

Main Links

GitHub

<https://github.com/meghna-dash/store-management-system>

Testing Videos

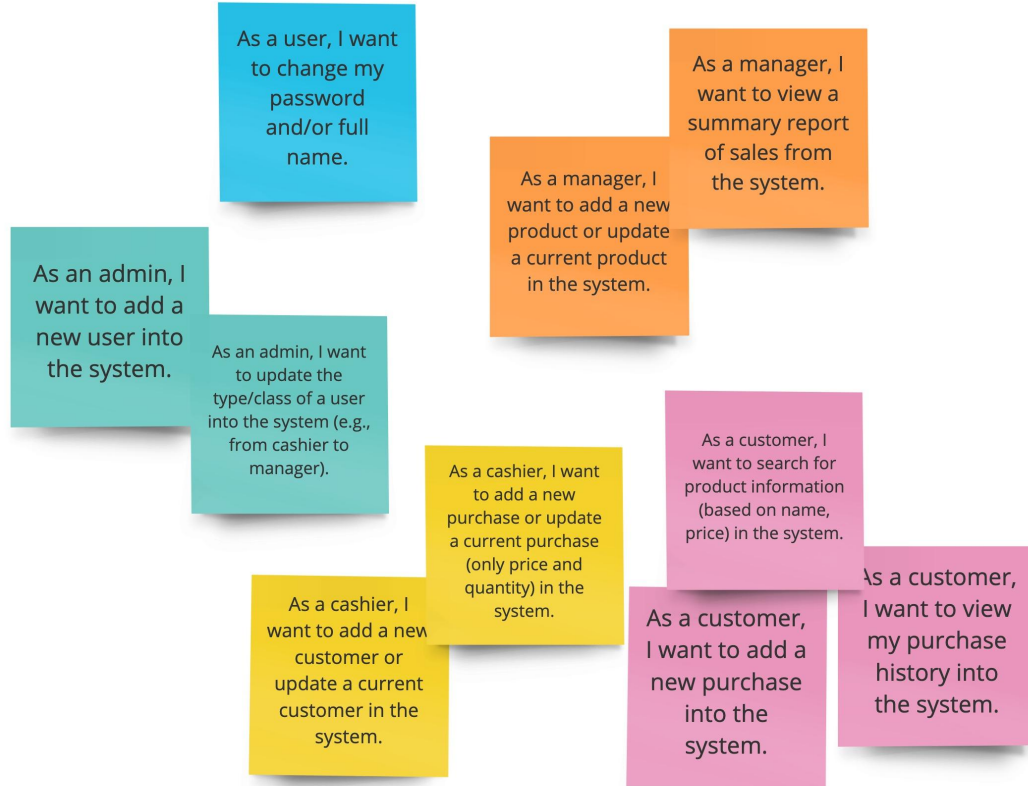
<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

<https://www.loom.com/share/6590eae61c7644859430876143037376>

Project 3

Store Management System

User Stories: Every persona has different needs.



Use Cases (User)

Task 1: Rewrite two common use cases for each user story. Sketch the screens the system should display in each use case.

Action	Regular Cases	Irregular Case
Change Name or Password	All required fields are filled correctly and of correct type.	Values are of the incorrect type or required field (id) is not filled.

Use Cases (Admin)

Task 1: Rewrite two common use cases for each user story. Sketch the screens the system should display in each use case.

Action	Regular Cases	Irregular Case
Add a New User	All required fields are filled correctly and of correct type.	Values are of the incorrect type or required field (id) is not filled.
Update the Type of a User	All required fields are filled correctly and of correct type.	Values are of the incorrect type or required field (id) is not filled.

Use Cases (Manager)

Task 1: Rewrite two common use cases for each user story. Sketch the screens the system should display in each use case.

Action	Regular Cases	Irregular Case
Add/Update a Product	All required fields are filled correctly and of correct type	Values are of the incorrect type or required field (id) is not filled.
View a Summary Report	The summary report is generated with no errors.	There is an error retrieving or showing the summary info.

Task 1: Rewrite two common use cases for each user story. Sketch the screens the system should display in each use case.

