# DishDash

**1. Stakeholder (Customer):**

   - The stakeholder is a non-profit organization dedicated to reducing food waste and supporting local communities.
   - Their mission is to create a user-friendly platform that connects food donors, such as restaurants and grocery stores, with local charities and individuals in need.

**2. Goals:**

  - Enumerate the goals of the application:
    1. Facilitate food donations from businesses and individuals.
    2. Connect donors with nearby food banks and charities.
    3. Raise awareness about food waste and encourage more people to donate.
    4. Provide an intuitive and visually appealing user interface.

  - How will the application support the customer's goals:
   - The application will provide an easy-to-use platform for food donation, including a map-based interface to locate nearby donation centers, a donation scheduling system, and educational content on food waste reduction.

**3. Success Measures:**
  - Quantitatively measure success with the following metrics:
    1. Number of successful food donations per month.
    2. User engagement metrics, such as the number of registered users and their activity levels.
    3. Reduction in food waste within the community, measured by the weight of donated food.
    4. User satisfaction ratings through feedback forms.

  - Desired values for these measures:
    - Increase food donations by 20% per month.
    - At least 5,000 registered users within the first year.
    - Achieve a 10% reduction in food waste within the community annually.
    - Maintain an average user satisfaction rating of 4 out of 5.

**4. Competition Analysis:**

   - Similar platforms include "**too good to go**". They have successfully attracted a user base and facilitated donations, but many lack user-friendly interfaces and educational content and availability. Users expect a seamless and intuitive donation process, clear communication, and information about the impact of their donations.

## 5. Functional Requirements
  - Describe the required functionality of the application, focusing on what the app will do.

  - Functions:
    1. User Registration and Profiles:
       - Allow users to create accounts and manage their profiles.
       - Enable donors and charities to provide necessary information.

    2. Donation Listings:
       - Display available donations, including type, quantity, and location.
       - Allow donors to create listings and set availability times.

    3. Donation Scheduling:
       - Enable users to schedule pickups or deliveries for donations.
       - Send notifications and reminders for scheduled events.

    4. Interactive Map:
       - Display donation centers, charities, and available donations on a map.
       - Enable users to search for nearby options.

    5. Educational Resources:
       - Provide articles, videos, and infographics on food waste reduction.
       - Raise awareness and educate users about the cause.

    6. Feedback and Support:
       - Offer a feedback mechanism for users to report issues or provide suggestions.
       - Provide responsive customer support.

## 6. Non-Functional Requirements:
  - **Performance:**
    - The application should load within 3 seconds on average.
    - It should be able to handle concurrent users without significant performance degradation.

  - **Security:**
    - Implement strong encryption for user data and payment information (if applicable).
    - Regularly update and patch security vulnerabilities.
    - Implement user authentication and authorization mechanisms.

  - **Availability:**
    - Aim for 99.9% uptime, with planned maintenance communicated in advance.
    - Implement redundancy and failover mechanisms to ensure availability.

  - **Software Quality:**
    - Adhere to coding standards and best practices.
    - Implement automated testing for critical functionalities.

- Conduct regular code reviews to maintain code quality.

- **Documentation:**
  - Maintain comprehensive documentation for developers and users.
  - Provide clear instructions for using the platform and its features.

- **Maintainability:**
  - Develop the application in a modular and maintainable manner.
  - Ensure that future updates and feature additions can be easily integrated.

- **Scalability:**
  - Design the application architecture to scale horizontally and vertically as needed.
  - Monitor resource utilization and scale accordingly.

- **Accessibility:**
  - Ensure the platform is accessible to users with disabilities, following WCAG guidelines.

- **Compliance:**
  - Adhere to relevant data protection regulations, such as GDPR.

## 7. Schedule:

- **Milestone 1: Project Kickoff (1 week)**
  - Define project scope, goals, and requirements.
  - Create a project plan and assign roles.

- **Milestone 2: Front-End Design (2 weeks)**
  - Create wireframes and mockups for the user interface.
  - Design the user experience and visual elements.

- **Milestone 3: User Registration and Profiles (3 weeks)**
  - Implement user registration and login functionality.
  - Develop user profile management features.

- **Milestone 4: Donation Listings (4 weeks)**
  - Design and implement the donation listing interface.
  - Enable users to create and manage donation listings.

- **Milestone 5: Donation Scheduling and Map (4 weeks)**
  - Develop donation scheduling features.
  - Implement the interactive map with location-based features.

- **Milestone 6: Educational Resources (2 weeks)**
  - Create and populate the educational content section.

- **Milestone 7: Feedback and Support (2 weeks)**
  - Implement feedback submission and support mechanisms.

- **Milestone 8: Testing and Quality Assurance (4 weeks)**
  - Conduct thorough testing of all functionalities.
  - Address any bugs and issues found during testing.


- **Milestone 9: Documentation (2 weeks)**
  - Create user documentation and developer guides.


- **Milestone 10: Deployment and Launch (1 week)**
  - Prepare for production deployment.
  - Launch the platform to the public.


- **Milestone 11: Post-launch Monitoring and Optimization (Ongoing)**
  - Continuously monitor performance and user feedback.
  - Make necessary improvements and optimizations.


- **Milestone 12: Compliance and Security Updates (Ongoing)**
  - Regularly update and maintain security features and compliance with regulations.