American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Software Engineering

Project Title

Meat Bazar

Project proposal submit to

MD. RAIHAN MAHMUD

Project proposal submit by:

Name	Id
MD.Nazmus Sadat Numan	22-48497-3
MD Doad	22-48506-3
Hasibul Islam Hasib	22-48507-3
Redwanul Haque Mazumder	22-48481-3

Project proposal:

The rising price of meat in Bangladesh has become a significant issue for consumers, particularly for middle and lower-income groups. Meat, which is an essential source of protein, has become increasingly unaffordable due to the presence of syndicates that artificially increases prices. These syndicates operate by purchasing livestock (cows, goats, and chickens) from farmers at a low cost and then raising the market price significantly before selling the meat to consumers. This practice has led to decrease in meat consumption among the general population, affecting their nutrition and well-being.

Despite the availability of livestock at a fair price from farmers, consumers are forced to pay much higher rates due to the control of intermediaries in the meat supply chain. The absence of a direct supply system between farmers and consumers allows middlemen to manipulate prices, contributing to market inefficiencies. Consumers have no choice but to accept these inflated prices, while farmers receive only a fraction of the final selling price, leading to an unfair distribution of profits.

To address this issue, our project, “Meat Bazar”, introduces a digital marketplace that directly connects farmers, distributors, and consumers, eliminating unnecessary middlemen. By integrating a structured supply chain within an online platform, Meat Bazar ensures that consumers can purchase meat at a lower price than the current market rate while allowing farmers to receive fair compensation for their livestock. This project aligns with the growing trend of digital marketplaces that prioritize transparency, affordability, and direct trade.

The root cause of this problem is the lack of a direct-to-consumer meat supply system, which allows syndicates to exploit both farmers and consumers. Existing meat distribution channels rely on multiple intermediaries who drive up prices without adding significant value to the supply chain. The absence of price regulation and transparency further exacerbates this issue, leading to an unfair and inefficient market. Meat Bazar aims to eliminate these inefficiencies by providing a platform where farmers can sell directly to consumers through trusted distributors, ensuring fair pricing and accessibility to affordable meat.

Login Page

Functional Requirements

- Users (Admin, Collector, Distributor, Customer) can log in using their email/phone and password.

- The system verifies credentials against the database before granting access.

- Error messages appear for incorrect login details.

- A "Forgot Password" link directs users to the password reset page.

- Successful login redirects users to their respective dashboards (Admin, Collector, Distributor, Customer).

Non-Functional Requirements

- The login page must load within 5 seconds for a smooth user experience.

- Data transmission must be secure using encryption.

- The page must be responsive (compatible with desktops, tablets, and mobiles).

- Error messages should be clear and user-friendly to guide users.

Project Development Constraints

- Must use Firebase Authentication.

- The system must differentiate between Admin, Collector, Distributor, and Customer logins.

- The system must store previous login information securely.

Signup Page

Functional Requirements

- Users can create an account by providing their name, email, phone number, address, and password.

- Option to select a role (Admin, Collector, Distributor, or Customer) during registration.

- Email/phone verification via OTP (One-Time Password).

- Password must meet security requirements: At least 8 characters, including one uppercase letter, one number, and one special character.

- Users must agree to the terms and conditions before signing up.

- Upon successful signup, users are redirected to their respective dashboards.

Non-Functional Requirements

- Signup must be completed quickly for an efficient user experience.

- OTP verification should be delivered within 30 seconds.

- Secure data storage using encryption techniques.

- An intuitive user interface for easy navigation.

- Fully responsive design for desktops, tablets, and mobiles.

Project Development Constraints

- User roles (Admin, Collector, Distributor, Customer) must be properly defined.
- Integration with a custom email service.
- Compliance with data protection laws.

Forgot Password Page

Functional Requirements

- Users can enter their registered email or phone number to reset their password.
- The system sends an OTP or password reset link via email or phone.
- Users enter the OTP and create a new password (following security rules).
- A success message confirms that the password has been changed.
- Users can now log in with the new password.