Use Cases (Cashier)

Action	Regular Cases	Irregular Case
Add/Update a Customer	All required fields are filled correctly and of correct type	Values are of the incorrect type or required field (id) is not filled.
Add/Update a Purchase	All required fields are filled correctly and of correct type	Values are of the incorrect type or required field (id) is not filled.

Use Cases (Customer)

Task 1: Rewrite two common use cases for each user story. Sketch the screens the system should display in each use case.

Action	Regular Cases	Irregular Case
Add/Update a Purchase	All required fields are filled correctly and of correct type	Values are of the incorrect type or required field (id) is not filled.
View Purchase History	The purchase history report is generated with no errors.	There is an error retrieving or showing the purchase info.
Search for a Product	All required fields are filled correctly and items are shown.	Values are of the incorrect type or required field (id) is not filled.

Design: User Views

Different types of users have different home pages.

Admin Home Page

Set System
Configuration

Add a
New User

Delete an
Existing User

Manager Home Page

View Sales
Summary
Report

Add a
New Product

Update an
Existing
Product

Cashier Home Page

Add or
Update a
Customer

Add or
Update a
Purchase

Report an
Issue to
Manager

Customer Home Page

Make a New
Purchase

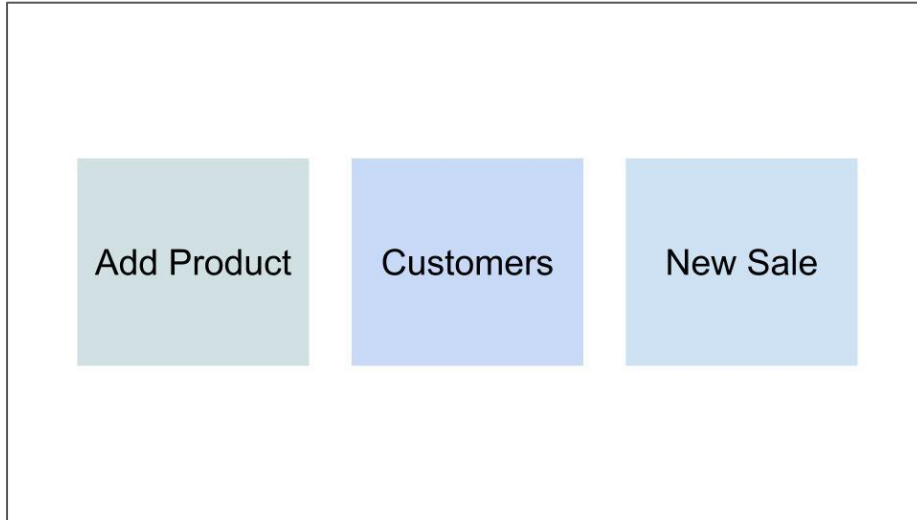
View
Purchases

Search for
Products

UI: Add/Update Product

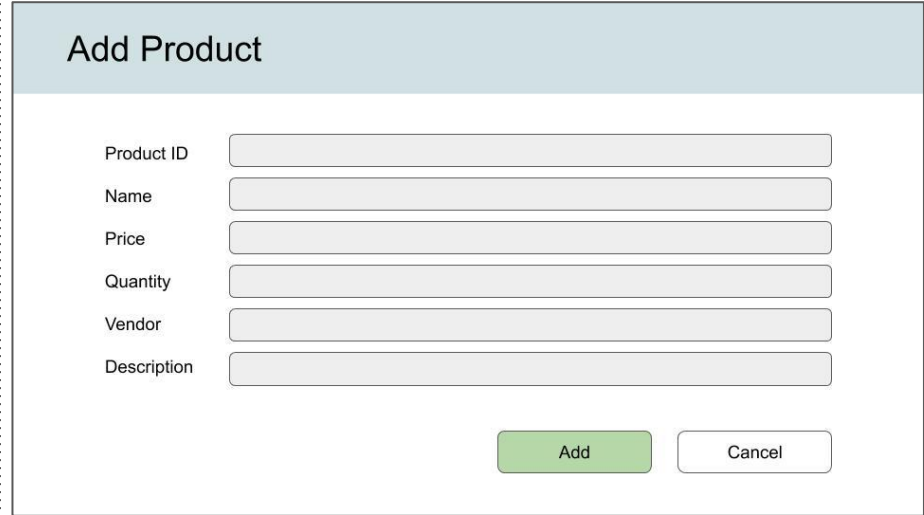
Design UI and write use case for "add or update a product in the system".

