



CS 383 – Group Project

Software Requirements Specification

SRS

for:

MealBridge

Presented to:

Dr. Rafaa Aljarbua

Table of Contents

1. Introduction

1. Purpose
2. Product scope
3. References
4. Structure

2. System Overview

1. Product perspective
2. Product features
3. User roles and characteristics
4. Operating environment
5. Design and implementation constraints
6. Assumptions and dependencies

3. Requirements Engineering

1. Requirements elicitation
2. Requirements analysis
3. Requirements validation

4. Functional Requirements

1. Requirement
2. Requirement
3. Requirement

5. Non-functional Requirements

4. Performance Requirements
5. Safety Requirements
6. Security Requirements
7. Software Quality Attributes

6. External Interface Requirements

1. User Interface requirements
2. Hardware Interface requirements
3. Software Interface requirements
4. Communication Interface requirements

Introduction

1.1. Purpose

The purpose of this document is to provide a detailed Software Requirements Specification (SRS) for the MealBridge application. This document outlines the functional and non-functional requirements, serving as a comprehensive guide for the development, testing, and deployment teams. The MealBridge application aims to address the issue of food waste by creating a cross-platform solution that efficiently connects surplus food sources with those in need.

1.2. Product scope

MealBridge is a cross-platform software that will serve as a technological bridge to connect surplus food donors with beneficiaries (e.g., charities, non-profit organizations, and volunteers). The system will enable donors, including restaurants, bakeries, and individuals, to post available surplus food. It will also allow beneficiaries, such as charities and volunteers, to access this information to organize and facilitate the donation process. The primary objective is to reduce food waste and support communities by providing an effective tool for managing food donations.

1.3. References

- **Benchmarking of Existing Food Donation Applications:**
Analysis of similar regional applications including *نادي بركة*, *نادي يوم*, and *نادي الخفاف* to understand common features, user flows, and best practices in donation and distribution systems.
- <https://www.lucidchart.com/pages/tutorial/uml-use-case-diagram>

1.4. Structure

This document is organized into five main sections:

- Introduction: Provides an overview of the project, its purpose, and the scope of the application.
- System Overview: Provides a high-level description of the product, its features, user classes, and operating environment.
- Requirements Engineering: Details of the methodology used to gather and analyze requirements.
- Functional Requirements: Specifies what the system must do.
- Non-functional Requirements: Defines quality attributes and constraints.
- External User Interface: Describes the user, hardware, software, and communication interfaces.

All requirements are uniquely identified for traceability.

System Overview

MealBridge is a comprehensive digital platform designed to tackle the dual challenges of food waste and food insecurity within communities. The system serves as a critical link between food donors such as restaurants, businesses, and individuals and verified charitable organizations. By leveraging technology to automate the donation matching and management process, MealBridge aims to ensure that surplus food is redirected quickly, safely, and efficiently to those in need. This section provides a detailed overview of the MealBridge system.

2.1. Product perspective

MealBridge is a digital donation management system connecting food donors with charitable organizations. It operates as an independent platform accessible across different devices and relies on external services for location matching and secure cloud-based data management. The system automates the donation process, including matching, communication, and tracking, to ensure efficiency, transparency, and safety. It forms part of a broader social initiative aimed at reducing food waste and promoting community welfare through technology. MealBridge is designed to be scalable, reliable, secure, and easy to use for all users.

2.2. Product features

The main features of MealBridge include the following:

- 1. User Registration and Authentication:** Secure sign-up and login for donors, charities, and administrators. Role-based access to ensure appropriate permissions for each user type.
- 2. Donation Creation and Management:** Donors can create donation posts including meal details, quantity, expiry date, and pickup location. Option to upload food photos and verify expiry for safety.
- 3. Automated Matching System:** the system automatically connects donors with nearby charities based on real-time location data.
 - o Notifications are sent instantly to both parties when a match occurs.
- 4. Donation Tracking and Confirmation:** Real-time status updates for donation progress (pending, matched, collected, completed).
 - o Charities confirm pickup and receipt through the app.
- 5. User Profiles and Ratings:** Each user can view and update their profile. Feedback and rating system for transparency and trust building.

2.3. User roles and characteristics

Donors:

Individuals, families, or restaurants that have extra meals to donate. They generally have basic technical skills and require a simple and intuitive interface. Their primary goal is to submit donation requests easily and quickly.

