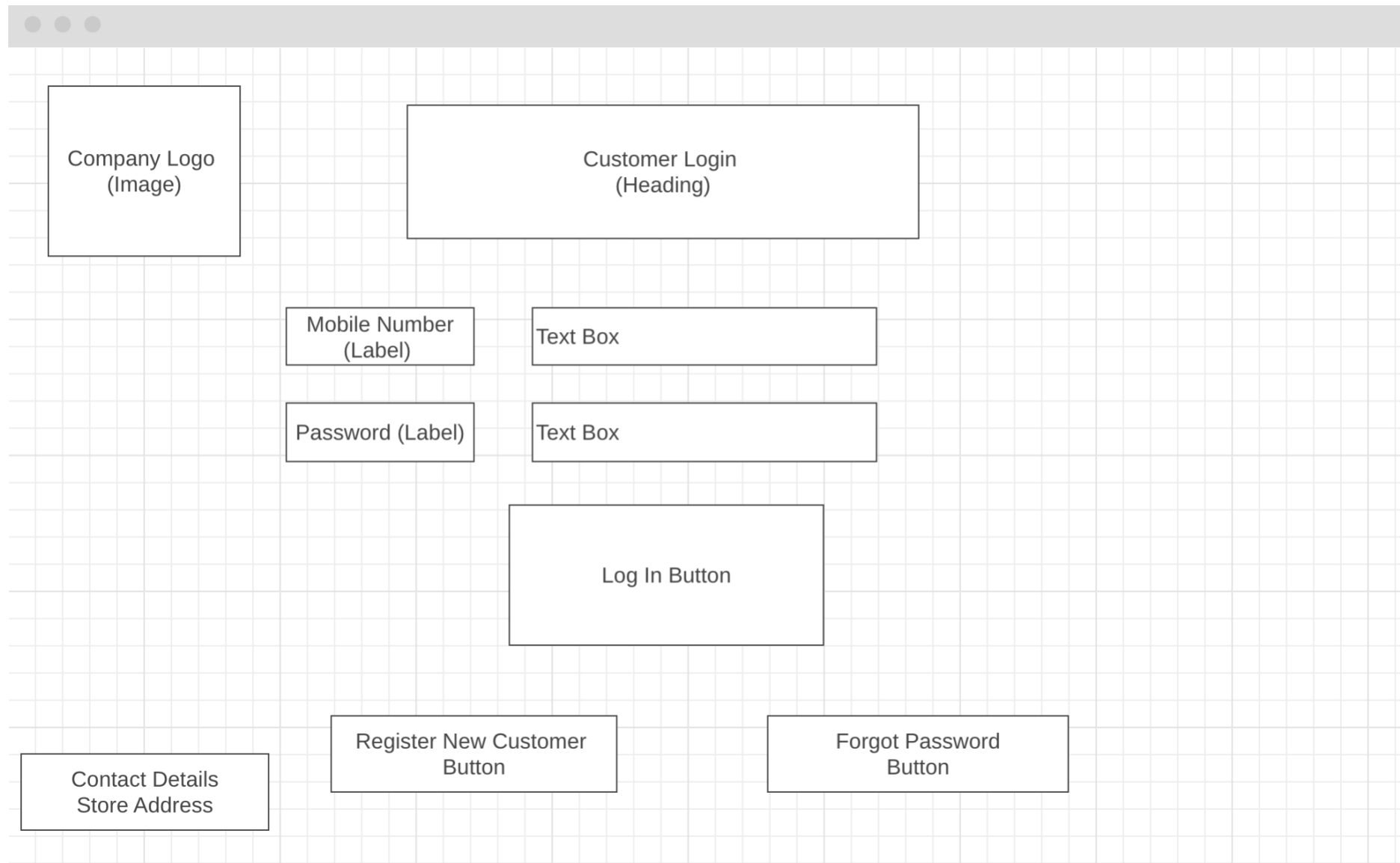
Proposed Queries

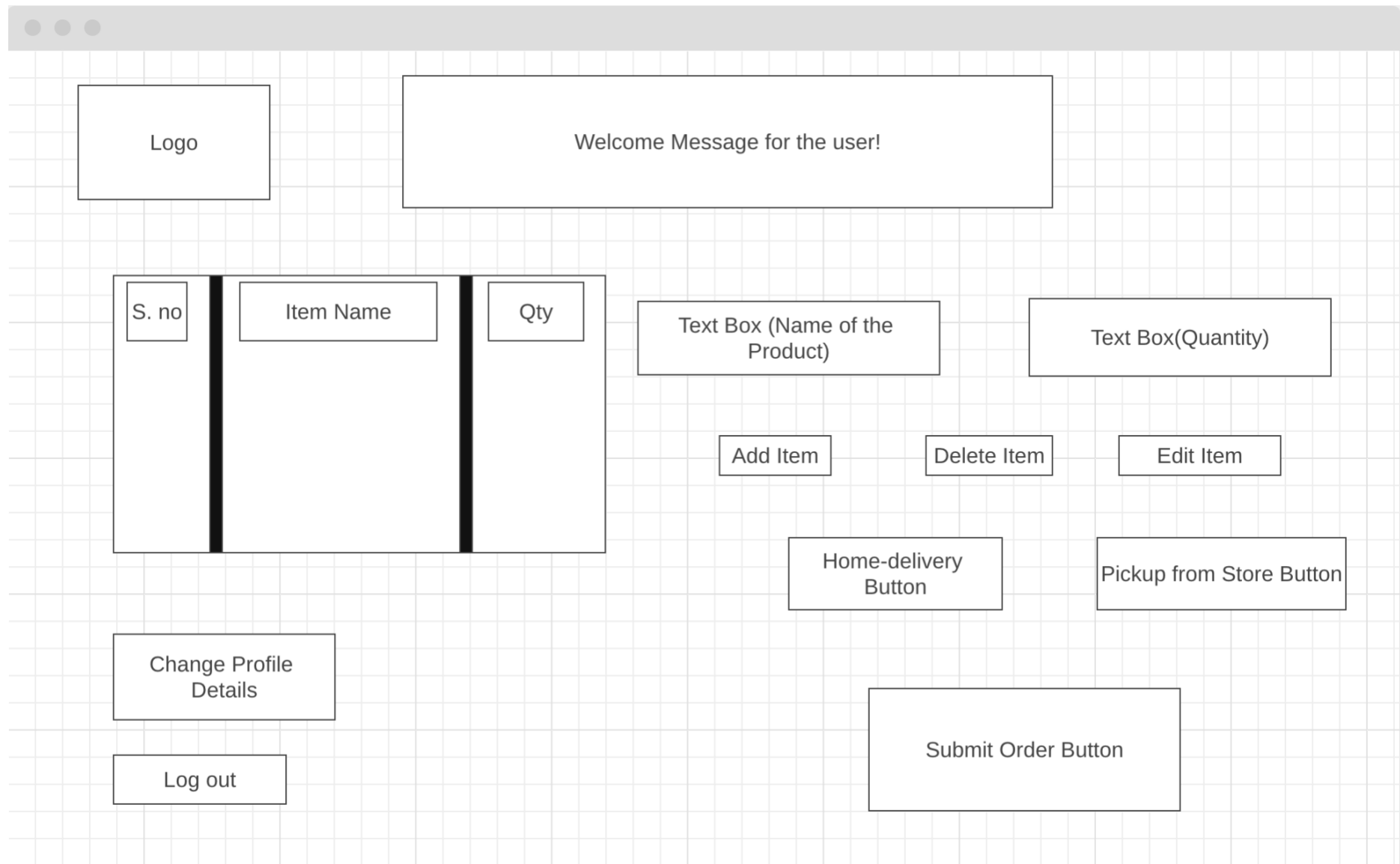
Query Name	Outline
newOrderTransaction	Used to insert the details of a new order into the POS feature to obtain a receipt for the order
newUser	Used to insert the details of a new employee into the database and to give them login access
reorderItems	Used to filter the list of items from the database which require reordering

Proposed Reports

Report Name	Outline
orderConfirmation	This report is sent to the customer via email once their order has been confirmed by the employee/admin at the pharmacy
orderDetailsView	This report allows the employee/admin to view the details of an active order
Point of Sale Terminal Receipt	This is a computer-generated receipt for the order made by a customer on the web application or at the POS.
monthlyRevenueReport	This report shows my client his total revenue for the past months so that he is able to keep track of how the pharmacy is running
inventoryReport	This report shows the employee/admin the current stocks when requested for through the click of a button
orderNewStockReport	This report is generated via notification on the web application warning the admin/employee if a certain product in the pharmacy is about to run out and needs to be restocked







Test Plan

Item No.	Test Item	Test method	Test data	Part of system tested	Expected outcome	Actual output
1	Buttons on the interface	Clicking on all of the buttons	Normal	GUI	They should link to respective pages, show alerts or carry out their specific function.	
2	Login page	Input the login credentials	Normal, Erroneous	GUI, Database	Only users with valid user names and passwords should be able to login	
3	Registering a new customer	Registering a new user	Normal, Boundary, Extreme, Erroneous	GUI, Database	A new customer should be registered into the database upon filling and submitting newCustomerRegistration form	
4	Placing an order	Click on submit order button after filling out the customerCart form	Normal, Erroneous	GUI, Database	Employee/Admin should receive the order request	

