

TasteBud Recipes Platform

Software Requirements Specification (SRS)

Document Version: 1.0

Date: June 22, 2025

Prepared for: TasteBud Technologies

Prepared by: Development Team

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the TasteBud Recipes platform requirements. It defines the functional and non-functional requirements for a web-based recipe sharing application that enables home cooks to share, discover, and rate recipes.

This document serves as the primary reference for developers, testers, and stakeholders throughout the development lifecycle.

1.2 Scope

The TasteBud Recipes platform is a web-based application that provides:

- User registration and authentication system
- Recipe creation, management, and sharing capabilities
- Recipe browsing and search functionality
- Recipe rating system
- Administrative content management features
- RESTful API for future mobile application integration

In Scope:

- Web application with responsive design
- User account management
- Recipe CRUD operations
- Search and browse functionality

- 1-5 star rating system
- Basic administrative features
- REST API implementation

Out of Scope (Phase 2):

- Image upload functionality
- Comments and reviews system
- Advanced content moderation
- Mobile applications
- Social media integration
- Payment processing

1.3 Definitions and Acronyms

Term	Definition
-------------	-------------------

API	Application Programming Interface
-----	-----------------------------------

CRUD	Create, Read, Update, Delete
------	------------------------------

MVC	Model-View-Controller
-----	-----------------------

REST	Representational State Transfer
------	---------------------------------

SRS	Software Requirements Specification
-----	-------------------------------------

UI/UX	User Interface/User Experience
-------	--------------------------------

Admin	Administrator user with elevated privileges
-------	---

Recipe	A set of cooking instructions with ingredients and metadata
--------	---

Rating	A 1-5 star evaluation of a recipe
--------	-----------------------------------

2. Overall Description

2.1 Product Perspective

TasteBud Recipes is a standalone web application designed to serve as a digital recipe sharing platform. The system consists of:

- **Web Frontend:** Responsive web interface built with ASP.NET MVC
- **Backend API:** RESTful Web API built with ASP.NET
- **Database:** Relational database for data persistence
- **Web Server:** Standard web hosting environment

The platform will integrate with modern web browsers and provide a foundation for future mobile application development.

2.2 User Classes and Characteristics

2.2.1 Anonymous Users

- **Description:** Visitors who have not registered or logged in
- **Characteristics:** Limited access, can only browse and search recipes
- **Technical Expertise:** Varied, from novice to expert
- **Primary Goals:** Discover recipes, evaluate platform before registration

2.2.2 Registered Users (Home Cooks)

- **Description:** Registered members who share and rate recipes
- **Characteristics:** Primary user base, ages 25-65, cooking enthusiasts
- **Technical Expertise:** Basic to intermediate computer skills
- **Primary Goals:** Share recipes, discover new recipes, build cooking reputation

2.2.3 Administrators

- **Description:** System administrators managing platform content
- **Characteristics:** Technical staff or trusted power users
- **Technical Expertise:** Advanced computer skills
- **Primary Goals:** Maintain platform quality, manage inappropriate content

2.3 Operating Environment

2.3.1 Client-Side Environment

- **Web Browsers:** Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
- **Devices:** Desktop computers, tablets, smartphones
- **Screen Resolutions:** 320px to 1920px+ width
- **Network:** Broadband or mobile internet connection

2.3.2 Server-Side Environment

- **Web Server:** IIS or compatible ASP.NET hosting
- **Framework:** ASP.NET Core MVC and Web API
- **Database:** SQL Server or compatible relational database
- **Operating System:** Windows Server or Linux with .NET support

2.4 Assumptions and Dependencies

2.4.1 Assumptions

- Users have basic computer literacy and internet access
- Users will provide accurate recipe information
- Initial user base will be primarily English-speaking
- Users will self-regulate content appropriately
- Standard web hosting infrastructure is available

2.4.2 Dependencies

- ASP.NET Core framework availability
 - Database server availability and reliability
 - Web hosting service uptime and performance
 - Third-party libraries for UI components
 - Browser support for modern web standards
-

3. Functional Requirements

3.1 User Management

3.1.1 User Registration

Requirement ID: FR-001

Priority: High

Description: The system shall allow new users to create accounts using email and password.

Detailed Requirements:

- Users must provide valid email address
- Password must meet minimum security requirements (8+ characters)
- Email verification is not required for initial version
- Duplicate email addresses are not allowed
- Registration form must validate input before submission
- System shall create user profile upon successful registration

3.1.2 User Login

Requirement ID: FR-002

Priority: High

Description: The system shall authenticate registered users securely.