Non-Functional Requirements

- The page must be secure to prevent unauthorized access.
- OTP should be delivered within 30 seconds for quick password recovery.
- The entire password reset process should be completed within 1 minute.
- The page must function smoothly on all devices (desktop, tablet, and mobile).

Project Development Constraints

- Limit password reset attempts per user to prevent spam.
- Requires integration with an email/SMS API (e.g., Firebase, Twilio).
- Provide clear success and error messages to guide the user.

Admin Requirements

Dashboard Features:

Functional Requirements

Personal Info

- Admin can access their personal details provided during signup.
- A back button allows returning to the dashboard.

Logout

- Admin can log out and return to the login page.
- A back button allows returning to the dashboard before logging out.

Manage Users (Collector, Distributor, Customer)

- Admin can view all registered Collectors, Distributors, and Customers.
- Admin can add new Collectors, Distributors, and Customers.
- Admin can remove any user from the system.
- Admin can view individual customer order histories.
- Admin can modify customer orders (keep or cancel).

Update Meat Inventory

- Admin can update the availability of meat types (Beef, Mutton, Chicken) in kg or full size (Cow, Goat, Chicken).

- Admin can set and update daily meat prices for customers.

Allocate Meat to Collectors

- Admin can assign specific quantities of meat (kg or full animal) to collectors.

- Admin can select which collector receives a specific allocation.

Manage Distributors

- Admin can view distributor activity related to order fulfilment and meat distribution.

- Admin can modify distributor records as needed.

Payment History

- Admin can view all payment transactions made by customers.

- Admin can track profits generated from Collectors and Distributors.

Non-Functional Requirements

- **System Performance:** The admin dashboard should load within 3 seconds for a smooth experience.

- **Data Security:** User and payment data must be encrypted and protected from unauthorized access.

- **Responsiveness:** The dashboard must be fully functional across desktop, tablet, and mobile devices.
- **Scalability:** The system should handle multiple user roles and transactions simultaneously.

- **Real-Time Updates:** Inventory updates, order status changes, and payment transactions should reflect instantly.

- **User Interface:** A clean and intuitive UI for easy navigation and management.

- **Error Handling**: Clear messages should be provided in case of incorrect data input or failed operations.

Project Development Constraints

- **Role-Based Access Control**: The system must differentiate Admins, Collectors, Distributors, and Customers with appropriate access levels.

- **Payment System Integration:** The system must support online payments (bKash, Nagad, bank cards) and cash on delivery.

- **Compliance with Data Protection Laws:** The system must follow local data security regulations to protect customer information.

- **Order Management Restrictions:** Admins can only modify customer orders before they are marked as completed.

- **Secure API Integrations:** Any external services (e.g., payment gateways, email/SMS notifications) must be securely integrated.

User Requirements

Dashboard Features:

Functional Requirements

Personal Info:

- Users can access their personal details provided during signup.

- A back button allows users to return to the dashboard.

Logout

- Users can log out and return to the login page.
- A back button allows users to return to the dashboard before logging out.

Order History

- Users can view their purchase history, including the type of meat, quantity, and total payment.
- A back button allows users to return to the dashboard.

Order Tracking

- Customers can track the status of their orders (Processing, Pending, Picked Up).
- The rider's name and phone number are displayed if the order is picked up.
- A back button allows users to return to the dashboard.

Meat Ordering

- Users can choose from three types of meat (Beef, Mutton, Chicken).
- Each selection leads to a new tab where customers can select quantity (in kg or full animal) and view pricing.
- A "Payment" button appears after selecting the quantity.
- Users must make a payment to confirm the order.

- A back button allows users to return to the dashboard.

Payment

- Users can choose payment options for Beef, Mutton, and Chicken.
- Supported payment methods include bKash, Nagad, bank card payments, and cash on delivery.
- A back button allows users to return to the previous selection page.

