PROJECT: RESTONAV

SYSTEM REQUIREMENTS SPECIFICATIONS

TEAM 5

2141158 Riya Mary Cleetus

2141156 Parvathy D

2141165 Kriti Bajpai

PROBLEM DEFINITION

This web application eases the users to find restaurants in different cities within the country. Restaurant owners can add in their restaurants in the website and hence promote their restaurants. The surplus food in the restaurants can be transported to the poverty areas if and when notified by the restaurant owners to us.

The main problem addressed by this project is twofold:

1. Difficulty in Finding Restaurants: Many users face challenges in finding suitable restaurants in different cities within the country. With numerous dining options available, users often struggle to efficiently explore and select restaurants that meet their preferences, leading to time-consuming and potentially frustrating experiences.
2. Food Surplus and Food Waste: Restaurants often have surplus food at the end of the day, which can go to waste if not managed properly. This food waste contributes to environmental issues and inefficient resource utilization. At the same time, there are poverty-stricken areas where people struggle to access sufficient food to meet their basic needs.

The RestoNav web application aims to address these problems by offering a user-friendly platform that allows users to easily find restaurants in various cities. By providing a centralized location for restaurant information, it streamlines the process of discovering dining options, thereby enhancing the overall user experience.

Furthermore, the platform provides an opportunity for restaurant owners to promote their establishments to a wider audience, potentially attracting more customers and boosting their businesses.

In addition to helping users and restaurant owners, RestoNav also tackles the issue of food surplus and waste. It allows restaurant owners to notify the platform when they have excess food, creating a system to transport this surplus food to poverty-stricken areas. By facilitating the redistribution of surplus food to those in need, the web application aims to contribute to social change and make a positive difference by reducing food waste and helping underprivileged communities.

Overall, RestoNav endeavors to bridge the gap between diners and restaurants while promoting more sustainable and socially responsible practices within the food industry.

REQUIREMENTS SPECIFICATION

NON-FUNCTIONAL REQUIREMENTS

1. User Interface:

The user interface should be intuitive, visually appealing, and responsive for different screen sizes and devices.

2. Performance:

The application should load restaurant listings and details quickly, even with a slow internet connection.

The search functionality should provide results promptly, even when dealing with a large number of restaurants.

3. Security:

User data should be stored securely and transmitted over encrypted channels.

Proper authentication and authorization mechanisms should be in place to protect user accounts.

4. **Offline Access**:

The application should have limited offline access, allowing users to view previously accessed restaurant details and favourite restaurants.

5. Compatibility:

The application should be compatible with both Android and iOS platforms.

6. Data Accuracy and Integrity:

The restaurant information and user reviews should be regularly updated and validated for accuracy.

Assumptions:

Users have access to a reliable internet connection to fetch real-time data.

GPS or location services are available on the user's device.

Users will have the necessary permissions to use location services and access the camera (for uploading photos, if applicable).

FUNCTIONAL REQUIREMENTS

Functional requirements describe what a system or software application should do in terms of its features and functionalities. In the context of the given database schema for a restaurant reservation and surplus food management system, here are some functional requirements:

1. User Management:

Allow users to register and create an account with a unique username, email, and password.

Provide user roles to distinguish between general users, restaurant owners, and administrators.

Enable users to log in and log out of their accounts.

2. Restaurant Management:

Allow restaurant owners (users with the role "restaurant owner") to add and manage their restaurants.

Enable restaurant owners to update restaurant details, such as name, contact information, address, cuisine type, and website URL.

Support the association of each restaurant with a specific city from the City Table.

3. Menu Management:

Allow restaurant owners to add, edit, and delete menu items for their respective restaurants.

Enable the inclusion of menu item details, such as name, description, price, and category.

4. Reservation Management:

Enable users to make reservations at a particular restaurant, specifying the date, time, and party size.

Allow restaurant owners to view, confirm, or cancel reservations for their establishments.

Provide users with the ability to view their reservation details and modify or cancel their reservations.

