

Software Requirements Specification (SRS)

Project Name: TasteBud Recipes

Version: 1.0

Date: June 22, 2025

Document Type: TasteBud Recipes SRS

1. Introduction

Purpose:

TasteBud Recipes is a community-driven platform where users can share, search, and rate cooking recipes.

Scope:

The system provides the following core features:

- User registration and login
 - Recipe CRUD (Create, Read, Update, Delete)
 - Rating and review system
 - Search by title and ingredients
 - Admin dashboard for user and recipe management
 - User profile management
-

2. User Types

Guest Users:

- Browse and search recipes
- View public user profiles

Registered Users:

- All guest user capabilities
- Add, edit, delete their own recipes
- Rate and review other users' recipes
- Manage personal profile and dashboard

Admin:

- Manage user accounts
 - Delete inappropriate content
 - View usage statistics and analytics
 - Add and manage Category
-

3. Functional Requirements

User Authentication

- Register, Login, Logout
- Password reset via email

User Profile

- View and update personal profile
- View other public profiles

Recipe Management

- Add, update, delete, and view recipes

Search Functionality

- Search recipes by title or ingredients
- Browse categories

Rating System

- Rate (1–5 stars), update, and view ratings
- Restrict self-rating and duplicate ratings

Admin Dashboard

- Manage users (add user – block user)
 - Manage Recipe(display and delete recipe)
 - Manage Recipe Category(add and delete)
-

4. Non-Functional Requirements

Performance

- Page load within 3s (95% of the time)
- Search responses under 2s
- API responds within 1s

Scalability

- Up to 50,000 users and 100,000 recipes
- Support 1,000 concurrent users

Security

- HTTPS encryption
- Hashed passwords
- Input validation (SQL Injection, XSS protection)
- Session timeouts
- Data encryption at rest

Availability

- 99.5% uptime
- Daily backups and disaster recovery tested

Accessibility

- Keyboard navigation
- Screen reader support
- High color contrast

Browser Compatibility

- Must work across modern web browsers (no mobile-specific design required)
-

5. API Endpoints (RESTful)

Authentication

- POST /api/auth/register
- POST /api/auth/login
- POST /api/auth/logout
- POST /api/auth/reset-password

Recipe Management

- GET /api/recipes
- POST /api/recipes
- GET /api/recipes/{id}
- PUT /api/recipes/{id}

- DELETE /api/recipes/{id}

Search and Discovery

- GET /api/search/recipes
- GET /api/categories
- GET /api/recipes/recent

Rating System

- POST /api/recipes/{id}/ratings
- GET /api/recipes/{id}/ratings
- PUT /api/ratings/{id}
- DELETE /api/ratings/{id}

User Management

- GET /api/users/profile
- PUT /api/users/profile
- GET /api/users/{id}/recipes
- DELETE /api/users/account

Admin

- GET /api/admin/users
 - DELETE /api/admin/users/{id}
 - GET /api/admin/recipes
 - DELETE /api/admin/recipes/{id}
 - GET /api/admin/analytics
-

6. Database Design

Users Table

- Id (PK)
- Username (Unique, Not Null)
- Email (Unique, Not Null)
- PasswordHash (Not Null)
- Role (User/Admin)
- IsActive (Boolean)
- CreatedAt, UpdatedAt (Timestamps)

Recipes Table

- Id (PK)
- Title (Not Null)
- Description, Ingredients, Steps (Text)
- AuthorId (FK to Users.Id)
- CreatedAt, UpdatedAt
- CategoryId (FK to Categories.Id)

Ratings Table

- Id (PK)
- RecipeId (FK)
- UserId (FK)
- Value (1–5)
- CreatedAt

Categories Table

- Id (PK)
- Name (Unique)

Relationships

- Users → Recipes (1:N)
- Users → Ratings (1:N)
- Recipes → Ratings (1:N)
- Recipes ↔ Categories (M:N)