# Cooking Companion Domain Model



**RecipeInfo**
- + instructions: [InstructionStage]
- + ingredients: [(quantity: string, description: string)]
- + cookware: [string]
- + abstract: string
- + cooktime: string
- + servings: int

**Instruction Stage**
- + stageName: string
- + stepNames: [string]
- + steps: [string]

**Tags**
- + ingredientTags: [string]
- + dietaryTags: [string]
- + cookwareTags: [string]
- + cuisineTags: [string]
- + collectionTags: [string]

**Recipe**
- + AuthorUID: int
- + summary: Summary
- + tags: Tags
- + publicity: boolean
- + recipeInfo: RecipeInfo

**Summary**
- + title: string
- + rating: int
- + imgURL: str
- + recipeID: int
- + getRecipe(): Recipe

**Account**
- + firstName: string
- + lastName: string
- + bio: string
- + login: Authorization

**User**
- + username: string
- + UID: int
- + account: Account

**Search**
- + searchTags: Tags
- + searchTerms: string
- + searchQuery: string
- + buildQuery(): null
- + getResutls(): RecipeList

**Authorization**
- + email: string
- + salt: string
- + hash: string
- + loggedIn: boolean
- + authorize(password: string): boolean

**UserCollection**
- + userID: int
- + publicity: boolean

**RecipeList**
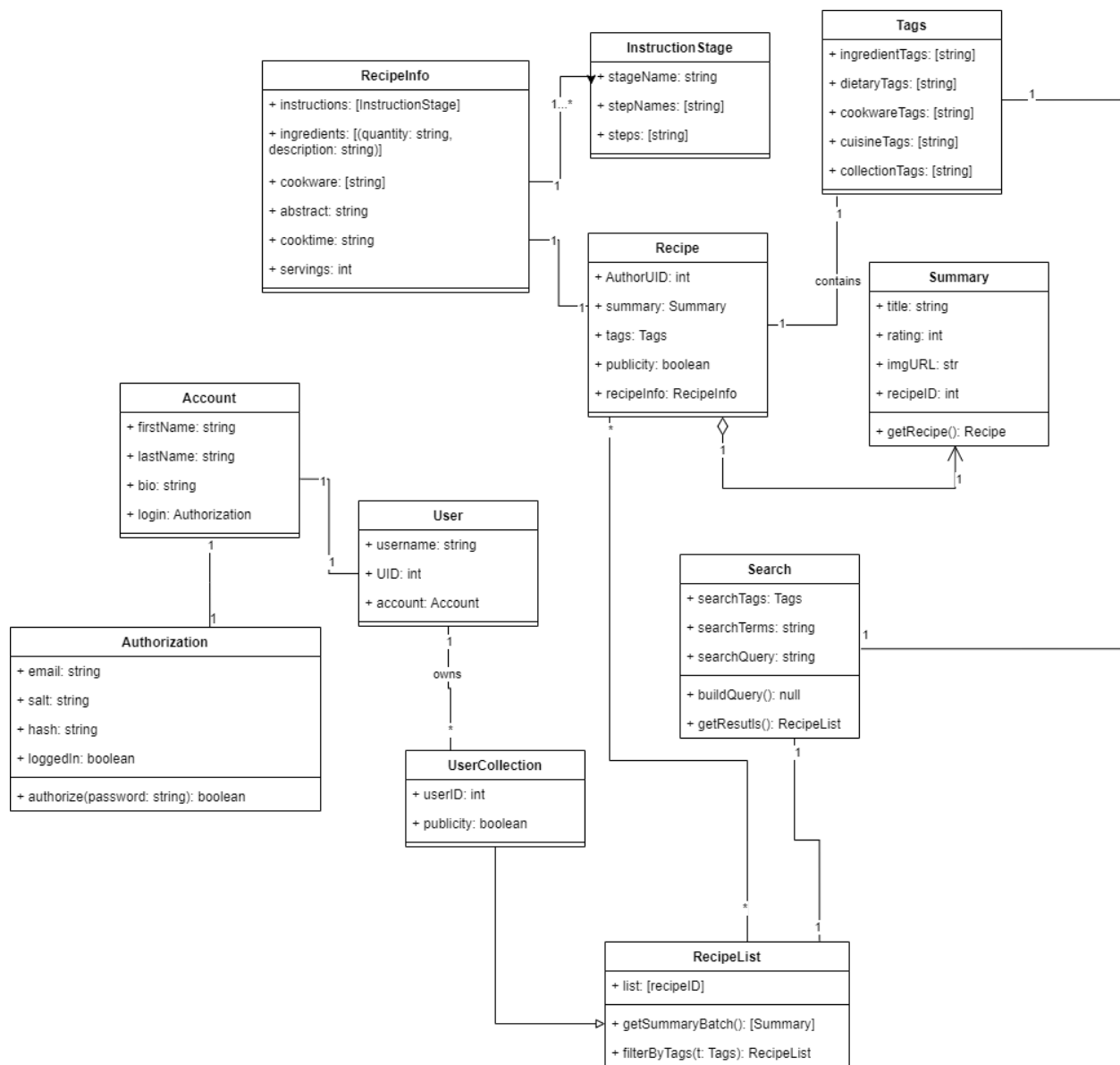- + list: [recipeID]
- + getSummaryBatch(): [Summary]
- + filterByTags(t: Tags): RecipeList