Charity Organizations / Volunteer Teams:

Authorized organizations responsible for receiving and distributing donated meals. They typically have moderate technical experience and require features that support reviewing requests, scheduling pickup times, and updating donation statuses.

System Administrator:

A technically experienced user responsible for managing the application's backend activities. This includes verifying organization accounts, monitoring system performance, handling reports, and managing data integrity.

2.4. Operating environment

- The system will operate as a mobile application available on both Android and iOS platforms.
- Users are required to have internet connectivity to submit and receive real-time updates on donation requests.
- The application will rely on cloud-based storage to store user accounts, donation details, and communication logs securely.
- The system will run on devices with standard hardware specifications and will support a wide range of smartphone models.

2.5. Design and implementation constraints

Mobile Platform Requirement

The system must be developed as dedicated, high-performance applications for mobile devices (smartphones). This is a mandatory constraint because the app needs guaranteed, fast access to device features like GPS and the camera to support core functions such as location matching and photo uploads.

Performance Constraint

All essential user actions, including user login, new donation creation, and the automated matching process, must be completed within 3 seconds. The system must also be capable of supporting at least 100 simultaneous users without performance degradation.

Security and Data Protection Constraint

The system is strictly required to use secure, encrypted connections (like HTTPS) for all data sent over the internet. Furthermore, sensitive user information, specifically recipient addresses and phone numbers, must be protected by encryption when stored in the database.

Food Safety Assurance Constraint

The system design must include mandatory verification steps (such as required photo uploads and confirmation of expiry dates) during the donation process to ensure the quality and safety of the donated food.

2.6. Assumptions and dependencies

2.6.1. Assumptions

User Willingness and Participation: It is assumed that there is a sufficient and active user base of corporate/individual donors and verified charity organizations willing to use the application in the initial target operating area.

User Technical Proficiency: It is assumed that users possess a smartphone with an active internet connection and have the basic ability to use mobile applications to create or claim donations.

Data Integrity and Honesty: It is assumed that donors will accurately and truthfully provide information regarding the food safety, preparation time, and expiry dates of the donated meals.

2.6.2. Dependencies

Reliance on Recipient Response: The fast delivery of donated food depends heavily on the charity organizations or volunteers responding quickly to the match notification and being ready to pick up the food immediately. If they are slow, the system's goal of reducing food waste will be hard to achieve.

Need for Reliable Map Services: The core functions of the app, such as finding the closest charity to a donation and calculating the travel distance, entirely rely on the accurate performance of external map services. If the map service fails or provides bad data, the matching system will not work correctly.

External Verification for Charities: To keep the platform trustworthy, the system needs an official, external method to check and confirm that the charity organizations joining the app are legitimate.

Requirements Engineering

This section outlines the process of identifying, analyzing, and validating the requirements for the MealBridge application. By using a combination of structured techniques and user-centered methods, the team ensured that the system's features align with real user needs and the project's overall objectives. The following subsections describe how requirements were gathered, refined, and confirmed to form a solid foundation for the development of MealBridge.

3.1 Requirements Elicitation

To collect and define the requirements of *MealBridge*, several techniques were used:

1. Brainstorming Sessions:

The project team held multiple brainstorming sessions to generate ideas about the app's main features and functionality. During these sessions, we discussed how the system can effectively connect food donors (restaurants, bakeries, individuals) with receivers (charities and volunteers). The main goal was to identify what each user type needs and how to make the process simple and efficient.

2. Observation of Similar Applications:

We observed existing applications and websites that focus on food donation and waste reduction (e.g., *حنطة النعمة*, *بلدو*) to understand their strengths, weaknesses, and user experience designs. This helped us identify common features such as donor listings, pickup scheduling, and notification systems, and notice areas where we could improve, such as simplifying the registration process and supporting Arabic language.

3. Interviews and Informal Discussions (4–6 participants):

Short interviews and informal discussions were conducted with 4–6 potential users to gather insights about their expectations and challenges regarding food donation and collection. These conversations helped the team understand the basic needs of both donors and receivers, as well as identify key features that would make the app simple and practical to use.

These methods allowed us to gather both functional and non-functional requirements to guide the development of *MealBridge*.

