

# **Daily Recipe DB Model & DB Design Report**

CS157A Introduction to Database Management Systems

San Jose State University

Team 5

Jiaxiang Guo

Soyeon Wang

Gricelda Tecun

## **Table of Contents**

<b>1. Functional Requirements</b>	<b>3</b>
Functions	3
User Flow Diagram	5
<b>2. Entity Relationship Diagram</b>	<b>6</b>
<b>3. Relational Schemas</b>	<b>7</b>
Entity-set schemas	7
Relationship-set schema	7
<b>4. Screenshot of Tables</b>	<b>9</b>

# 1. Functional Requirements

## Functions

### Login in / Sign Up:

- Users will be required to sign up for their own accounts to get access to the website.
- Users will be able to login to their own account to visit the website.

### Search for Recipes:

- Users will be able to search for a recipe using the title as a criterion. The search functionality will provide the users with a shortcut to find their targeted recipes if posted before by the users.

### View Profile:

- Users will be able to view their profile that their information of username, age, and country are displayed.

### Edit Profile:

- Users will be able to edit their profile, such as editing the username and other information. Also, a changing password option is provided in this function.

### View Recipe:

- Users will be able to see the recipes posted by other users. In other words, the user will be able to access the Recipe Forum, where other user's recipes are posted and ordered by date of creation.

### Create Recipe:

- User can create a new recipe with a list of ingredients, time of preparation, photos, personal comments about your new recipe, process of preparation,
- User can create their own recipe with the elements they want to add, but the most important will be the name of the recipe, the ingredients, and process of preparation

### Subscribe / Unsubscribe:

- Users will be able to subscribe another user who uploaded their favorite recipe, so they can follow up on his/her newest updated recipe. Also, an unsubscribe option is provided for users to unfollow whom they are no more interested in.

#### Report an inaccurate recipe:

- Users will be able to report inappropriate recipes that have cross contamination or that are harmful for human consumption. We need to make humans aware of what they are eating because there are many animals, plants, and other ingredients that are not suitable for intake.
- If some users keep reporting without proper reasons, users would be warned and cannot use the site for a certain period of time.

#### Bookmark Recipe:

- Users will be able to add the recipes they liked to a bookmark list.
- Users will be able to have a list of bookmarked recipes, so they can just look at their list quickly.
- Users will be able to delete a recipe from their bookmark list.

#### View My Favorite Recipes:

- Users will be able to access the recipes that they added in their favorite list.

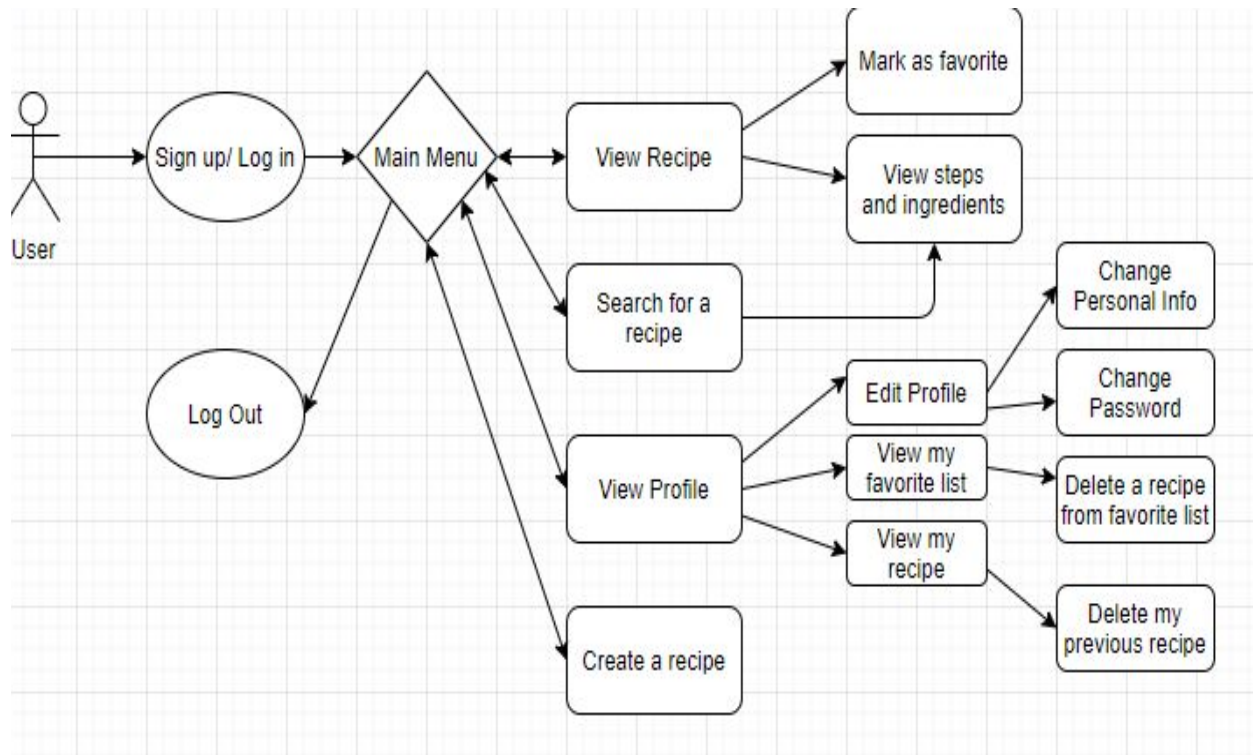
#### Delete Recipes:

- Users will be able to delete their own recipes that they already posted.

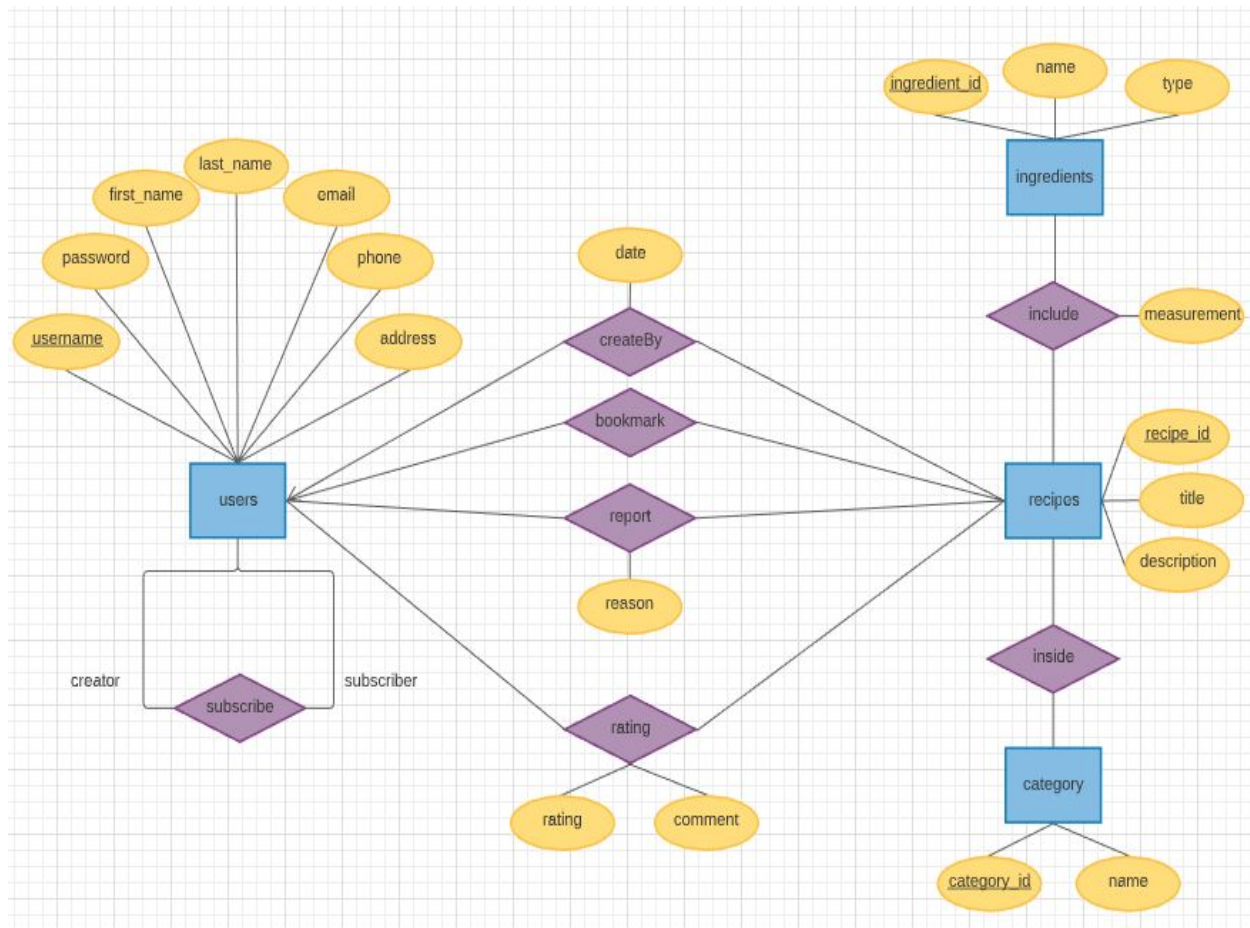
#### Share the link:

- Users can share the link of their recipes on social media such as Facebook, Instagram, etc.

## User Flow Diagram



## 2. Entity Relationship Diagram



### 3. Relational Schemas

- **Entity-set schemas**

**users(username, password, last\_name, first\_name, email, address, phone)**

- User table contains every user login information.

**recipes(recipe\_id, title, description)**

- Recipes table contains every recipe information created by the user.

**ingredients(ingredient\_id, name, type)**

- Ingredients table contains every ingredient that a recipe includes.

**category(category\_id, name)**

- Category table contains types where a recipe belongs to.

- **Relationship-set schema**

**subscribe(creator\_username, subscriber\_username)**

- Subscribe table contains user's favorite recipe uploader which is a user as well. Self-relationship with users table.

**createdby(username, recipe\_id, date)**

- Createdby table contains a relationship between users table and recipes table that shows who is the creator of the recipe.

**bookmarks(username, recipe\_id)**

- Bookmarks table contains a user's favorite recipe, it shows the relationship with users table and recipes table.

**report(username, recipe\_id, reason)**

- Report table contains a reported recipe and the reason why it is reported. it is the relational table between uses table and recipes table.

**include(recipe\_id, ingredient\_id, measurement)**

- Include table contains all ingredients that are needed for a recipe. It is the relational table between recipes table and ingredient table.

**rating(username, recipe\_id, rating, comment)**

- Rating table contains all rating scores from users for recipes. It is the relational table between recipes table and users table.

**inside(recipe\_id, category\_id)**

- Inside table contains all the recipe's categories. It shows where a recipe should be in. It is the relational table between recipes table and category table.



## 4. Screenshot of Tables

- users(username, password, last\_name, first\_name, email, address, phone)

The screenshot shows a database management interface with a left-hand sidebar displaying a schema tree. The 'recipe' schema is selected, and the 'users' table is highlighted. The main window displays the SQL query `SELECT * FROM recipe.users;` and the resulting data grid. The data grid shows 15 rows of sample data and a final row of NULL values. The columns are: username, password, last\_name, first\_name, email, address, and phone.