The above diagram models the relationship between classes of the Cooking Companion's Account Authentication, Recipe Storage, and Search Engine systems.

User, Account, and Authorization classes are designed to manage user-related attributes, account information, and authentication processes.

Recipe, Summary, RecipeInfo, and InstructionStage classes organize and store recipe information, covering elements such as the recipe's basic structure, summary details, procedure and ingredient information, and individual instruction stages.

Tags, Search, RecipeList, and UserCollection classes facilitate the ability for users to organize, search, and filter recipes, while also allowing them to create and manage their personalized recipe collections.

**User/Account/Authorization:**

*User:*  This encapsulates all attributes of a user and their account.  It includes the user's username, a link to their account information, and has a userID so that they can own recipes in the database and own UserCollections to save recipes. The class represents a given person using the application.

*Account:*  This is information that could be found on a profile page and connections to the authorization class, but isn't used to identify the user when making or saving recipes or UserCollections. The user information it contains is the User's first name, last name, and bio.

*Authorization:*  This class contains all information to authenticate and log in a user.  It stores their email which is checked before the password is authenticated.  Passwords are salted and then hashed so that plain text passwords are not stored and so that an attacker cannot lookup precomputed hashes for common passwords.

**Recipe/Summary/RecipeInfo/InstructionStage:**

*Recipe:* The Recipe class contains the basic structural information for a recipe that relates to its location and how it will be identified by the system, such as the author's ID and its publicity. It additionally has several objects with additional information about the recipe, which are one Summary object, one RecipeInfo object, and one Tags object. This is the main object that will point to all other info about any given recipe input into the system, and is what will be called on to be displayed on a single-recipe-view page.

*Summary:* The Summary class contains a summary of important details about a Recipe, including its title, its rating, the URL to any images associated with the recipe, and the recipe ID. It has an aggregation relationship with the Recipe class. It has a function getRecipe() which can find its associated recipe. The Summary is what will be displayed to the user when viewing search results and collections.

*RecipeInfo:* The RecipeInfo class contains the details of the procedures and ingredients of a recipe, essentially any information that would be found in an old-fashioned cookbook. It contains a list of cookware, a list of ingredients, an abstract, cooktime, and number of servings. It also contains a list of InstructionStage objects, one for each stage of the cooking process. The RecipeInfo is the majority of content that is user-driven in the application.

*InstructionStage:* The InstructionStage class is the details for a given stage of the recipe instructions. It includes the name of the stage, a list of step names, and then another list of the description of each step. The InstructionStage is what will be called to display all the instructions of a recipe, and also when displaying a step-by-step view of the recipe.

**Tags/Search/RecipeList/UserCollection:**

*Tags:* The Tags class contains multiple simple descriptions of a recipe and is included in each recipe instance. The descriptions are organized into different categories, namely ingredient tags, dietary tags, cookware tags, cuisine tags, and collection tags, each with a list of the tags in the recipe that are in those categories. A Tags object can be used by the User to filter and refine a search.

*Search:* This class controls searching functionality. It takes in search terms that the user entered into the search bar, along with the filters they selected to generate the search query, and then calls the database to execute the query. The database returns a RecipeList of all recipes that matched the query.

*RecipeList:* The RecipeList class is a compilation of a set of recipes. It is a list of recipe IDs, which are unique to each recipe. It has a method that can filter itself by tags, which takes in a Tags object and returns a subset of the original RecipeList in a new RecipeList object, where each recipe matches the new tags. It has one subclass, which is UserCollection.

*UserCollection:* This is a user's saved list of recipes. It is a collection where they can favorite and save recipes. It contains the userID of the user who created it and whether it is public or private. It is also a subclass of the RecipeList superclass.