Detailed Requirements:

- Users login with email and password combination
- System shall maintain secure session management
- Failed login attempts shall be tracked and limited
- Password shall be encrypted/hashed in database
- Session timeout after period of inactivity
- Remember me functionality for convenience

3.1.3 User Profile Management

Requirement ID: FR-003

Priority: Medium

Description: Users shall have individual profiles showing their recipe contributions.

Detailed Requirements:

- Profile displays user's created recipes
- Users can view their own profile and others' profiles
- Profile shows user registration date
- Profile displays total number of recipes created
- Basic profile information editing capabilities

3.2 Recipe Management

3.2.1 Create Recipe

Requirement ID: FR-004

Priority: High

Description: Authenticated users shall be able to create new recipes.

Detailed Requirements:

- Recipe creation form with following mandatory fields:
 - Recipe title (max 100 characters)
 - Description (max 1000 characters)
 - Ingredients list (structured format)
 - Step-by-step instructions
 - Preparation time (minutes)
 - Cooking time (minutes)
 - Number of servings
 - Difficulty level (Easy/Medium/Hard dropdown)
 - Category (Breakfast/Lunch/Dinner/Dessert/Snacks dropdown)
- Form validation for all required fields
- Recipes published immediately upon submission

- Auto-save draft functionality (optional)

3.2.2 View Recipe

Requirement ID: FR-005

Priority: High

Description: All users shall be able to view complete recipe details.

Detailed Requirements:

- Recipe detail page displays all recipe information
- Clear formatting for ingredients and instructions
- Display recipe metadata (times, servings, difficulty, category)
- Show recipe author and creation date
- Display current average rating and rating count
- Print-friendly view option
- Responsive design for mobile viewing

3.2.3 Update Recipe

Requirement ID: FR-006

Priority: High

Description: Users shall be able to edit their own recipes only.

Detailed Requirements:

- Edit functionality available only to recipe author
- Pre-populated form with existing recipe data
- Same validation rules as recipe creation
- Version history tracking (optional)
- Confirmation dialog before saving changes

3.2.4 Delete Recipe

Requirement ID: FR-007

Priority: High

Description: Users shall be able to delete their own recipes.

Detailed Requirements:

- Delete functionality available only to recipe author
- Confirmation dialog before deletion
- Soft delete with option to restore (admin feature)
- Associated ratings removed with recipe deletion
- Cascade delete for related data

3.3 Search Functionality

3.3.1 Recipe Search

Requirement ID: FR-008

Priority: High

Description: Users shall be able to search recipes by name and ingredients.

Detailed Requirements:

- Search by recipe title (partial match, case-insensitive)
- Search by ingredients (multiple ingredient support)
- Combined search criteria support
- Search results displayed in list format with key information
- Pagination for large result sets
- Sort options (relevance, rating, date, alphabetical)
- Search suggestions/auto-complete (optional)

3.3.2 Browse Recipes

Requirement ID: FR-009

Priority: High

Description: Users shall be able to browse all recipes with filtering options.

Detailed Requirements:

- Browse all recipes in paginated format
- Filter by category (Breakfast, Lunch, Dinner, Dessert, Snacks)
- Filter by difficulty level (Easy, Medium, Hard)
- Filter by preparation time ranges

- Sort options available
- Recipe preview with key information displayed

3.4 Rating System

3.4.1 Rate Recipe

Requirement ID: FR-010

Priority: High

Description: Authenticated users shall be able to rate recipes from 1-5 stars.

Detailed Requirements:

- Star-based rating interface (1-5 stars)
- One rating per user per recipe
- Users can update their existing rating
- Visual feedback during rating process
- Immediate update of average rating display

3.4.2 Display Ratings

Requirement ID: FR-011

Priority: High

Description: System shall calculate and display recipe ratings.

Detailed Requirements:

- Average rating calculated from all user ratings
- Display average rating with star visualization
- Show total number of ratings
- Rating information visible on recipe lists and detail pages
- Real-time rating updates

3.5 Admin Features

3.5.1 Admin Dashboard

Requirement ID: FR-012

Priority: Medium

Description: Administrators shall have access to content management dashboard.