1. **Actor** selects Add Product



A horizontal menu bar containing three rectangular buttons. The first button on the left is teal and labeled "Add Product". The middle button is blue and labeled "Customers". The third button on the right is light blue and labeled "New Sale".

2. **System** displays Add Product Screen



The "Add Product" screen features a light blue header with the title "Add Product". Below the header, there are six input fields, each preceded by a label: "Product ID", "Name", "Price", "Quantity", "Vendor", and "Description". At the bottom right of the screen, there are two buttons: a green "Add" button and a white "Cancel" button with a gray border.

3. **Actor** adds information & clicks Add button.

Add Product

Product ID	<input type="text" value="100"/>
Name	<input type="text" value="Cap"/>
Price	<input type="text" value="\$10.99"/>
Quantity	<input type="text" value="30"/>
Vendor	<input type="text" value="Sporting Goods"/>
Description	<input type="text" value="Dark blue cap; for men and women."/>

4. **System** hides Add Product screen and shows confirmation.

New product added successfully!

X

5. **Actor** clicks on Okay button



6. **System** closes pop-up and returns to main screen



Three colored squares are arranged horizontally. The first square on the left is teal and contains the text 'Add Product'. The middle square is blue and contains the text 'Customers'. The third square on the right is light blue and contains the text 'New Sale'.

Add Product

Customers

New Sale

Add Product

Product ID

Name

Price

Quantity

Vendor

Description

Add

Cancel

Add Product

Product ID

100

Name

Cap

Price

\$10.99

Quantity

30

Vendor

Sporting Goods

Description

Dark blue cap; for men and women.

Add

Cancel

New product added
successfully!

Okay

Three colored squares are arranged horizontally. The first square on the left is teal and contains the text 'Add Product'. The middle square is blue and contains the text 'Customers'. The third square on the right is light blue and contains the text 'New Sale'.

