Store Products Management System

Project Description

We are developing the Store Products Management System, an application that simplifies the management of a store's core operations, Like products management, billing, and user account administration. Our system allows staff to handle tasks like adding, editing, and deleting products and managing user accounts with different roles (Admin and Guest).

Overview

Our System's primary goal is to develop an accessible application that helps store staff manage products and accounts while reducing manual, repetitive tasks. Admins will be able to update the products and manage user accounts, while guests can view products and potentially place orders.

Team Members

We are a team of three:

1. Rishikesh
2. Sai Saketh
3. Jaydeep

Responsibilities for Team Members

Rishikesh - Frontend Development:
> Responsible for designing the user interface, focusing on accessibility and ease of use. He will handle layouts for the login, item management, and user account management screens.

Jayadeep - Backend Development:
> Will work on the logic, implementing core functionalities like adding, deleting, and updating products, as well as managing user roles and authentication processes.

Sai Saketh - Database Design and Management:
> Will design the database structure, creating tables for users, products, and orders. Including organizing data efficiently and handling CRUD operations.

Feature List

User Login and Authentication:
> 1. We will implement a login system that verifies user credentials, assigning roles (Admin or Guest) to control access to different features.
> *Prototype: A basic login screen with role-based access.*

Store Products Management System

User Management:
  1. Admin Only:

      Admins will have access to functions for creating, deleting, and modifying user accounts.

  Prototype Implementation: Admin functionalities for account management.

Products Management:
  1. Add New products:

      Admins can add new products to the store, specifying details like name, price, and description.
  2. Edit Existing products:

      Admins can update details for products, such as pricing or availability.
  3. Delete products:

      Admins can remove Products.

  Prototype Implementation: Basic interface for adding, editing, and deleting products.
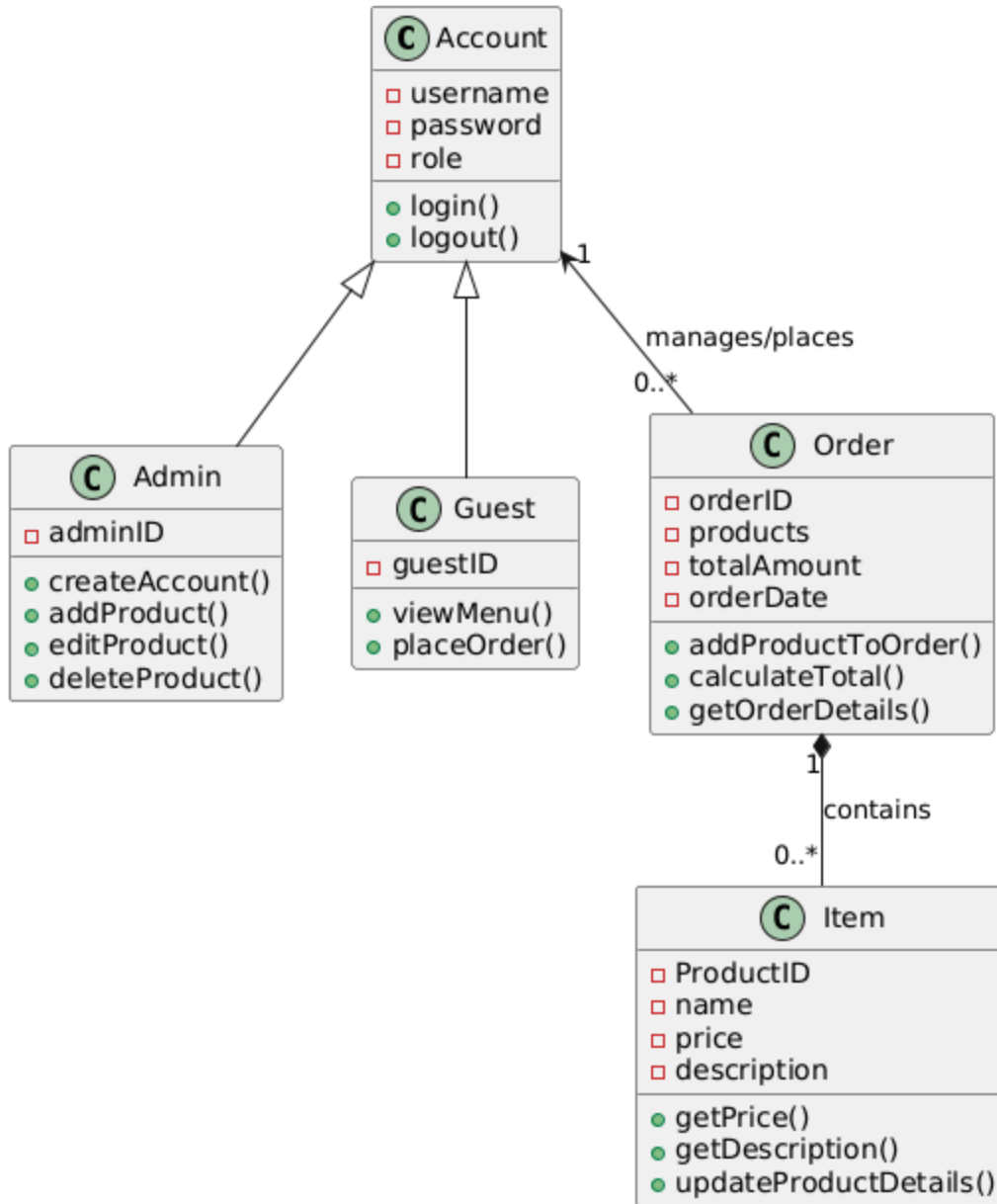
View products:
  1. Guest Access:

      Guests can view all available products, providing an overview of what the store offers.

