# 📘 Recipe Social Media App – Data Models & Firestore Design (Updated)

## Table of Contents

1. User Model
2. Saved Recipes Subcollection
3. Followers and Following Subcollections
4. Recipe Model
5. Recipe Subcollections
   * Comments
   * Ratings
   * Likes
   * Enhancements
6. Security Rules (Examples)

## 1️⃣ User Model

**Collection Path:**

users/{uid}

(uid is the Firebase Authentication UID)

**Fields:**

| Field | Type | Required | Description |
| --- | --- | --- | --- |
| uid | String | ✅ | Firebase Auth UID |
| email | String | ✅ | Email address |
| username | String | ✅ | Unique display name |
| profileImageUrl | String | ✅ | Avatar image URL |
| bio | String | ❌ | User biography |
| location | String | ❌ | City/country |
| createdAt | Timestamp | ✅ | Date created |
| lastLoginAt | Timestamp | ✅ | Date of last sign-in |
| preferences | Map | ❌ | Dietary/notification settings |
| isAdmin | Boolean | ❌ | Admin privileges |
| verified | Boolean | ❌ | Verified badge |
| isCookbookPrivate | Boolean | ❌ | If true, saved recipes are private |

**Example Document:**

{  
 "uid": "user\_123",  
 "email": "user@example.com",  
 "username": "foodie101",  
 "profileImageUrl": "https://example.com/avatar.jpg",  
 "bio": "Passionate about healthy cooking.",  
 "location": "Toronto, Canada",  
 "createdAt": "2025-07-14T12:00:00Z",  
 "lastLoginAt": "2025-07-14T19:00:00Z",  
 "preferences": {  
 "dietary": "vegetarian",  
 "notifications": true  
 },  
 "isAdmin": false,  
 "verified": false,  
 "isCookbookPrivate": false  
}

## 2️⃣ Saved Recipes Subcollection

**Path:**

users/{uid}/savedRecipes/{recipeId}

**Fields:**

| Field | Type | Required | Description |
| --- | --- | --- | --- |
| recipeId | String | ✅ | Reference to the saved recipe |
| savedAt | Timestamp | ✅ | When saved |
| notes | String | ❌ | User notes about the recipe |
| tags | Array | ❌ | User-defined tags (e.g., “Favorites”) |

**Example Document:**

{  
 "recipeId": "recipe\_456",  
 "savedAt": "2025-07-14T20:00:00Z",  
 "notes": "Try with almond milk.",  
 "tags": ["Favorites", "Breakfast"]  
}

## 3️⃣ Followers and Following Subcollections

**Followers Path:**

users/{uid}/followers/{followerUid}

**Following Path:**

users/{uid}/following/{followingUid}

**Fields:**

| Field | Type | Description |
| --- | --- | --- |
| followedAt | Timestamp | When the follow happened |

**Example Document:**

{  
 "followedAt": "2025-07-14T18:30:00Z"  
}

## 4️⃣ Recipe Model

**Collection Path:**

recipes/{recipeId}

**Fields:**

| Field | Type | Required | Description |
| --- | --- | --- | --- |
| id | String | ✅ | Recipe ID |
| authorId | String | ✅ | UID of creator |
| title | String | ✅ | Recipe title |
| description | String | ❌ | Short description |
| ingredients | Array | ✅ | List of ingredients with quantities |
| instructions | Array | ✅ | Preparation steps |
| coverImageUrl | String | ❌ | Main image URL |
| additionalImages | Array | ❌ | Extra images |
| cuisine | String | ❌ | Cuisine type |
| tags | Array | ❌ | Tags/keywords |
| nutritionInfo | Map | ❌ | Nutrition data (optional, AI-generated) |
| cookingTime | Number | ❌ | Minutes |
| difficulty | String | ❌ | Difficulty level |
| servings | Number | ❌ | Servings count |
| isPublic | Boolean | ✅ | Visibility flag |
| likesCount | Number | ✅ | Likes counter |
| commentsCount | Number | ✅ | Comments counter |
| rating | Map | ❌ | Aggregate rating {average, count} |
| createdAt | Timestamp | ✅ | Creation timestamp |
| updatedAt | Timestamp | ✅ | Last update timestamp |

**Example Document:**

{  
 "id": "recipe\_456",  
 "authorId": "user\_123",  
 "title": "Classic Pancakes",  
 "description": "Fluffy pancakes perfect for breakfast.",  
 "ingredients": [  
 {"name": "Flour", "quantity": "2 cups"},  
 {"name": "Milk", "quantity": "1.5 cups"},  
 {"name": "Eggs", "quantity": "2"}  
 ],  
 "instructions": [  
 "Mix dry ingredients.",  
 "Add wet ingredients.",  
 "Cook on griddle."  
 ],  
 "coverImageUrl": "https://example.com/pancakes.jpg",  
 "additionalImages": [],  
 "cuisine": "American",  
 "tags": ["breakfast", "easy"],  
 "nutritionInfo": {  
 "calories": 300,  
 "protein": "8g",  
 "carbs": "40g",  
 "fat": "10g"  
 },  
 "cookingTime": 20,  
 "difficulty": "Easy",  
 "servings": 4,  
 "isPublic": true,  
 "likesCount": 15,  
 "commentsCount": 3,  
 "rating": {  
 "average": 4.5,  
 "count": 10  
 },  
 "createdAt": "2025-07-14T15:00:00Z",  
 "updatedAt": "2025-07-14T15:00:00Z"  
}