Detailed Requirements:

- Admin-only accessible dashboard
- Overview statistics (total users, recipes, ratings)
- Recent activity summary
- Quick access to management functions

3.5.2 User Management

Requirement ID: FR-013

Priority: Medium

Description: Administrators shall be able to view and manage users.

Detailed Requirements:

- List all registered users
- View user profiles and activity
- Ability to deactivate user accounts
- Search users by email or username

3.5.3 Content Management

Requirement ID: FR-014

Priority: Medium

Description: Administrators shall be able to manage recipe content.

Detailed Requirements:

- View all recipes in system
 - Edit any recipe if necessary
 - Delete inappropriate recipes
 - Restore deleted recipes
 - Bulk operations for content management
-

4. Non-Functional Requirements

4.1 Performance Requirements

4.1.1 Response Time

Requirement ID: NFR-001

Priority: High

Description: System response times shall meet user expectations.

Specifications:

- Page load time: ≤ 3 seconds for standard pages
- Search results: ≤ 2 seconds for typical queries
- Recipe creation/update: ≤ 5 seconds for form submission
- API response time: ≤ 1 second for standard operations

4.1.2 Throughput

Requirement ID: NFR-002

Priority: High

Description: System shall support expected user load.

Specifications:

- Support 100-500 concurrent users initially
- Handle 1000+ recipe views per hour
- Process 100+ search queries per minute
- Support 50+ simultaneous recipe creations

4.1.3 Scalability

Requirement ID: NFR-003

Priority: Medium

Description: System architecture shall support future growth.

Specifications:

- Database design supports horizontal scaling
- Stateless application design for load balancing
- Efficient indexing for search performance

- Optimized queries to minimize resource usage

4.2 Security Requirements

4.2.1 Authentication Security

Requirement ID: NFR-004

Priority: High

Description: User authentication shall be secure and robust.

Specifications:

- Password hashing using industry-standard algorithms
- Secure session management with appropriate timeouts
- Protection against brute force attacks
- HTTPS encryption for all authentication processes

4.2.2 Data Protection

Requirement ID: NFR-005

Priority: High

Description: User data shall be protected from unauthorized access.

Specifications:

- Input validation and sanitization to prevent injection attacks
- Authorization checks for all user operations
- Secure API endpoints with proper authentication
- Protection of sensitive user information

4.2.3 Privacy

Requirement ID: NFR-006

Priority: Medium

Description: User privacy shall be maintained.

Specifications:

- Users can only access and modify their own recipes
- Email addresses not publicly displayed
- Basic privacy controls for user profiles

- Compliance with data protection best practices

4.3 Usability Requirements

4.3.1 Ease of Use

Requirement ID: NFR-007

Priority: High

Description: Interface shall be intuitive for target users.

Specifications:

- Navigation shall be clear and consistent
- Forms shall provide helpful validation messages
- Error messages shall be user-friendly and actionable
- Common tasks completable within 3 clicks

4.3.2 Accessibility

Requirement ID: NFR-008

Priority: Medium

Description: Platform shall be accessible to users with disabilities.

Specifications:

- Keyboard navigation support
- Screen reader compatibility
- Sufficient color contrast ratios
- Alternative text for images (when implemented)

4.3.3 Mobile Responsiveness

Requirement ID: NFR-009

Priority: High

Description: Interface shall work effectively on mobile devices.

Specifications:

- Responsive design for screens 320px to 1920px+ width
- Touch-friendly interface elements
- Readable text without zooming

- Efficient mobile navigation

4.4 Reliability Requirements

4.4.1 Availability

Requirement ID: NFR-010

Priority: High

Description: System shall be available when users need it.

Specifications:

- 99% uptime target
- Graceful degradation during peak loads
- Scheduled maintenance during low-usage periods
- Quick recovery from system failures

4.4.2 Data Integrity

Requirement ID: NFR-011

Priority: High

Description: User data shall be preserved accurately.

Specifications:

- Database transactions ensure data consistency
- Regular automated backups
- Data validation prevents corrupted entries
- Recipe content preserved exactly as entered by users

4.4.3 Error Handling

Requirement ID: NFR-012

Priority: Medium

Description: System shall handle errors gracefully.