  Prototype Implementation: A simple item viewing feature for guests.

UML Class Diagram

(Our initial UML, will update as we go)

Store Products Management System

C Account
- □ username
- □ password
- □ role
- ● login()
- ● logout()

C Admin
- □ adminID
- ● createAccount()
- ● addProduct()
- ● editProduct()
- ● deleteProduct()

C Guest
- □ guestID
- ● viewMenu()
- ● placeOrder()

C Order
- □ orderID
- □ products
- □ totalAmount
- □ orderDate
- ● addProductToOrder()
- ● calculateTotal()
- ● getOrderDetails()

manages/places
0..*
1

C Item
- □ ProductID
- □ name
- □ price
- □ description
- ● getPrice()
- ● getDescription()
- ● updateProductDetails()

contains
0..*
1

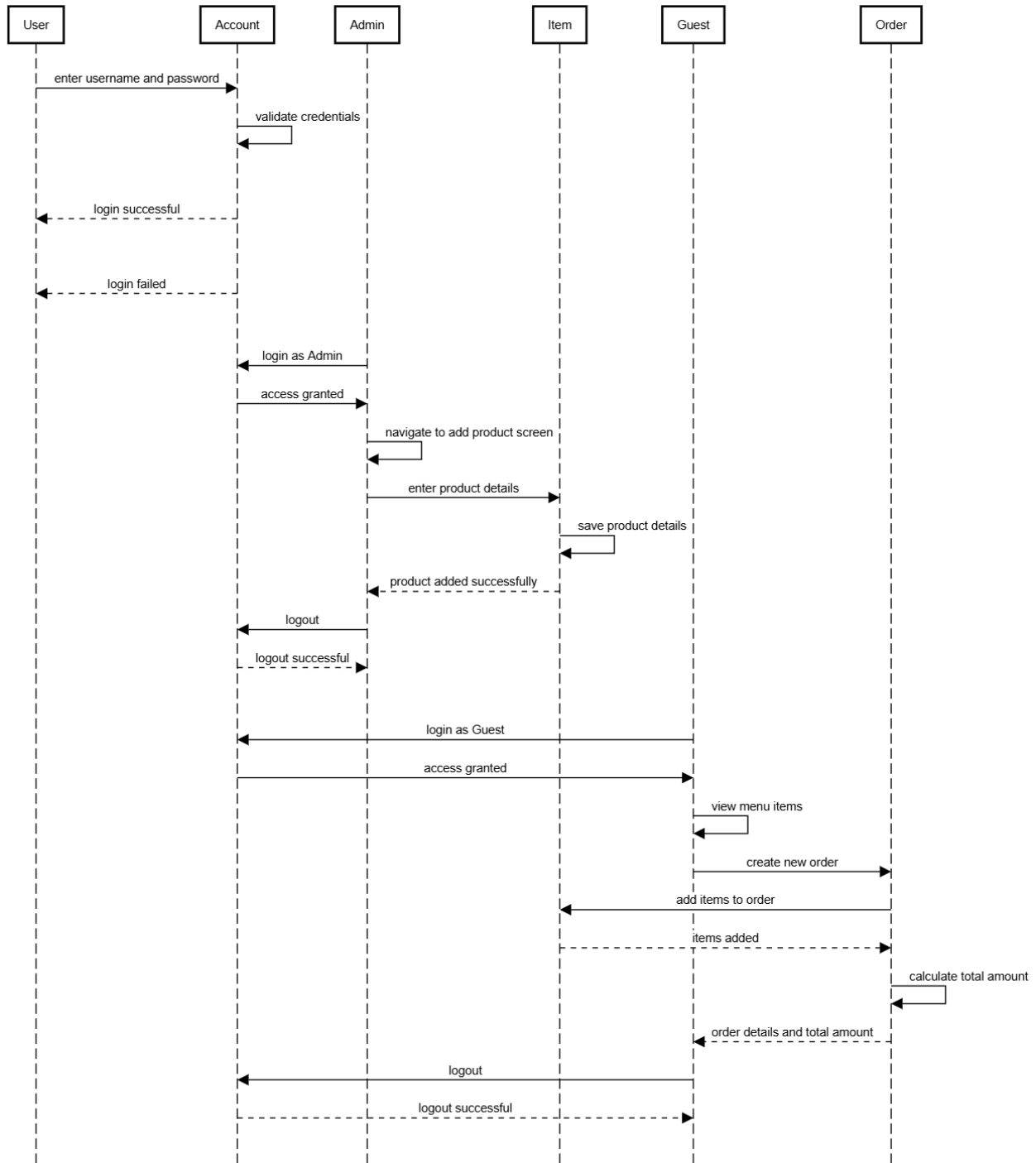**Sequence Diagram**
(Our initial sequence diagram, will update as we go)

Store Products Management System

Database Design (ERD)

(Our initial ERD , will update as we go)

Store Products Management System

Orders
account_id	int fk
total_amount	decimal(10, 2)
order_date	timestamp
order_id	int pk

Order_Items
order_item_id	int pk
order_id	int fk
product_id	int fk
quantity	int

Items
product_id	int pk
name	varchar(100)
price	decimal(10, 2)
description	text

Accounts
username	varchar(50)
password	varchar(255)
role	enum('Admin', 'Guest')
account_id	int pk

Admins
admin_id	int pk
account_id	int fk

Guests
guest_id	int pk
account_id	int fk