Software Requirements Specification (SRS)

Project Title: DIEATO – Healthy Food Ordering App

Submitted By: Sariya Mazhar

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to specify the functional and non-functional requirements for the **DIEATO** application, which allows users to order healthy, diet-focused meals online. This app ensures convenience, balanced nutrition, portion control, and an easy ordering experience.

1.2 Scope

- **DIEATO** is a web-based application designed to:
- Allow customers to explore a curated menu of healthy meals.
- Provide features like cart management, secure ordering, and user authentication.
- Support informative sections like **About Us** and **Contact Us**.
- Enable users to place and track their orders efficiently.
- The app will include the following major modules:
- **User Module**: Registration, login, and profile management.
- **Menu Module**: Explore categories, search food items, and filter by diet preferences.
- **Cart Module**: Add/remove items, view cart, checkout.
- **Order Module**: Place an order and confirm payment.
- **Information Module**: About Us, Contact Us, customer support.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirement Specification

UI: User Interface

DIEATO: The app name (Diet + Eat)

1.4 References

[i] IEEE 830-1998 standard for SRS documentation.

[ii] www.w3schools.com W3 schools for the learning.

[iii] www.reactjs.com ReactJS official documentation.

[iv] Youtube reference for certain functions

2. Overall Description

2.1 Product Perspective

Type: Web-based application (React frontend).

Users: Customers seeking healthy food options.

External Interfaces: Payment gateway (future integration), Email/phone support.

2.2 Product Functions

Explore menu (category-wise food display).

Add items to cart and modify quantity.

Place orders with address and payment details.

Contact customer support through form or details.

View app information (About Us).

Login/sign-up popup for user authentication.

2.3 User Characteristics

Primary Users: Health-conscious individuals, students, working professionals.

Users are expected to have basic familiarity with mobile/web apps.

2.4 Constraints

Requires an internet connection.

Supported on modern browsers (Chrome, Edge, Firefox).

Secure login and data handling required.

2.5 Assumptions and Dependencies

Payment system integration will use a third-party provider (e.g., Razorpay/Stripe).

Meal data is maintained by admin in backend.

Delivery system depends on third-party logistics partners.

3. Specific Requirements

3.1 Functional Requirements

1. User Authentication

FR1.1: The system shall allow users to sign in/sign out.

FR1.2: The system shall provide a login popup modal.

2. Menu Exploration

FR2.1: The system shall display food categories (Breakfast, Healthy Meals, Juices, etc.).

FR2.2: The user shall be able to filter food by category

FR2.3: The system shall allow users to view food item details (image, name, description, price).

3. **Cart Management**

FR3.1: Users shall be able to add/remove food items.

FR3.2: Users shall be able to update item quantities.

FR3.3: The system shall show total price dynamically.

4. **Order Placement**

FR4.1: Users shall be able to provide delivery details.

FR4.2: Users shall be able to confirm an order.

FR4.3: Users shall receive order confirmation.

5. **Information & Support**

FR5.1: The system shall display **About Us** information.

FR5.2: The system shall display **Contact Us** info (location, email, phone).

FR5.3: The contact form shall allow users to send messages.

3.2 Non-Functional Requirements

Performance: The system should load the menu page within 3 seconds.

Security: User credentials must be encrypted.

Usability: Simple and intuitive UI for quick ordering.

Reliability: The app must be available 99% of the time.

Scalability: The system should support multiple simultaneous users.

---

4. External Interface Requirements

UI Interfaces:

Navbar for navigation.

Menu section with clickable categories.

Cart and order summary interface.

Hardware Interfaces: Standard desktop/laptop/mobile device with internet.

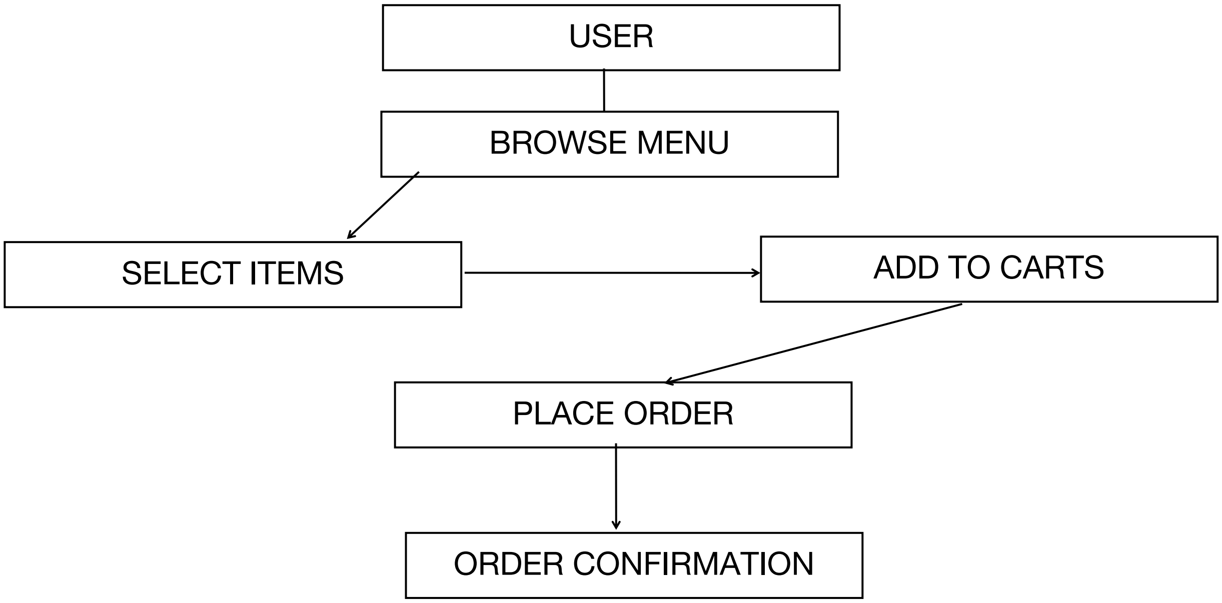
Software Interfaces: React frontend, API backend, database (future integration).

5. System Models

5.1 Use Case Diagram (Textual Representation)

User → Browse Menu → Select Items → Add to Cart → Place Order → System processes → Order Confirmation.

5.2 Data Hierarchy:



6. Future Enhancements

- Integration with real payment gateways.
- Real-time order tracking.
- Subscription-based diet plans.
- AI-based meal recommendations.