Specifications:

- User-friendly error messages
- Logging of system errors for debugging
- Fallback options when features unavailable

- Prevention of data loss during errors
-

5. Use Cases

5.1 Use Case: Register New User

Use Case ID: UC-001

Title: Register New User

Primary Actor: Anonymous User

Goal: Create a new user account to access platform features

Preconditions: User has internet connection and valid email address

Success Guarantee: User account created and can log in immediately

Main Success Scenario:

1. User navigates to registration page
2. System displays registration form
3. User enters email address and password
4. User confirms password
5. User submits registration form
6. System validates input data
7. System creates new user account
8. System displays success message
9. User is automatically logged in
10. System redirects to user dashboard

Extensions:

- 3a. Email already exists: System displays error message
- 6a. Invalid email format: System displays validation error
- 6b. Password too weak: System displays password requirements
- 6c. Passwords don't match: System displays mismatch error

5.2 Use Case: Login

Use Case ID: UC-002

Title: User Login

Primary Actor: Registered User

Goal: Authenticate and access personalized platform features

Preconditions: User has registered account

Success Guarantee: User authenticated and logged into system

Main Success Scenario:

1. User navigates to login page
2. System displays login form
3. User enters email and password
4. User submits login form
5. System validates credentials
6. System creates user session
7. System redirects to dashboard or previous page

Extensions:

- 5a. Invalid credentials: System displays error message
- 5b. Account locked: System displays account status message
- 5c. Too many failed attempts: System temporarily locks account

5.3 Use Case: Create Recipe

Use Case ID: UC-003

Title: Create Recipe

Primary Actor: Registered User

Goal: Share a new recipe with the community

Preconditions: User is logged in

Success Guarantee: Recipe is published and available to all users

Main Success Scenario:

1. User navigates to create recipe page
2. System displays recipe creation form
3. User enters recipe title

4. User enters recipe description
5. User adds ingredients list
6. User enters step-by-step instructions
7. User sets preparation and cooking times
8. User specifies number of servings
9. User selects difficulty level
10. User chooses recipe category
11. User submits recipe form
12. System validates all required fields
13. System saves recipe to database
14. System displays success message
15. System redirects to recipe detail page

Extensions:

- 12a. Missing required fields: System displays validation errors
- 12b. Invalid time values: System displays format requirements
- 13a. Database error: System displays error message and preserves form data

5.4 Use Case: Search Recipes

Use Case ID: UC-004

Title: Search Recipes

Primary Actor: Any User (Anonymous or Registered)

Goal: Find recipes matching specific criteria

Preconditions: Recipes exist in the system

Success Guarantee: Relevant recipes displayed to user

Main Success Scenario:

1. User navigates to search page or uses search box
2. System displays search interface
3. User enters search terms (recipe name or ingredients)

4. User submits search query
5. System processes search request
6. System retrieves matching recipes
7. System displays search results with pagination
8. User clicks on recipe to view details

Extensions:

- 6a. No results found: System displays "no recipes found" message
- 6b. Too many results: System displays first page with pagination
- 3a. User uses advanced search: System provides category and difficulty filters

5.5 Use Case: Rate Recipe

Use Case ID: UC-005

Title: Rate Recipe

Primary Actor: Registered User

Goal: Provide feedback on recipe quality

Preconditions: User is logged in and viewing a recipe

Success Guarantee: Rating is recorded and average rating updated

Main Success Scenario:

1. User views recipe detail page
2. System displays current rating and rating interface
3. User clicks on star rating (1-5 stars)
4. System records user's rating
5. System updates recipe's average rating
6. System displays confirmation of rating submission
7. System updates rating display on page

Extensions:

- 3a. User has already rated: System updates existing rating
- 4a. Database error: System displays error message

- 7a. User wants to change rating: System allows rating update

5.6 Use Case: Admin Manage Content

Use Case ID: UC-006

Title: Admin Manage Content

Primary Actor: Administrator

Goal: Monitor and manage platform content for quality

Preconditions: User has administrator privileges

Success Guarantee: Content is reviewed and appropriate actions taken

Main Success Scenario:

1. Admin logs into system
2. System redirects to admin dashboard
3. Admin views recent recipes and user activity
4. Admin identifies potentially inappropriate content
5. Admin clicks on recipe to review details
6. Admin decides on appropriate action (edit, delete, or approve)
7. System executes admin action
8. System logs admin activity
9. System updates content status
10. System displays confirmation message

Extensions:

- 4a. No issues found: Admin continues monitoring
- 6a. Admin edits recipe: System updates recipe content
- 6b. Admin deletes recipe: System removes recipe from public view
- 7a. System error: Admin action fails, error logged for investigation