## 5️⃣ Recipe Subcollections

### 5.1 Comments

**Path:**

recipes/{recipeId}/comments/{commentId}

**Fields:**

| Field | Type | Description |
| --- | --- | --- |
| id | String | Comment ID |
| authorId | String | UID of commenter |
| text | String | Comment text |
| createdAt | Timestamp | Date posted |
| likesCount | Number | Cached likes count |
| parentCommentId | String | For nested replies |
| isDeleted | Boolean | Soft delete for moderation |

**Example Document:**

{  
 "id": "comment\_123",  
 "authorId": "user\_456",  
 "text": "I love this recipe!",  
 "createdAt": "2025-07-14T21:00:00Z",  
 "likesCount": 2,  
 "parentCommentId": null,  
 "isDeleted": false  
}

**Comment Likes Subcollection Path:**

recipes/{recipeId}/comments/{commentId}/likes/{userId}

| Field | Type | Description |
| --- | --- | --- |
| likedAt | Timestamp | When liked |

### 5.2 Ratings

**Path:**

recipes/{recipeId}/ratings/{userId}

| Field | Type | Description |
| --- | --- | --- |
| rating | Number | 1–5 stars |
| createdAt | Timestamp | When rated |

### 5.3 Likes

**Path:**

recipes/{recipeId}/likes/{userId}

| Field | Type | Description |
| --- | --- | --- |
| likedAt | Timestamp | When liked |

### 5.4 Enhancements

**Path:**

recipes/{recipeId}/enhancements/{enhancementId}

**Fields:**

| Field | Type | Description |
| --- | --- | --- |
| id | String | Enhancement ID |
| recipeId | String | Parent recipe ID |
| createdBy | String | UID of the user |
| enhancementType | String | e.g., MAKE\_HEALTHIER |
| title | String | Title of enhancement |
| description | String | Description of changes |
| ingredients | Array | Updated ingredients |
| instructions | Array | Updated steps |
| nutritionInfo | Map | Updated nutrition info |
| coverImageUrl | String | Optional image |
| cuisine | String | Updated cuisine |
| createdAt | Timestamp | When generated |
| notes | String | User notes |

**Example Document:**

{  
 "id": "enhancement\_123",  
 "recipeId": "recipe\_456",  
 "createdBy": "user\_789",  
 "enhancementType": "MAKE\_HEALTHIER",  
 "title": "Classic Pancakes (Healthier)",  
 "description": "Reduced sugar and added whole wheat flour.",  
 "ingredients": [...],  
 "instructions": [...],  
 "nutritionInfo": {...},  
 "coverImageUrl": "https://example.com/healthier-pancakes.jpg",  
 "cuisine": "American",  
 "createdAt": "2025-07-14T22:00:00Z",  
 "notes": "Wanted to lower calories."  
}

## 6️⃣ Security Rules (Examples)

**User Profile & Saved Recipes:**

match /users/{userId} {  
 allow read: if true;  
 allow write: if request.auth.uid == userId;  
  
 match /savedRecipes/{recipeId} {  
 allow read: if  
 !get(/databases/$(database)/documents/users/$(userId)).data.isCookbookPrivate ||  
 request.auth.uid == userId;  
 allow write: if request.auth.uid == userId;  
 }  
  
 match /followers/{followerId} {  
 allow read, write: if request.auth.uid == userId;  
 }  
  
 match /following/{followingId} {  
 allow read, write: if request.auth.uid == userId;  
 }  
}

**Recipes & Subcollections:**

match /recipes/{recipeId} {  
 allow read: if resource.data.isPublic == true ||  
 request.auth.uid == resource.data.authorId;  
  
 allow write: if request.auth.uid == resource.data.authorId;  
  
 match /comments/{commentId} {  
 allow read: if true;  
 allow write: if request.auth.uid != null;  
  
 match /likes/{userId} {  
 allow read: if true;  
 allow write: if request.auth.uid == userId;  
 }  
 }  
  
 match /ratings/{userId} {  
 allow read: if true;  
 allow write: if request.auth.uid == userId;  
 }  
  
 match /likes/{userId} {  
 allow read: if true;  
 allow write: if request.auth.uid == userId;  
 }  
  
 match /enhancements/{enhancementId} {  
 allow read: if true;  
 allow write: if request.auth.uid != null;  
 }  
}