3.2 Requirements Analysis

After collecting the requirements, the team analyzed and categorized them to ensure clarity and consistency:

- **Classification:** Requirements were divided into functional and non-functional categories.
- **Feasibility Check:** Each requirement was reviewed to determine whether it could realistically be implemented within the project's scope, time, and technical constraints.
- **Conflict Resolution:** Any overlapping or conflicting requirements from different stakeholders were discussed and refined during team meetings.
- **Prioritization:** The most essential features such as donor registration, listing donations, and notifying nearby charities were given higher priority, while secondary features (like donation history) were marked for future implementation.

3.3 Requirements Validation

To confirm that the gathered requirements accurately represent user needs:

- **Team Review:**

The team collaboratively examined all requirements to ensure they were understandable, complete, and aligned with the project scope.

- **Requirements Checklist Review:**

The team validated requirements using a checklist focusing on completeness, clarity, consistency, testability, and feasibility.

- **Cross-Verification with Similar Systems:**

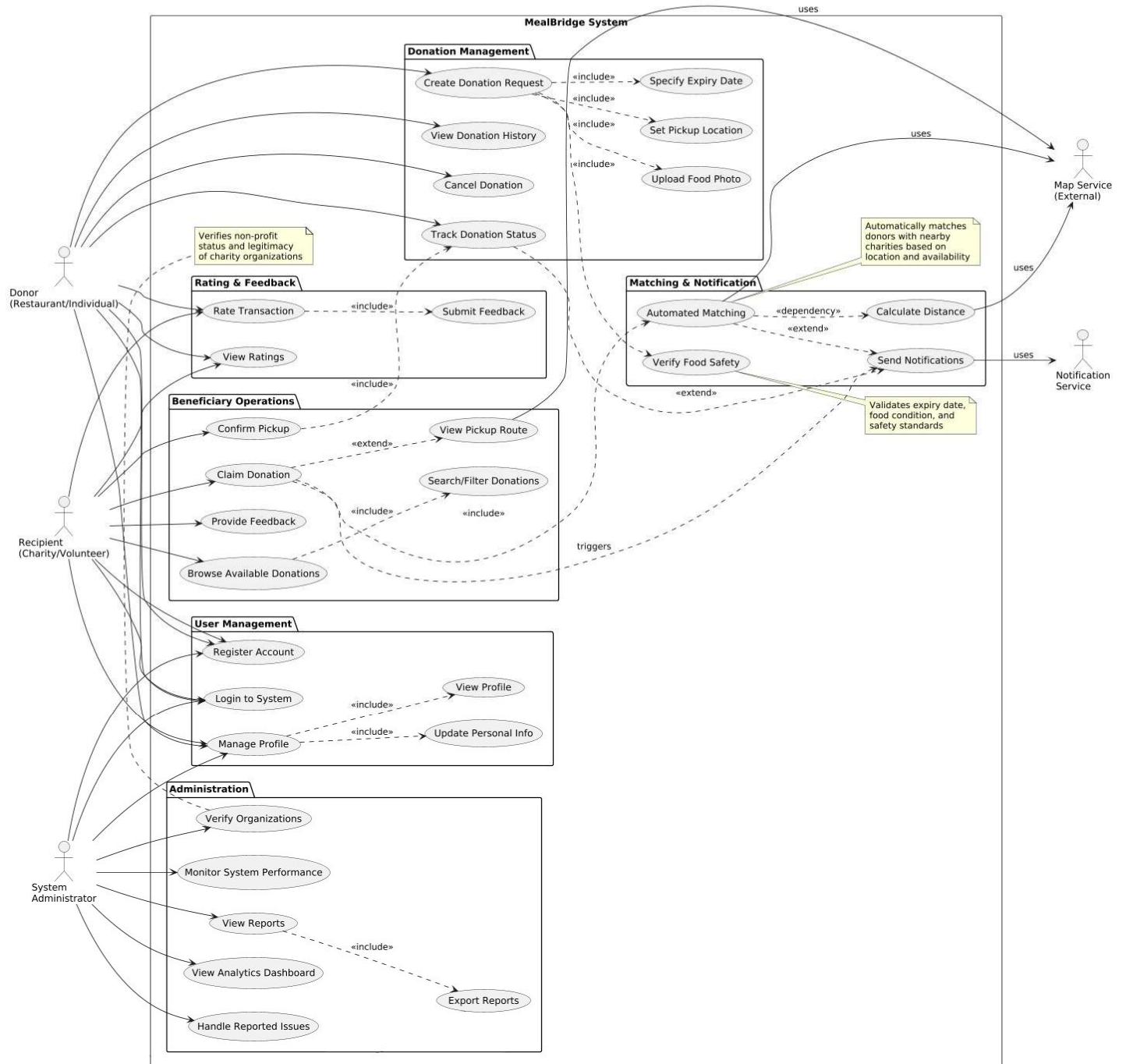
Requirements were compared with features from existing food donation platforms to ensure they align with industry norms and avoid missing essential functions.