5	Notification: accept/decline an order	Place order from consumer interface and simultaneously check admin/employee interface	Normal	GUI, Database	Customer should receive notification on email whether his order has been accepted or declined	
6	Viewing the active orders	Click on the view active orders button	Normal	GUI, Database	Should produce a table view of the active orders	
7	Adding new products	Click on add product button and fill form	Normal, Erroneous	GUI, Database	Add a new product into the database with all its details	
8	Updating information on existing products	Click on edit product information button and change information in form	Normal, Erroneous	GUI, Database	Edit the information in the database and the displayed values on the GUI should change	
9	Deleting existing products	Click on the delete button and select productID	Normal	GUI, Database	Record should be deleted from the database however; previous revenue records mustn't be affected	
10	View entire inventory	Click on the view inventory	Normal	GUI, Database	Table view of inventory is displayed	

		button in the nav pane				
11	Notification: that stock of a product is below reorder quantity	Decrease quantity of a product below reorder level	Normal	GUI, Database	Window alert stating that the product needs to be reordered	
12	POS Terminal	Fill out the POS terminal form for an order	Normal, Erroneous	GUI, Database	Receipt should be generated and the order should be logged in the database	
13	Returning a product	Click on the return product button in the nav pane and fill the form	Normal, Erroneous	GUI, Database	Should return the product and change the inventory. Should change the total revenue and order records	
14	Filtering products which need to be reordered	Click on the reorder products button	Normal	GUI, Database	Filter data should appear in table format	
15	Creating a new admin/employee user	Click on the new user registration button and fill out the form	Normal, Erroneous	GUI, Database	New user should be created	