CPSC 321 Final Project

Cocktail Creator

Overview and Goals

Command line application with a test-based GUI

Inspired by Epicurious' Recipe Finder feature

Holds recipes of 30 different cocktails

Stores data about what ingredients you have on hand

- Has the capability to recommend cocktails based on what you have and what you like

4 OLTP Features and 5 OLAP Features

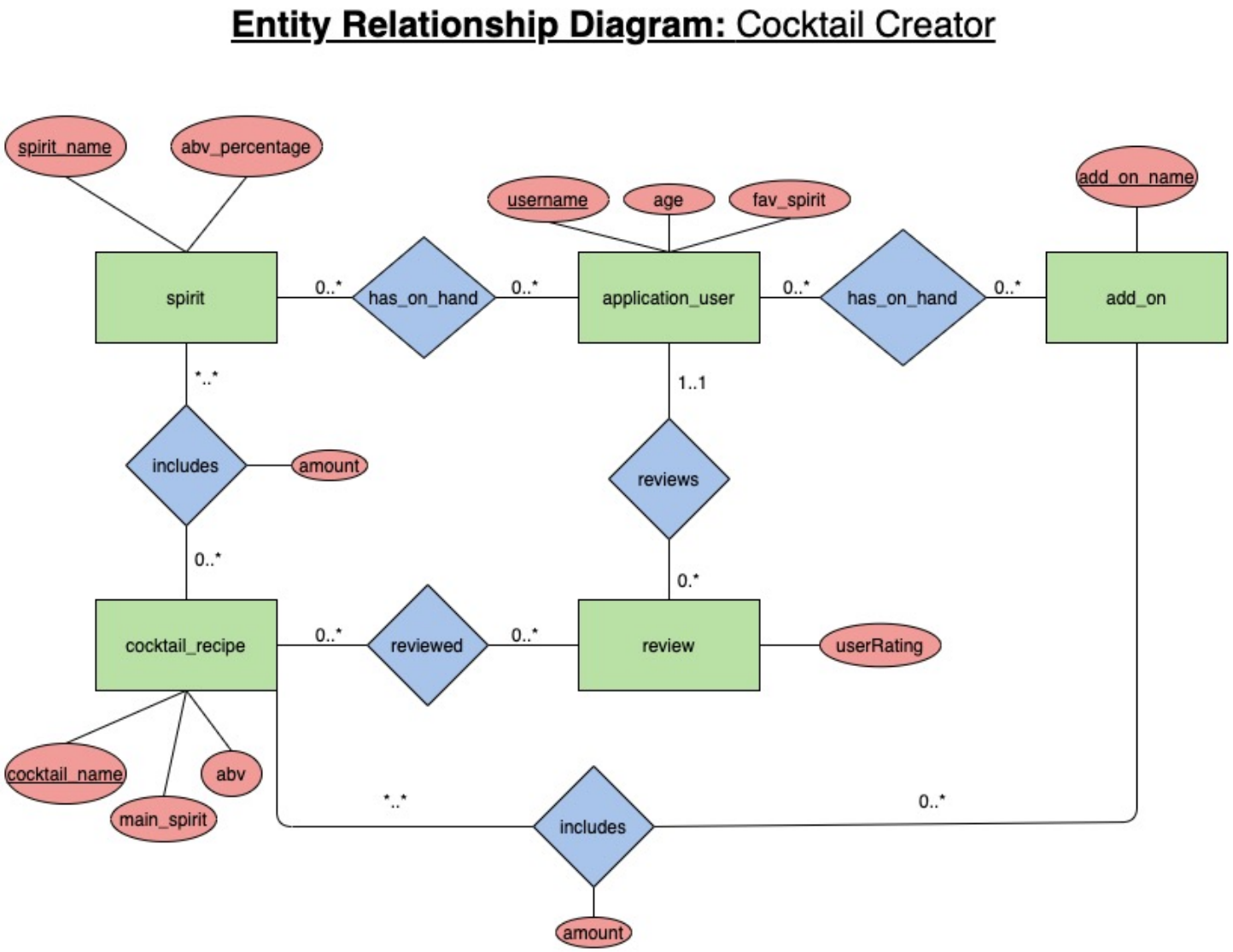
Goal of Application: Create an app that I would want to use and find valuable