username	password	last_name	first_name	email	address	phone
bbbbbbbbb	zxcfqkh	Wu	Butler	898czxc@gmail.com	2131 54th ave	4159886532
ddddddddd	qasdw	Lee	Yu	12sad@gmail.com	1241 42th ave	4159695486
ggggggggg	zxcfqkh	Wu	Jamie	12zcwera@gmail.com	6591 54th ave	4159832156
izeduvek	urmd3v2nb	Chen	Louis	8324sa@gmail.com	362 9th ave	415564874
jasonkwok	lol1234567	Guo	Jason	123@gmail.com	123 5th ave	415123456
jjjjjjjjj	zxcfqkh	Octor	Han	124sadwqexv@gmail.com	9863 74th ave	4151463857
lllllllll	zxcqsqa	Lu	Maybe	azcqwe@gmail.com	6215 66th ave	4151654896
mmmmmmmm	qweasd	Zhu	Dylan	123aszqdg@gmail.com	9874 12th ave	4159635842
okuvadav	l5mh3edff3	Liu	Michael	12412@gmail.com	431 12th ave	415533456
ooooooooo	zxcqwe	Lee	Mike	xzxcqw@gmail.com	119 3th ave	4159658741
qqqqqqqqq	asdwaz	Lee	Sin	124sadzz@gmail.com	9971 99th ave	4158965123
qweasdzxc	gfgsdfx	James	Lebron	masdhiwqe@gmail.com	125 1th ave	4159865423
udufifas	uhz0f6z24v	Wu	Kelvin	1412s12@gmail.com	623 34th ave	415564756
vbefgqw	asdqwxc	Kobe	Byrant	kobe@gmail.com	1321 1th ave	4155554121
zzzzzzzzz	wwwwwww	Angela	Baby	9213s@gmail.com	364 65th ave	4159865316
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: **users**

Columns:

- username varchar(30) PK
- password varchar(30)

- recipes(recipe\_id, title, description)

SCHEMAS

Filter objects

- jiaxiangguo
- recipe**
  - Tables
    - bookmarks
    - category
    - createdby
    - include
    - ingredients
    - inside
    - rating
    - recipes**
    - report
    - subscribe
    - users
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

Table: **recipes**

Columns:

**recipe\_id** int AI PK  
title varchar(30)

1 • SELECT \* FROM recipe.recipes;

Limit to 1000 rows

Result Grid

	recipe_id	title	description
▶	1	GRILLED BASIL CHICKEN	After washing basil and tomatoes, blot them dr...
	6	MOROCCAN CHICKEN	Wash hands with soap and water. In large skille...
	7	FRESH SALAD	Salad recipes are my favorite way to showcase ...
	8	SUPER SLAM eggs	Home style
	9	Breakfast Burrito	Traditional pupusas from Mexico
	10	Hotpot. Hotpot	Good vibes all around.
	11	Beef Tacos	Home made
	12	Shrimp Vermicelli and Garlic	Traditional pupusas from China
	13	Chinese Noodles	Best traditional Chinese foods
	14	Broccoli Salad	You won't miss the bacon in this lightened-up ta...
	15	PUPUSAS	Traditional pupusas from EL Salvador
	16	Cheesy Ground Beef Tacos	Good vibes all around.
	17	Cheesy Ground Beef Tacos	Good vibes all around.
	18	Greek Feta Dip	Need some gluten-free appetizers this holiday s...
	19	Pizza	Home made style
*	NULL	NULL	NULL

- ingredients(ingredient\_id, name, type)

The screenshot shows a database management interface with a 'SCHEMAS' panel on the left and a 'Result Grid' on the right. The 'recipe' schema is expanded, showing a list of tables including 'ingredients'. The 'ingredients' table is selected, and its structure is displayed in the 'Information' panel at the bottom left. The 'Result Grid' on the right shows the output of the SQL query 'SELECT \* FROM recipe.ingredients;', displaying 15 rows of ingredient data.

**Table: ingredients**

**Columns:**

ingredient_id	name	type
1	Eggs	chill
2	Tomato	chill
3	lettuce	chill
4	potato	ambient
5	shrimps	chill
6	Totilla	hot
7	water	ambient
8	sugar	ambient
9	salt	ambient
10	Pepper	ambient
11	celery	ambient
12	onion	chill
13	chile v...	chill
14	flour	ambient
15	nudos	ambient
NULL	NULL	NULL

- category(category\_id, name)

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays a tree view with 'recipe' expanded, showing various tables including 'category'. The 'category' table is selected. Below the tree, the 'Table: category' is shown with its columns: 'category\_id' (int AI PK) and 'name' (varchar(255)).

