Recipe Sharing Platform - Software Requirements Specification (SRS)

Version: 1.0
Date: June 22, 2025
Project: TasteBud Recipes Platform
Client: TasteBud Technologies

Developer: Ibtisam Al Hinai

---

1. Introduction

1.1 Purpose

This document describes the requirements for the Recipe Sharing Platform where users can share, discover, and rate cooking recipes.

1.2 Scope

The platform includes:

- User registration and login
- Recipe creation and management
- Recipe browsing and search
- Recipe rating system
- Basic admin features
- Web interface (MVC) and API

1.3 Definitions

- **API: Application Programming Interface**
- **CRUD: Create, Read, Update, Delete**
- **ERD: Entity Relationship Diagram**
- **MVC: Model-View-Controller**
- **SRS: Software Requirements Specification**
- **UI/UX: User Interface/User Experience**

- **Recipe: A set of cooking instructions including ingredients, steps, and metadata**
- **Rating: A 1-5 star evaluation of a recipe by a user**

2. Overall Description

2.1 User Types

- **Guest Users**: Can browse and search recipes
- **Registered Users**: Can create recipes and rate others' recipes
- **Administrators**: Can manage users and delete recipes

2.2 Operating Environment

- Web browsers (Chrome, Firefox, Safari, Edge)
- ASP.NET MVC and Web API
- SQL Server database
- Responsive design for mobile devices

2.3 Assumptions

- Users have internet access
- Modern web browsers with JavaScript
- English language only
- Text-only recipes (no images)
- Users will self-moderate content

3. Functional Requirements

3.1 User Management

FR-001: User Registration

Users can create accounts with email, password, and display name.

FR-002: User Login

Users can login with email and password to access protected features.

FR-003: User Profile

Users can view and update their profile information.

3.2 Recipe Management

FR-004: Create Recipe

Authenticated users can add recipes with:

- Title (required)
- Description
- Ingredients list (multiple ingredients, required)
- Step-by-step instructions (multiple steps, required)
- Prep time (max 240 minutes)
- Cook time (max 480 minutes)
- Servings (1-20)
- Difficulty (Easy, Medium, Hard)
- Category (Breakfast, Lunch, Dinner, Dessert, Snacks, Other)

FR-005: View Recipe

All users can view complete recipe details including average rating.

FR-006: Edit Recipe

Users can modify their own recipes only.

FR-007: Delete Recipe

Users can delete their own recipes only.

FR-008: Browse Recipes

All users can browse all recipes with pagination.

3.3 Search

FR-009: Search Recipes

Users can search recipes by title and individual ingredients.

3.4 Rating System

FR-011: Rate Recipe

Authenticated users can rate recipes 1-5 stars (one time only per recipe).

FR-012: View Ratings

Display average rating and total number of ratings for each recipe.

3.5 Home Page

FR-013: Latest Recipes

Home page shows 10 most recently added recipes.

FR-014: Top Rated Recipes

Home page shows highly rated recipes.

3.6 Admin Features

FR-015: Manage Users

Admins can view all users and deactivate accounts.

FR-016: Delete Recipes

Admins can delete any recipe from the platform.

4. Database Design

4.1 Database Tables

Users Table (ASP.NET Identity)

- Id (string, PK)
- Email (string, unique)
- PasswordHash (string)
- DisplayName (string)
- EmailConfirmed (bool)
- SecurityStamp (string)
- PhoneNumber (string)

- LockoutEnabled (bool)
- AccessFailedCount (int)
- FirstName(string)
- LastNamr(string)
- Country(string)

Categories Table

- CategoryId (int, PK, Identity)
- Name (string, 50) - Breakfast, Lunch, Dinner, Dessert, Snacks, Other

Recipes Table

- RecipeId (int, PK, Identity)
- Title (string, 200, required)
- Description (text)
- PrepTime (int, 0-240)
- CookTime (int, 0-480)
- Servings (int, 1-20)
- Difficulty (int) - 1=Easy, 2=Medium, 3=Hard
- CreatedAt (datetime, default: now)
- ModifiedAt (datetime)
- UserId (string, FK to Users.Id)
- CategoryId (int, FK to Categories.CategoryId)