Add Product

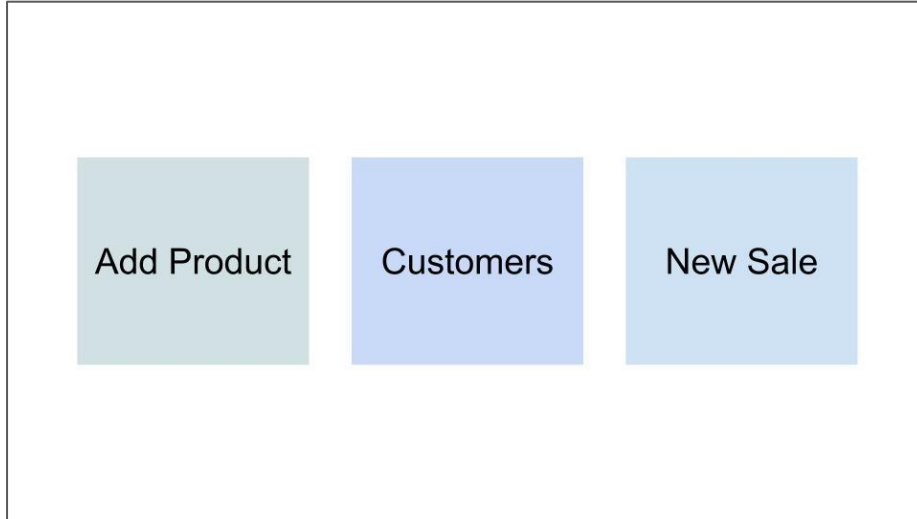
Customers

New Sale

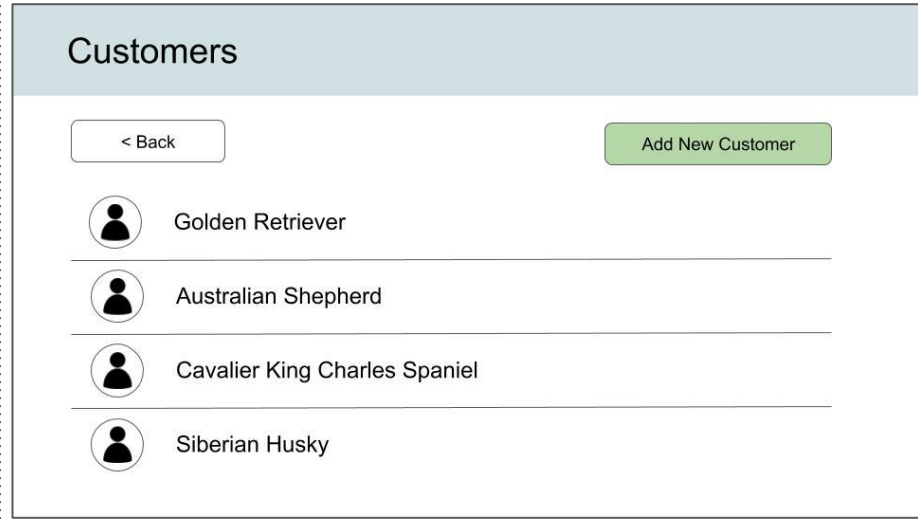
UI: Add/Update a Customer

Design UI and write use case for "add or update a customer in the system".

1. **Actor** selects Customers



2. **System** displays list of all customers





3. **Actor** clicks Add New Customer


Customers


< Back

Add New Customer

Golden Retriever

Australian Shepherd

Cavalier King Charles Spaniel

Siberian Husky

4. **System** shows Add New Customer screen

Add New Customer

Customer ID

Name

Add

Cancel

5. **Actor** adds information & clicks Add button.

Add New Customer

Customer ID

101

Name

Bichon Frise

Add

Cancel

6. **System** hides Add Customer screen and shows confirmation.

New customer added successfully!

Okay

7. **Actor** clicks on Okay button

New customer added
successfully!

Okay

8. **System** closes pop-up and shows updated list of customers

Customers

< Back

Add New Customer



Golden Retriever



Australian Shepherd



Cavalier King Charles Spaniel



Siberian Husky

Three colored squares are arranged horizontally. The first square on the left is teal and contains the text 'Add Product'. The middle square is blue and contains the text 'Customers'. The third square on the right is light blue and contains the text 'New Sale'.

Add Product

Customers

New Sale

Customers

< Back

Add New Customer



Golden Retriever



Australian Shepherd



Cavalier King Charles Spaniel



Siberian Husky

Add New Customer

Customer ID

Name

Add

Cancel

Add New Customer

Customer ID

101

Name

Bichon Frise

Add

Cancel

New customer added
successfully!

Okay

Customers

< Back

Add New Customer



Golden Retriever



Australian Shepherd



Cavalier King Charles Spaniel



Siberian Husky

Three colored squares are arranged horizontally. The first square on the left is teal and contains the text 'Add Product'. The middle square is blue and contains the text 'Customers'. The third square on the right is light blue and contains the text 'New Sale'.