Non-Functional Requirements

- **System Performance:** The dashboard should load within 3 seconds for smooth navigation.
- **Security:** User data, especially personal details and payment information, must be securely encrypted.
- **Responsiveness:** The interface must be optimized for desktop, tablet, and mobile devices.
- **Scalability:** The system should support multiple users simultaneously without performance issues.
- **User-Friendly Interface:** Clear labels, intuitive navigation, and a simple layout for easy usability.
- **Payment Processing Reliability:** Transactions should be completed within 10 seconds for a seamless experience.
- **Real-Time Order Tracking:** The order status should be updated in real time.

Project Development Constraints

- **Integration with Payment Gateways:** The system must support bKash, Nagad, and bank card transactions.

- **Data Protection Compliance:** Must follow relevant data protection laws for securing user information.

- **Error Handling:** The system must provide clear error messages in case of failed transactions or incorrect user input.

- **Real-Time Data Syncing:** Order status and payment confirmations must update instantly.

- **Limited Order Cancellation:** Customers should only be able to cancel orders within a specific time frame before processing.

Distributor Requirements

Dashboard Features:

- Order Overview: See pending, in-progress, and completed orders at a glance.
- Meat Stock Levels: Monitor available meat stock and expected deliveries.
- Customer Orders: View new and pending customer orders in real time.
- Payment Tracking: Track payments made to collectors and received from customers.
- Notifications & Alerts: Receive updates on new orders, payment confirmations, and inventory updates.
- Sales & Performance Reports: View insights on total sales, most ordered items, and revenue trends

Order Management

Functionality:

- Meat Supply Requests: Distributors can request specific types and quantities of meat from collectors.
- Order Status Tracking: Real-time updates on requested meat from collectors.
- Customer Orders Management: Distributors can view and manage customer orders.
- Order History Access: Distributors can track past orders to manage inventory efficiently.
- Order Modification & Cancellation: Modify or cancel requests before processing.
- Automated Invoicing: Generate invoices for received supplies and customer sales.
- Payment Tracking: Monitor transactions with collectors and customers.

Non-Functionality:

- Instant Order Notifications: Get notified when a collector confirms or processes an order.
- Bulk Order Support: Handle large order requests from multiple customers efficiently.
- Seamless Workflow: Ensure an efficient order request, confirmation, and delivery system.

Project Development Constraints (PDC):

- Secure Order Data Storage: Encrypt and protect transaction records.
- Real-Time Logistics Integration: Track inbound and outbound orders.

- Automated Inventory Updates: Update available stock dynamically.
- Scalable System: Ensure smooth performance even during peak demand.

Collector Requirements

Dashboard Features:

- Distributor Requests: View incoming meat requests from distributors.
- Pending Orders: Monitor orders that need to be processed and dispatched.
- Payment Status: Track payments received from distributors and pending payments.
- Meat Supply History: Review past orders fulfilled for distributors.
- Notifications & Alerts: Receive real-time updates on new distributor requests and payments.
- Financial Overview: View total earnings and payment transactions received from the admin.

Order Management

Functionality:

- View Distributor Requests: Collectors can see supply requests from distributors.
- Order Processing: Accept, modify, or reject distributor requests.
- Order Fulfillment Tracking: Update the status of meat orders in real time.
- Order History Access: Track past fulfilled orders for reference.
- Generate Payment Slips: Provide distributors with payment receipts.
- Payment Confirmation: Verify received payments from distributors.

Non-Functionality:

- Instant Order Notifications: Receive alerts when a distributor places an order.
- Bulk Order Processing: Handle large meat supply requests efficiently.
- Optimized Delivery Coordination: Ensure smooth logistics for fulfilling distributor requests.

Project Development Constraints (PDC):

- Secure Transaction Data: Protect financial records with encryption.
- Automated Payment Reconciliation: Match received payments with processed orders.
- Real-Time Updates: Ensure instant updates on payment status and order completion.