Ingredients Table

- IngredientId (int, PK, Identity)
- RecipeId (int, FK to Recipes.RecipeId)
- IngredientText (string, 500, required)
- Quantity (int, required)

Instructions Table

- InstructionId (int, PK, Identity)
- RecipeId (int, FK to Recipes.RecipeId)
- StepNumber (int, required)
- InstructionText (text, required)

Ratings Table

- RatingId (int, PK, Identity)
- RecipeId (int, FK to Recipes.RecipeId)
- UserId (string, FK to Users.Id)
- RatingValue (int, 1-5)
- CreatedAt (datetime, default: now)
- Unique constraint on (RecipeId, UserId)

4.2 Relationships

- User → Recipes (One-to-Many)
- User → Ratings (One-to-Many)
- Recipe → Ratings (One-to-Many)
- Recipe → Ingredients (One-to-Many)
- Recipe → Instructions (One-to-Many)
- Category → Recipes (One-to-Many)

4.3 Sample Data

Categories will be pre-populated with: Breakfast, Lunch, Dinner, Dessert, Snacks, Other.

---

5. Non-Functional Requirements

5.1 Performance

- Page load time: under 3 seconds
- Support 100 concurrent users
- Database queries: under 1 second

5.2 Security

- Secure password hashing (ASP.NET Identity)
- Role-based authorization
- Input validation on all forms

5.3 Usability

- Simple, clean interface
- Mobile responsive design
- Large, clickable buttons
- Easy-to-read recipe format

5.4 Reliability

- 99% uptime
- Graceful error handling
- User-friendly error messages

6. Use Cases

UC-001: Register New User

Actor: Guest User
Steps:

1. User goes to registration page
2. User enters email, password, display name
3. System validates and creates account
4. User redirected to login page

UC-002: User Login

Actor: Registered User
Steps:

1. User goes to login page
2. User enters email and password

3. System authenticates user
4. User redirected to dashboard

UC-003: Create Recipe

Actor: Authenticated User
Steps:

1. User goes to create recipe page
2. User fills in recipe details
3. User submits form
4. System saves recipe
5. User sees success message

UC-004: Search Recipes

Actor: Any User
Steps:

1. User enters search term
2. System searches recipes
3. System displays results
4. User can click to view recipe details

UC-005: Rate Recipe

Actor: Authenticated User
Steps:

1. User views recipe details
2. User clicks star rating
3. System saves rating
4. System updates average rating

UC-006: Admin Delete Recipe

Actor: Administrator
Steps:

1. Admin views recipe
2. Admin clicks delete button
3. System confirms action
4. System removes recipe

7. API Endpoints

Authentication

- POST /api/auth/register
- POST /api/auth/login

Recipes

- GET /api/recipes - Get all recipes
- GET /api/recipes/{id} - Get specific recipe
- GET /api/recipes/search?term={term} - Search recipes
- POST /api/recipes - Create recipe (auth required)
- PUT /api/recipes/{id} - Update recipe (auth required)
- DELETE /api/recipes/{id} - Delete recipe (auth required)

Ratings

- POST /api/recipes/{id}/rate - Rate a recipe (auth required)

8. Technical Architecture

8.1 Three-Tier Structure

- **Presentation**: MVC Controllers/Views, API Controllers
- **Business Logic**: Service classes with validation
- **Data Access**: Repository pattern with Entity Framework

8.2 Technology Stack

- ASP.NET MVC 5

- ASP.NET Web API 2
- Entity Framework 6 (Code First)
- ASP.NET Identity
- SQL Server
- HTML5, CSS3, JavaScript/jQuery

9. Constraints

9.1 Time Constraints

- 3-day development timeline
- Phase 1 features only

9.2 Feature Constraints

- No image uploads
- No recipe comments
- Static categories only
- One-time ratings only
- Admin can only delete (not edit) recipes

9.3 Technical Constraints

- Maximum 500 users initially
- Standard web hosting
- Modern browsers only

10. Acceptance Criteria

10.1 Functional

- User registration and login works
- Users can create, edit, delete own recipes
- All users can browse and search recipes

- Rating system functions correctly
- Admin can manage content
- Both MVC and API work

10.2 Technical

- Three-tier architecture implemented
- Entity Framework with Code First
- ASP.NET Identity authentication
- Responsive mobile design
- All API endpoints functional