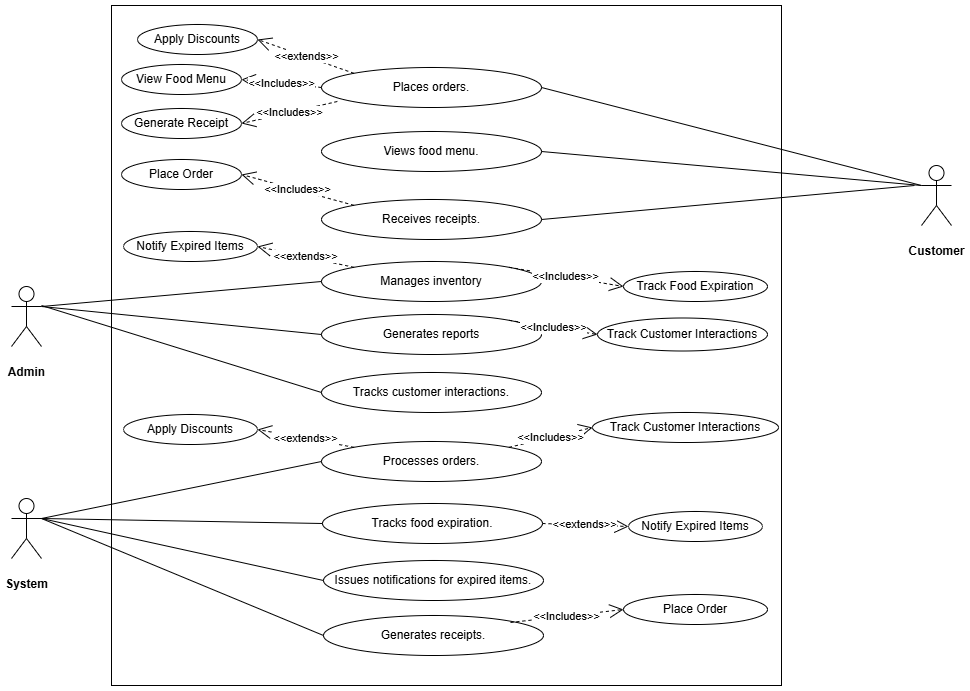
1.use case diagram



2.Recuriments

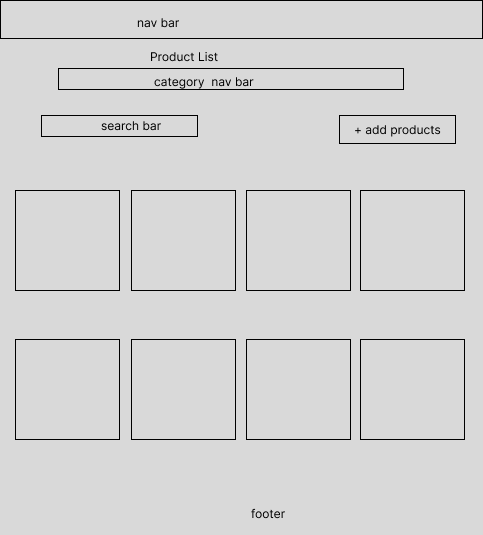
**Functional Requirements**

1. **Order Management**:
   * Place orders by selecting food items from the menu.
   * Calculate order total, including discounts.
   * Display item details like price, quantity, and expiration warnings.
   * Allow customers to search for orders using phone numbers or order IDs.
   * Generate and display receipts for completed orders (PDF format).
2. **Menu and Inventory Management**:
   * View all food items categorized (e.g., burgers, fries, beverages).
   * Add, update, and delete food items in the inventory.
   * Track item details such as price, quantity, and expiration dates.
   * Notify the admin of expired food items and allow removal of expired items.
3. **Customer Management**:
   * Add customer details during order placement or through a separate process.
   * Update, edit, and delete customer details.
   * View order history for specific customers.
4. **Report Generation**:
   * Generate monthly sales reports.
   * Identify top customers based on order frequency.
   * Generate annual sales reports and food item sales reports.
5. **User Interaction**:
   * Enable users to search for food items by name, code, or category.
   * Allow admins to view notifications for expired items or stock shortages.

**Non-Functional Requirements**

1. **Performance**:
   * The system should handle up to 100 concurrent users without significant delays.
   * Page loading time should not exceed 3 seconds on average.
2. **Responsiveness**:
   * The web application must be responsive and display properly on desktops, tablets, and smartphones.
3. **Scalability**:
   * The system should be capable of accommodating an expanding menu and growing customer base.
4. **Usability**:
   * Use a clean, intuitive interface with clear navigation for customers and admins.
   * Provide visual indicators (e.g., buttons, icons) for key actions like placing orders or editing inventory.
5. **Security**:
   * Ensure secure handling of sensitive customer data using HTTPS.
   * Implement role-based access control for admin features.
6. **Maintainability**:
   * Follow modular coding practices for ease of updates and bug fixes.
   * Use standardized frameworks and libraries for frontend (e.g., Bootstrap) and backend development.
7. **Reliability**:
   * The system should have 99.9% uptime and handle unexpected errors gracefully with appropriate notifications.
8. **Accessibility**:
   * Ensure compliance with basic web accessibility standards (e.g., WCAG) for usability by all users, including those with disabilities.
9. **Documentation**:
   * Provide user manuals for both customers and administrators.
   * Maintain technical documentation for system maintenance and updates.
10. **Cross-Browser Compatibility**:
    * The system should function correctly on major browsers (e.g., Chrome, Firefox, Safari, Edge).