Technologies Used

- Python and SQL Programming Languages
- MariaDB Database
- Dynamic SQL
  - MySQL Connector

*Further instructions regarding technology are in the README.md file on GitHub

ER Diagram


Entity Relationship Diagram: Cocktail Creator
spirit_name
abv_percentage
username
age
fav_spirit
add_on_name
spirit
0..*
has_on_hand
0..*
application_user
0..*
has_on_hand
0..*
add_on
..
1..1
includes
amount
reviews
0..*
0.*
cocktail_recipe
0..*
reviewed
0..*
review
userRating
cocktail_name
abv
main_spirit
..
includes
0..*
amount

OLTP/OLAP Queries

OLTP

- Add a cocktail review
- Query list of ingredients user has
- Edit the spirits user has
- Query a cocktail recipe

OLAP

- Find the highest rated cocktail
- Find the most complex cocktails
- Find a recipe based on the ingredients user has on hand
- Find recipes that meet filter criteria
- Find the spirits that is most often used in cocktails

OLAP: Find a recipe based on the ingredients user has on hand

```sql
SELECT a.cocktail_name, TRUNCATE(SUM(sub_sum)/c.number_of_ingredients, 2) AS percentageOfRecipe
FROM (((SELECT cocktail_name, count(*) as sub_sum
        FROM ingredient_list_spirit ils JOIN spirit_user_has suh USING (spirit)
        WHERE suh.username = 'tsdiuco'
        GROUP BY cocktail_name)
       UNION
       (SELECT cocktail_name, count(*) as sub_sum
        FROM ingredient_list_add_on ils JOIN add_on_user_has auh USING (add_on)
        WHERE auh.username = 'tsdiuco'
        GROUP BY cocktail_name)) AS a)
      JOIN
      (SELECT b.cocktail_name, SUM(b.num) AS number_of_ingredients
       FROM ((SELECT cocktail_name, count(*) AS num
              FROM ingredient_list_spirit
              GROUP BY cocktail_name)
             UNION
             (SELECT cocktail_name, count(*) AS num
              FROM ingredient_list_add_on
              GROUP BY cocktail_name)) AS b
       GROUP BY b.cocktail_name) AS c USING (cocktail_name)
GROUP BY a.cocktail_name
ORDER BY percentageOfRecipe DESC, cocktail_name
LIMIT 5;
```

OLAP: Find recipes that meet filter criteria

```sql
SELECT cr.cocktail_name, cr.abv, TRUNCATE(r.avg_rating, 2)
FROM cocktail_recipe cr LEFT OUTER JOIN
    (SELECT cocktail, AVG(userRating) AS avg_rating
     FROM review
     GROUP BY cocktail) AS r ON (cr.cocktail_name = r.cocktail)
WHERE cr.abv <= 40 AND
      cr.abv >= 20 AND
      (r.avg_rating >= 3 OR r.avg_rating IS NULL) AND
       cr.main_spirit = 'Tequila'
ORDER BY r.avg_rating DESC, cr.abv DESC;
```

OLAP: Find the most complex cocktails

```sql
SELECT a.cocktail_name, SUM(a.num) AS number_of_ingredients
FROM ((SELECT cocktail_name, count(*) AS num
        FROM ingredient_list_spirit
        GROUP BY cocktail_name)
      UNION
      (SELECT cocktail_name, count(*) AS num
        FROM ingredient_list_add_on
        GROUP BY cocktail_name)) AS a
GROUP BY a.cocktail_name
ORDER BY number_of_ingredients DESC, a.cocktail_name
LIMIT 7;
```

Live Demo