Add Product

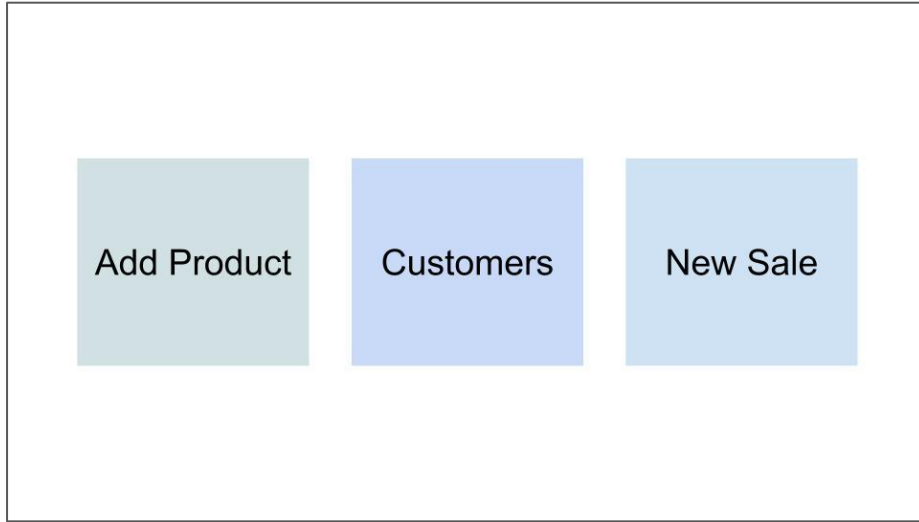
Customers

New Sale

UI: Add/Update a Purchase

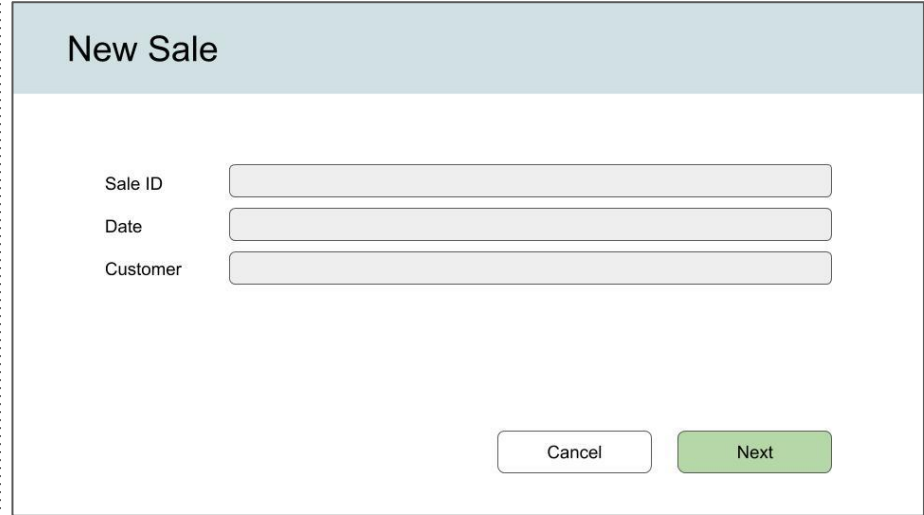
Design UI and write use case for "record or edit a purchase from a customer into the system".

1. **Actor** selects New Sale



A horizontal navigation menu with three rectangular buttons. The first button is light teal and labeled 'Add Product'. The second button is light blue and labeled 'Customers'. The third button is a slightly darker blue and labeled 'New Sale'.

2. **System** displays Add New Sale page



The 'Add New Sale' form has a light blue header bar with the title 'New Sale'. Below the header, there are three input fields stacked vertically, each with a label to its left: 'Sale ID', 'Date', and 'Customer'. At the bottom right of the form, there are two buttons: a white 'Cancel' button and a green 'Next' button.

3. **Actor** adds basic information & clicks Next button.

New Sale

Sale ID

102

Date

09/14/2019

Customer

Bichon Frise

Cancel

Next

4. **System** shows fields to add items.

New Sale

Items

Item ID

Quantity

1

+

 Add Another Item

Cancel


Next

5. **Actor** add items in the sale and clicks add.

New Sale

Items

Item ID	<input type="text" value="112"/>	Quantity	<input type="text" value="1"/>
Item ID	<input type="text" value="143"/>	Quantity	<input type="text" value="2"/>

 Add Another Item

6. **System** shows confirmation and hides the New Sale screens

New sale added successfully!

7. **Actor** clicks okay

New sale added
successfully!

Okay

6. **System** takes the user back to the main screen

Add Product

Customers

New Sale

Three colored squares are arranged horizontally. The first square on the left is teal and contains the text 'Add Product'. The middle square is blue and contains the text 'Customers'. The third square on the right is light blue and contains the text 'New Sale'.

Add Product

Customers

New Sale

New Sale

Sale ID

Date

Customer

Cancel

Next

New Sale

Sale ID

102

Date

09/14/2019

Customer

Bichon Frise

Cancel

Next

New Sale

Items

Item ID

Quantity

1



Add Another Item

Cancel

Next

New Sale

Items

Item ID

112

Quantity

1

Item ID

143

Quantity

2



Add Another Item

Cancel

Add

New sale added
successfully!