5. Review and Rating System:

Allow users to write reviews and rate restaurants based on their dining experiences.

Display restaurant ratings and reviews to users for making informed decisions.

Provide restaurant owners the ability to respond to user reviews.

6. Surplus Food Management:

Allow restaurant owners to list surplus food items with details such as food item, quantity, expiry date, and notification date.

Notify restaurant owners about surplus food items approaching their expiry dates.

Provide users or NGOs with the ability to view and request surplus food items for donation or transportation.

7. Food Transportation:

Allow NGOs to view available surplus food items and request transportation of specific items.

Enable restaurant owners to accept or decline transportation requests from NGOs.

Track and record the details of each food transportation operation, including the transportation date and destination area.

8. City Management:

Allow administrators to manage (add, edit, delete) cities and their corresponding names in the system.

These functional requirements outline the key features and actions that the restaurant reservation and surplus food management system should support to meet the needs of its users and stakeholders.

SYSTEM REQUIREMENTS

1. Development Environment:

- ✓ Operating System: Windows, macOS, or Linux
- ✓ Processor: Multi-core processor (e.g., Intel Core i5 or higher)
- ✓ RAM: Minimum 8GB RAM (16GB or more recommended for smoother performance)
- ✓ Storage: Sufficient free disk space for code, assets, and development tools
- ✓ Web Browser: Latest versions of popular browsers like Chrome, Firefox, or Edge

2. Web Application Technology Stack:

- ✓ Web Development Framework: Choosing a framework that is JavaScript-based frameworks.
- ✓ Database: Set up a relational database (MySQL)

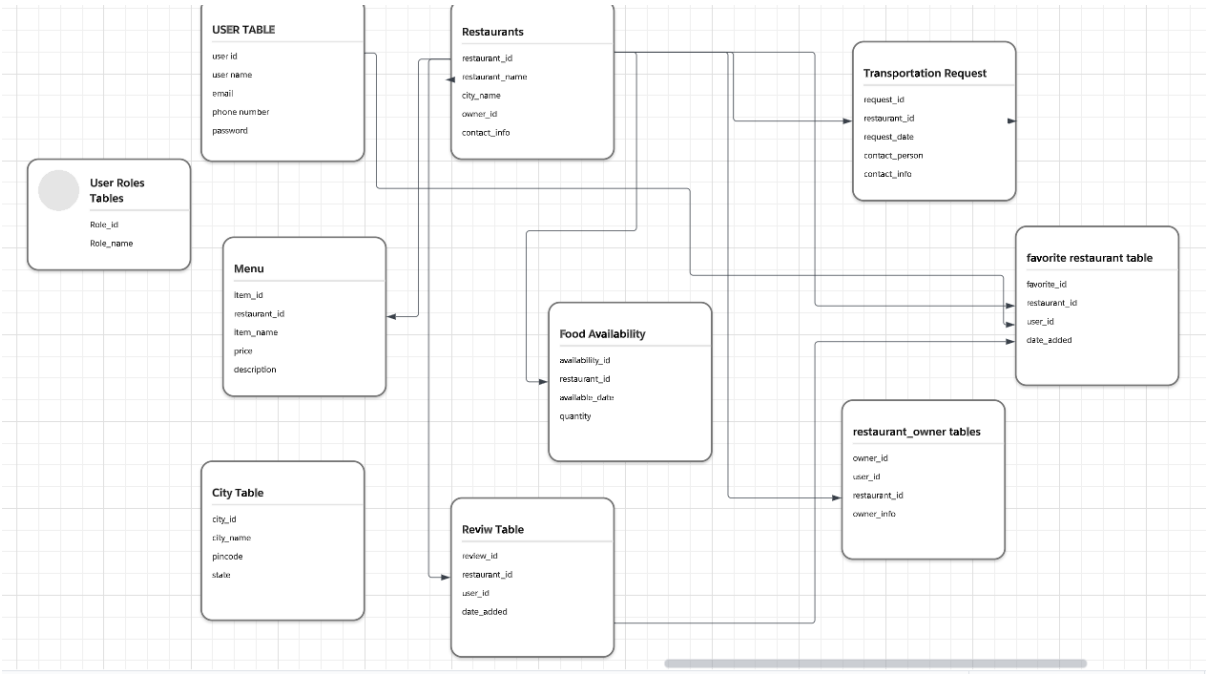
3. Target Deployment Environment:

- ✓ Web Hosting: Choosing a reliable web hosting service that supports your chosen backend technology and provides sufficient server resources.
- ✓ Server Requirements: Ensure that your web hosting plan meets the minimum requirements of your chosen web development framework and backend technology.
- ✓ Bandwidth: Consider the expected number of users and data transfer requirements to select an appropriate hosting plan with adequate bandwidth.

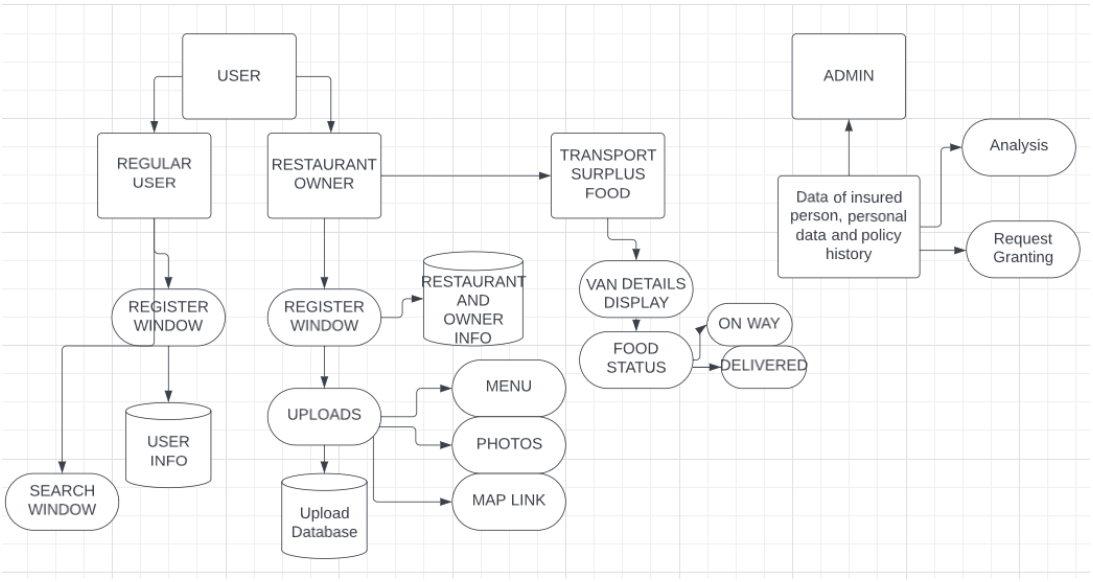
4. Security Considerations:

- ✓ Implement best security practices to protect user data and prevent vulnerabilities, such as SQL injection and cross-site scripting (XSS).
- ✓ Regularly update software, frameworks, and libraries to patch security vulnerabilities.