On the right, the SQL editor shows the query: `SELECT * FROM recipe.category;`

Below the SQL editor, the 'Result Grid' displays the query results. The grid has two columns: 'category\_id' and 'name'. The results are as follows:

category_id	name
1	sweet
2	Dinner
3	spicy
4	cold
5	ambient
6	chill
7	salty
8	dietetic
9	gluten free
10	easy to make
11	breakfast
12	dessert
13	frozen
14	fast food
15	healthy
NULL	NULL

- `subscribe(creator_username, subscriber_username)`

**SCHEMAS**

Filter objects

- cs157a
- recipe**
- sakila
- soyeon\_wang
- sys

Administration Schemas Information

**Table: subscribe**

**Columns:**

- creator\_username
- subscriber\_username

va  
pk  
va  
pk

1 • `SELECT * FROM recipe.subscribe;`

Result Grid Filter Rows:

creator_username	subscriber_username
bbbbbbbb	gggggggg
mmmmmmmm	gggggggg
bbbbbbbb	izeduvek
okuvadav	izeduvek
oooooooo	izeduvek
udufifas	jasonkwok
gggggggg	jjjjjjjj
oooooooo	jjjjjjjj
gggggggg	llllllll
mmmmmmmm	llllllll
oooooooo	llllllll
bbbbbbbb	mmmmmmmm
gggggggg	qweasdzxc
okuvadav	qweasdzxc
oooooooo	qweasdzxc
udufifas	zzzzzzzz
* NULL	NULL



- createdby(username, recipe\_id, date)

**SCHEMAS**

Filter objects

- cs157a
- recipe**
- sakila
- soyeon\_wang
- sys

Administration Schemas

Information

**Table: createdby**

**Columns:**

<u>username</u>	varchar(30) PK
<u>recipe_id</u>	int PK
date	date

1 • **SELECT \* FROM recipe.createdby;**

**Result Grid** Filter Rows:

	username	recipe_id	date
▶	bbbbbbbb	1	2020-09-30
	ddddddd	2	2020-10-10
	ggggggg	3	2020-10-15
	bbbbbbbb	4	2020-10-15
	izeduvek	5	2020-10-17
		6	2020-10-17
	ggggggg	7	2020-10-25
	izeduvek	8	2020-10-25
	mmmm...	9	2020-10-25
	bbbbbbbb	34	2020-10-28
		35	2020-10-28
	okuvadav	36	2020-10-30
	bbbbbbbb	37	2020-10-30
	uduffas	38	2020-10-30
	bbbbbbbb	39	2020-10-30
		40	2020-10-30
*	NULL	NULL	NULL

- bookmarks(username, recipe\_id)

**SCHEMAS**

Filter objects

- cs157a
- recipe**
- sakila
- soyeon\_wang
- sys

Administration Schemas

Information

**Table: bookmarks**

**Columns:**

username varchar(30) PK

recipe\_id int PK

1 • **SELECT \* FROM recipe.bookmarks;**

**Result Grid** Filter Rows: E

	username	recipe_id
▶	okuvadav	1
	oooooooo	1
	okuvadav	2
	okuvadav	3
	oooooooo	5
	qqqqqqqq	5
	qqqqqqqq	6
	qqqqqqqq	7
	zzzzzzzz	8
	zzzzzzzz	9
	zzzzzzzz	34
	zzzzzzzz	35
	izeduvek	38
	izeduvek	39
	izeduvek	40
*	NULL	NULL

- report(username, recipe\_id, reason)