- **Goal Alignment Check:**

Each requirement was evaluated against MealBridge's main objective reducing food waste and efficiently connecting donors with recipients to confirm it contributes directly to the system's purpose.

UML Use Case Diagram

MealBridge System - Use Case Diagram



Functional Requirements

R-1: User Registration and Login

Description: The system shall allow users to register and log in using their credentials.

Rationale: To ensure authorized access and protect user data.

Priority: High.

R-2: Donation Request Creation

Description: The system shall enable donors to create donation requests by specifying meal details, quantity, and pickup location.

Rationale: To allow donors to share excess meals easily and efficiently.

Priority: High.

R-3: Automated Matching System

Description: The system shall automatically match donation requests with nearby charity organizations and notify both parties.

Rationale: To reduce manual coordination and ensure efficient food distribution.

Priority: Medium.

R-4: Donation Tracking and Updates

Description: The system shall provide real-time tracking for donation status and send updates to both donor and receiver.

Rationale: To enable users to easily track the donation process and improve the experience for both parties.

Priority: Medium.

R-5: User Profile Management

Description: The system shall allow users to create, view, and edit their profiles. For donors (businesses/individuals), this shall include organization name, address, and food types typically offered. For beneficiaries (charities/volunteers), this shall include organization name, address, proof of non-profit status (optional but encouraged), and types of food needed.

Rationale: To build user identity and trust within the platform, and to enable better, more relevant matching.

Priority: High.

R-6: Beneficiary Request Management

Description: The system shall allow beneficiaries to view available donations and make claim donation requests.

Rationale: To ensure a smooth and verifiable food handover process.

Priority: High.

R-7: Search and Filter Donations

Description: The system shall allow beneficiaries to search and filter available donation requests based on criteria such as food type, quantity, location proximity, and pickup time.

Rationale: To help beneficiaries quickly find suitable donations that meet their specific needs and capacity.

Priority: High.

R-8: Notification Management

Description: The system shall send push notifications and/or email alerts for new donation matches, pickup reminders, status updates, and system announcements. Users shall be able to customize notification preferences.

Rationale: To keep users informed in real-time and reduce missed donations while allowing users to control notification frequency.

Priority: Medium.

R-9: Reporting and Analytics

Description: The system shall provide an admin dashboard with reports showing key metrics such as total meals donated, top donors, active beneficiaries, and food waste reduced (in meals).

Rationale: To measure the impact of the platform, provide valuable insights to stakeholders, and identify areas for improvement.

Priority: Low.

R-10: Food Safety and Expiry Verification

Description: The system shall allow donors to input the expiry date and food condition details when submitting a donation. The system shall automatically verify that the donation meets predefined safety criteria (e.g., not expired, prepared within safe time limits).

Rationale: To ensure all shared meals comply with food safety standards and protect beneficiaries' health.

Priority: High.

R-11: Feedback and Rating System

Description: The system shall allow both donors and beneficiaries to provide feedback and rate their experience after each completed donation transaction.

Rationale: To promote accountability, transparency, and continuous improvement in the donation process.

Priority: Medium.

Non-Functional Requirements

5.4. Performance Requirements

Description: The system shall process user requests (e.g., login, donation creation, and matching) within an average of **3 seconds**. It shall support at least **100 concurrent users** without noticeable performance degradation.

Rationale: To ensure fast and responsive interaction for all users, improving user satisfaction.

Priority: High.

5.5. Safety Requirements

Description:

The system shall include automatic **data backup** and recovery mechanisms to prevent data loss in case of server failure. It shall also prevent accidental deletion of important records by requiring confirmation before removal.

Rationale: To maintain data integrity and prevent data loss due to errors or system crashes.

Priority: Medium.

Description: The system shall detect common runtime errors (e.g., failed form submissions, invalid inputs) and provide clear, user-friendly error messages. It shall also automatically retry critical operations when safe to do so.

Rationale: To prevent user frustration, reduce data loss, and ensure a smooth recovery from unexpected failures.

Priority: High.