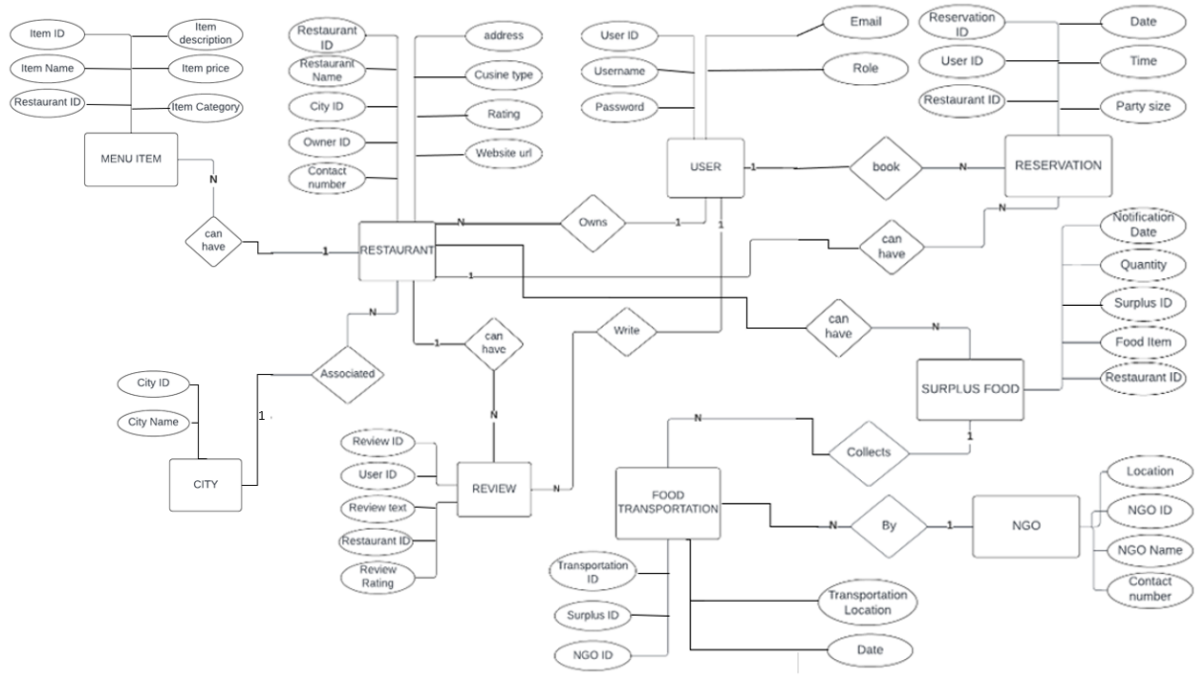
CONCEPTUAL MODEL



DATA FLOW DIAGRAM



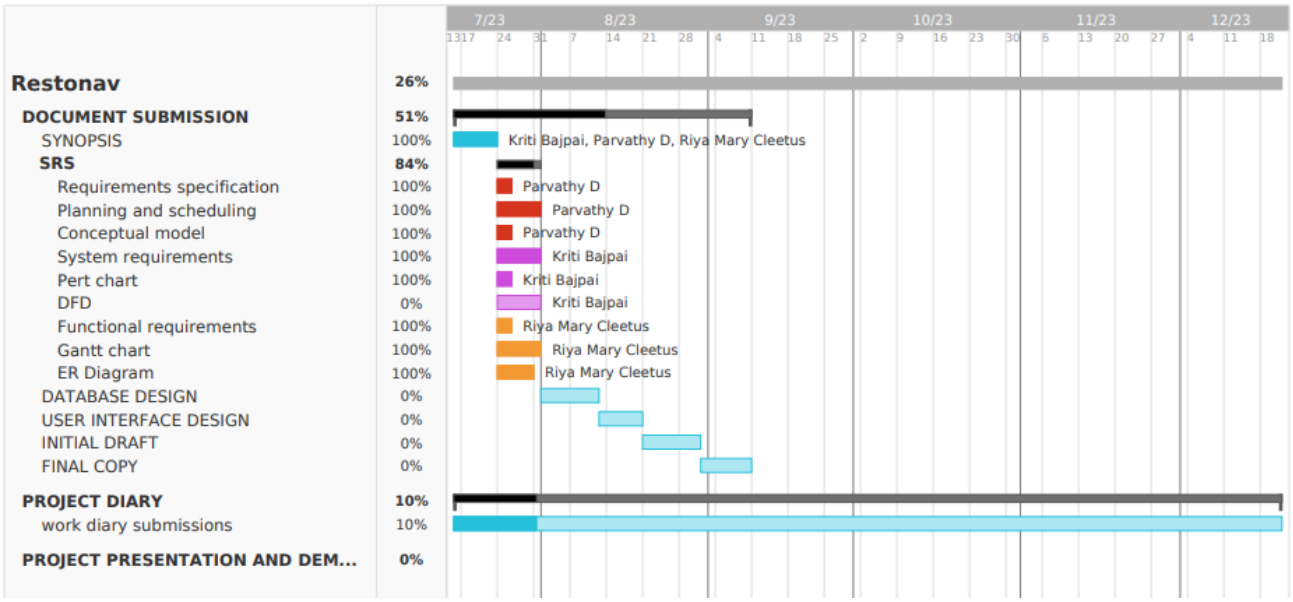
ER DIAGRAM



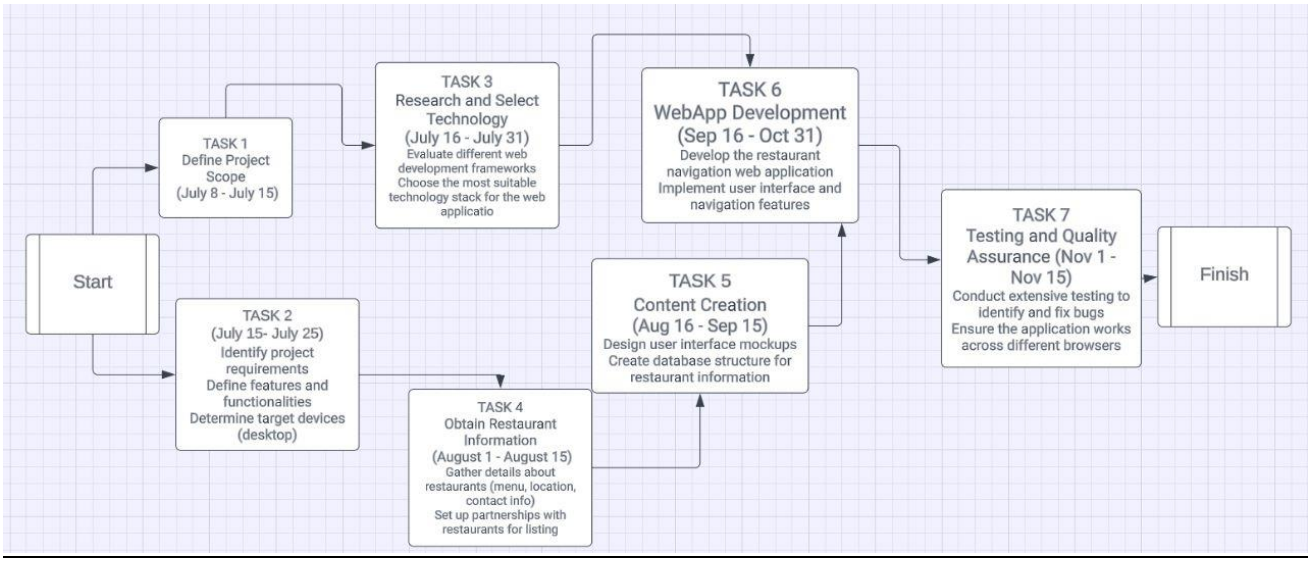
PLANNING AND SCHEDULING

PLANNING	SCHEDULING
Project Initiation	July 6th
Synopsis	July 8th-july 11th
Software requirement specification	July 15th-july 24th
Database design	July 26th-aug 10th
User Interface (UI)	Aug 10th -aug 21st
Backend System Architecture and Database Design	Aug 23rd -sept 4th
Frontend Development	Sept 5th -sept 22nd
Backend Development	Sept 22nd -oct 16th
Testing and Quality Assurance	Oct 17th- oct 24th
User Acceptance Testing (UAT)	Oct 26th – nov 2nd
Project Presentation and Demonstration	Nov 8th

GANTT CHART



PERT CHART



TOOLS PROPOSED:

Font-end: HTML, Python, Javascript, CSS.

Back-end: Mysql, php.

Other: gps.