Okay

Three colored squares are arranged horizontally. The first square on the left is teal and contains the text 'Add Product'. The middle square is blue and contains the text 'Customers'. The third square on the right is light blue and contains the text 'New Sale'.

Add Product

Customers

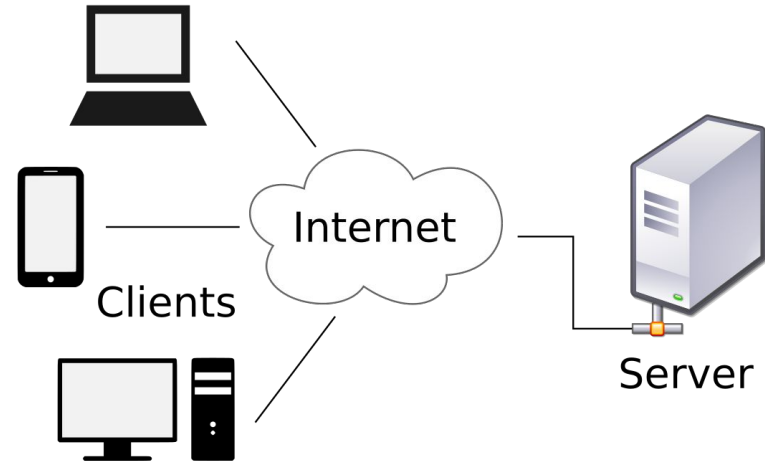
New Sale

Task 2: Redesign the Data Access layer at the client side that can load/save data to a remote server component. Describe the protocol.

Client Side

The client side needs to load and save data to a remote server component.

I am running the GUI locally. The local GUI makes requests to a local SQL server. I am constantly making requests to the database, starting with the login process. Once I am logged in, the user can only view information specific to their account. As the user navigates the application, the client makes requests to the server for the data it needs to show the user.

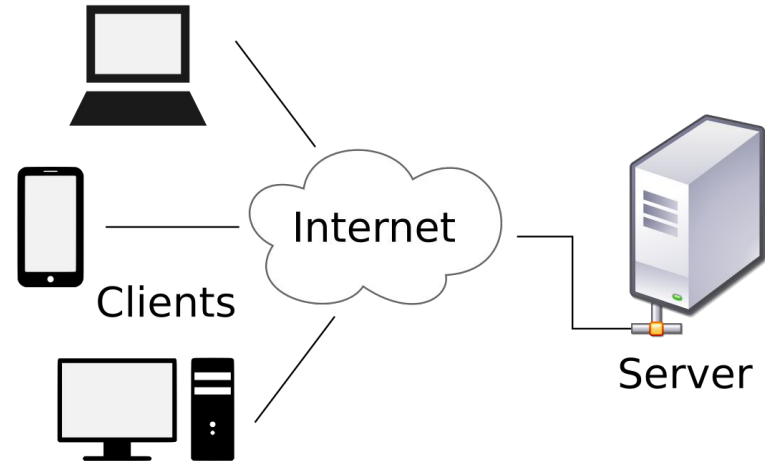


Task 2: Design the server component to perform load/save requests from the Data Access layer at the client side. Describe the protocol.

Server Side

The client side needs to load and save data to a remote server component.

I am running an instance of a SQLite server locally on my machine. The client makes requests to the server, and the server responds in a way that I can define. For example, if I request to view the purchase history of a specific customer by passing a customerId to the server, the server will return results only for that customer.



Task 3: Implement the system.

Implementation

GitHub

<https://github.com/meghna-dash>

Website

<https://meghnadash.design>

Testing

Task 4: Test the system with each use case.

Manger: Add+Update Product

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Customer: Add+Update Purchase

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Cashier: Add+Update Purchase

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Admin: Add a New User

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Testing

User: Change Password and Name

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Admin: Update the Type of User

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Task 4: Test the system with each use case.

Manger: View a Summary Report

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Customer: View Purchase History

<https://www.loom.com/share/4006b2d145a845829c68952602ca28b8>

Thank you!