5.6. Security Requirements

Description: The system must protect sensitive recipient information (such as addresses and phone numbers) by encrypting it when stored in the database. It must also use a secure connection (like HTTPS) when transferring data over the internet.

Rationale: To protect user privacy and prevent unauthorized access to personal information.

Priority: Critical.

5.7. Software Quality Attributes

Description: The system shall be compatible with major web browsers (Google Chrome, Safari and Edge) and support both desktop and mobile devices. The web interface shall be responsive and maintain full functionality on various screen sizes.

Rationale: To ensure accessibility and consistent experience for all users regardless of their device or platform.

Priority: Medium.

Description: The system shall support Arabic and English languages for both the interface and notifications, allowing users to switch their preferred language at any time.

Rationale: To increase accessibility and usability for users with different language preferences.

Priority: Medium.

External Interface Requirements

1.1. EIR-1: User Interface Requirements

Description:

MealBridge shall provide a responsive, user-friendly interface for mobile and web users with role-specific views for Donors, Recipients (charities/volunteers), and Administrators.

Requirements:

- The system shall provide native mobile apps (Android & iOS) and a responsive web interface that adapts to screen widths from 320px to 1920px.
- The UI shall support Arabic and English; users must be able to switch language.
- The UI shall present role-based navigation: Donor flows (Create Donation, View History), Recipient flows (Browse Donations, Claim), and Admin flows (Verify Users, Reports).
- The donation reporting flow (create donation → upload photo → set pickup info → confirm) shall be completable within three main steps or screens for usability.

1.2. EIR-2: Hardware Interface Requirements

Description:

MealBridge shall interface with mobile device hardware to capture location, images, and scan/confirm donation pickups.

Requirements:

- The mobile app shall request and use GPS/location services to detect user location (with user consent) for proximity matching and route suggestions.
- The mobile apps shall access the device camera and photo library to allow donors to attach images of donations.
- Works on Android and iOS devices with internet access.
- The app shall use device notification hardware (sound, vibration) to alert users of urgent pickup requests or confirmations.

1.3. EIR-3: Software Interface

Description:

MealBridge will integrate with third-party services for maps, authentication, storage, and notifications to provide core functionality.

Requirements:

- The backend shall be integrated with a mapping service (e.g., Google Maps API) to display maps, compute distances, and estimate routes and ETA.
- The system shall use a secure authentication provider (e.g., Firebase Authentication, OAuth 2.0 with Google/Apple sign-in) for account management.
- The system shall store images and large objects in cloud storage and record metadata in a cloud database service (e.g., Firestore)
- The system shall support push notifications using platform services.

1.4. EIR-4: Communication Interface Requirements

Description:

MealBridge shall use secure, standardized communication protocols between clients, backend services, and external APIs.

Requirements:

- All communication between the app and the server, and between the server and other services, shall be secure using standard internet encryption (HTTPS).
- The system shall provide a way for the app to send and receive information from the server using standard web requests (e.g., sending donation details, updating profiles, or performing admin tasks).
- Real-time updates (e.g., live donation availability, delivery tracking) shall be implemented via WebSocket or Firebase Realtime mechanisms.
- The system shall make sure that only authorized users can access their accounts and limit excessive requests to prevent misuse.

1.5. EIR-5: System/Partner Interfaces Requirements

Description:

MealBridge shall provide mechanisms for partner charities, admin staff, and third-party systems to exchange data for reporting, verification, and bulk integration.

Requirements:

- The system shall provide an Admin web panel that interfaces with the backend for user verification, donation approvals, and analytics dashboards.
- Authorized partner organizations shall be able to pull or push data via a secured API (endpoints for donation status, pickup confirmations, and distribution results) using API keys and role-based access.
- The system shall support exporting reports in CSV formats (e.g., weekly donation volume, source breakdown, hygiene incidents) for partner review and regulatory needs.

Contribution of each member:

Section	Student
Introduction	Muzna Abdelgadir
System Overview	2.1, 2.2 – Aroob Altuwajiri 2.3, 2.4 – Balqees Almohesn 2.5, 2.6 – Maha Alrashidi
Requirement Engineering	Bedor Alharbi & Muzna Abdelgadir
UML Use Case Diagram	Bedor Alharbi
Functional Requirements	Balqees Almohsen & Muzna Abdelgadir & Lena Alswed
Non-Functional Requirements	Aroob Altuwajiri & Maha Alrashidi
External Interface Requirements	Bedor Alharbi