3.wireframes

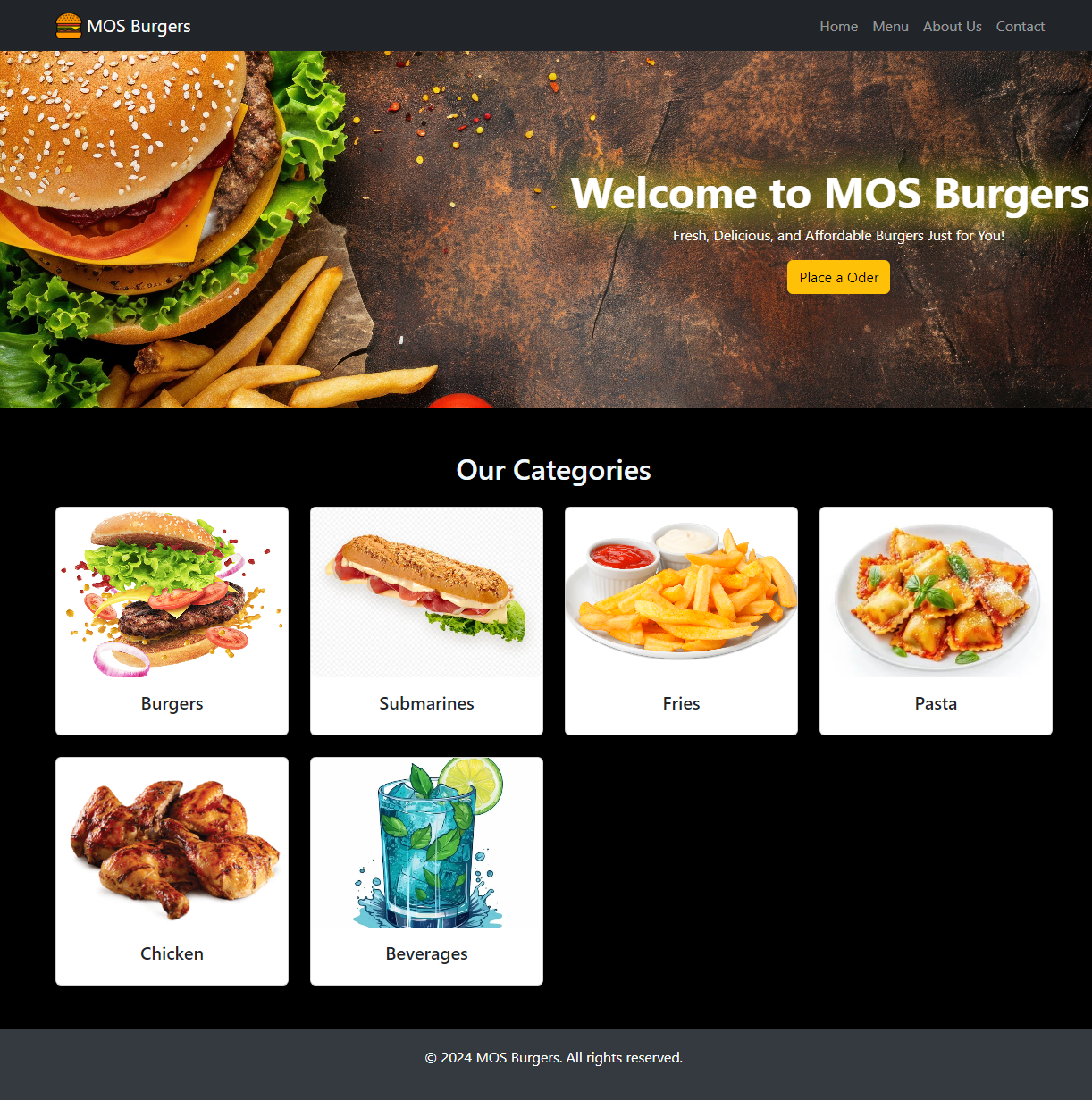
A screenshot of a computer

Description automatically generatedA screenshot of a web page

Description automatically generated

4.prototypes

i.home page



ii.product list page

A screenshot of a menu

Description automatically generated

iii.place order page

A screenshot of a menu

Description automatically generated