**SCHEMAS**

Filter objects

- cs157a
- recipe**
- sakila
- soyeon\_wang
- sys

Administration Schemas

Information

**Table: bookmarks**

**Columns:**

username varchar(30) PK

recipe\_id int PK

1 • **SELECT \* FROM recipe.report;**

Result Grid Filter Rows: Edit:

	username	recipe_id	reason
▶	bbbbbbbbb	2	inappropriate ingredient included
	bbbbbbbbb	5	empty
	ggggggggg	2	inappropriate ingredient included
	ggggggggg	5	empty
	izeduvek	1	inappropriate word included
	jijijijij	1	inappropriate word included
	lllllllll	1	inappropriate word included
	mmmmmmmm	1	inappropriate word included
	mmmmmmmm	5	empty
	okuvadav	1	inappropriate word included
	okuvadav	5	empty
	qqqqqqqqq	2	inappropriate ingredient included
	qqqqqqqqq	5	empty
	uduffas	1	inappropriate word included
	uduffas	2	inappropriate ingredient included
*	NULL	NULL	NULL



- include(recipe\_id, ingredient\_id, measurement)

The screenshot shows a database management interface with a 'SCHEMAS' panel on the left and a 'Result Grid' on the right. The 'recipe' schema is expanded, showing various tables. The 'include' table is selected, and its structure is displayed below the schema list. The 'Result Grid' shows the data for the 'include' table, with columns 'recipe\_id', 'ingredient\_id', and 'measurement'.

**Table: include**

**Columns:**

- recipe\_id** int PK
- ingredient\_id** int PK
- measurement** varchar(30)

**Result Grid:**

	recipe_id	ingredient_id	measurement
▶	1	2	2 spoon
	6	3	5 spoon
	6	4	3 spoon
	7	5	2 spoon
	7	6	2 spoon
	7	7	1 spoon
	8	6	1 spoon
	9	8	4 spoon
	10	8	3 spoon
	11	10	1 spoon
	12	10	3 spoon
	13	11	3 spoon
	14	12	5 spoon
	15	13	3 spoon
	16	15	1 spoon
*	NULL	NULL	NULL

- rating(username, recipe\_id, rating, comment)

**SCHEMAS**

Filter objects

- jiaxiangguo
- recipe**
  - Tables
    - bookmarks
    - category
    - createdby
    - include
    - ingredients
    - inside
    - rating**
    - recipes
    - report
    - subscribe
    - users
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Administration Schemas

Information

**Table: rating**

**Columns:**

<u>username</u>	varchar(30)
<u>recipe_id</u>	int PK

Limit to 1000 rows

1 • **SELECT \* FROM recipe.rating;**

**Result Grid** | Filter Rows: | Edit: | Exp

	username	recipe_id	rating	comment
▶	bbbbbbbbb	8	5	delicious
	ddddddddd	14	1	can be improved by using
	ggggggggg	9	4	delicious
	izeduvek	16	0	did not like the flavor
	jasonkwok	18	2	too much calories
	jasonkwok	19	2	too much sweet
	jjjjjjjjj	7	5	delicious
	lllllllll	11	2	not too delicious
	mmmmmmmm	10	3	delicious
	okuvadav	18	1	can be improved by using more celery
	ooooooooo	12	4	delicious
	qqqqqqqqq	13	2	did not like the Mix
	qweasdzxc	6	4	delicious
	udufifas	17	5	I love it
	vbefgqw	1	1	too sultry
	zzzzzzzzz	15	1	did not like the flavor
*	NULL	NULL	NULL	NULL

- `inside(recipe_id, category_id)`

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. The 'recipe' schema is expanded, showing various tables and views. The 'inside' table is highlighted. Below the tree, the 'Information' pane shows details for the 'inside' table, including its columns and their data types and constraints.

**Table: inside**

**Columns:**

- category\_id int PK
- recipe\_id int PK

On the right, the SQL editor shows the query: `SELECT * FROM recipe.inside;`. Below the editor, the 'Result Grid' displays the query results. The grid has two columns: 'category\_id' and 'recipe\_id'. The results are as follows:

category_id	recipe_id
2	6
2	9
2	11
3	8
3	12
3	13
6	1
6	6
7	10
7	15
9	16
9	17
9	18
10	1
10	6
10	7
11	6
11	14