Taste Genius- Data Driven Recipe Generator

Milestone: FINAL REPORT

Group 16

Student1 Pranav Kuramkote Sudhir

Student2 Sancia Saldanha

857-465-9377 (Pranav K S)

617-352-1569 (Sancia Saldanha)

kuramkotesudhir.p@northeastern.edu

saldanha.s@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1:

Signature of Student 2:

Submission Date: December 10 2023

EXECUTIVE SUMMARY

Taste Genius - Personalized Recipe Generator App

Objective:
Taste Genius is an innovative culinary application designed to revolutionize home cooking by personalizing meal preparation. It is developed for individuals with diverse dietary preferences and health-conscious goals. The app functions as a virtual kitchen assistant, offering tailored recipe suggestions based on users' dietary restrictions, health objectives, and available kitchen ingredients.

Key Features:
1. Personalized Recipes: Taste Genius uses advanced algorithms to suggest recipes that align with users' dietary preferences, whether they are vegetarian, on a fitness journey, or exploring new culinary styles.
2. Health and Diet Conscious: The app is uniquely calibrated to understand and cater to various health goals and dietary restrictions, ensuring each recipe supports the user's lifestyle and well-being.
3. Ingredient Utilization: By considering the ingredients already available in the user's kitchen, Taste Genius minimizes waste and maximizes efficiency in meal preparation.
4. User-Friendly Interface: The app boasts an intuitive and easy-to-navigate interface, making it accessible for users of all ages and tech-savviness.
5. Culinary Exploration: It encourages culinary creativity and exploration, offering a wide range of recipes to suit any taste palette.

Implementation and Technology:
1. Advanced Algorithms: The app employs sophisticated algorithms to analyze user input regarding health goals, dietary restrictions, and available ingredients.
2. Database Integration: A comprehensive database of recipes, ingredients, and dietary information supports the app's recommendation system.
3. User Interaction Design: Emphasis is placed on a seamless user experience, with a focus on easy recipe discovery and step-by-step cooking guidance.

Impact and Benefits:
1. Enhanced Meal Planning: Users enjoy a hassle-free cooking experience with meals tailored to their specific needs and preferences.
2. Health and Wellness Promotion: The app promotes healthier eating habits by aligning meal suggestions with users' health and fitness goals.
3. Culinary Diversity: It exposes users to a wide array of culinary options, encouraging them to try new and diverse recipes.
4. Time and Cost Efficiency: By utilizing available ingredients, the app helps in reducing food waste and shopping expenses.

Future Development:
1. Community Features: Plans to introduce social sharing and community features, allowing users to share recipes and tips.
2. Integration with Smart Home Devices: Future updates may include integration with smart kitchen appliances for an even more streamlined cooking experience.
3. Expanded Recipe Database: Ongoing efforts to expand the recipe database to include global cuisines and specialty diets.

INTRODUCTION:

The culinary landscape has undergone a significant transformation in recent years, mirroring the diverse and evolving dietary preferences and health consciousness of individuals worldwide. This shift has ushered in a new era of personalized cooking experiences, necessitating innovative solutions to meet these varied needs. "Taste Genius" emerges as a revolutionary application in this context, designed to cater to the dynamic culinary preferences and lifestyle choices of individuals across the globe.

The core concept of Taste Genius revolves around providing personalized recipe suggestions and culinary guidance. This app is not just a collection of recipes; it's a smart, intuitive assistant that understands the user's dietary restrictions, health goals, and even the ingredients available in their kitchen. It's akin to having a personal chef who knows your palate and nutritional needs intimately.

The need for such a solution is evident in the growing trend of personalized nutrition and the increasing awareness around health and wellness. People are more attentive than ever to what they eat, how it affects their health, and the environmental impact of their food choices. Vegetarianism, veganism, gluten-free diets, and other specific dietary preferences are no longer niche; they're mainstream. Taste Genius taps into this zeitgeist, offering a unique service that bridges the gap between dietary needs and culinary desires.

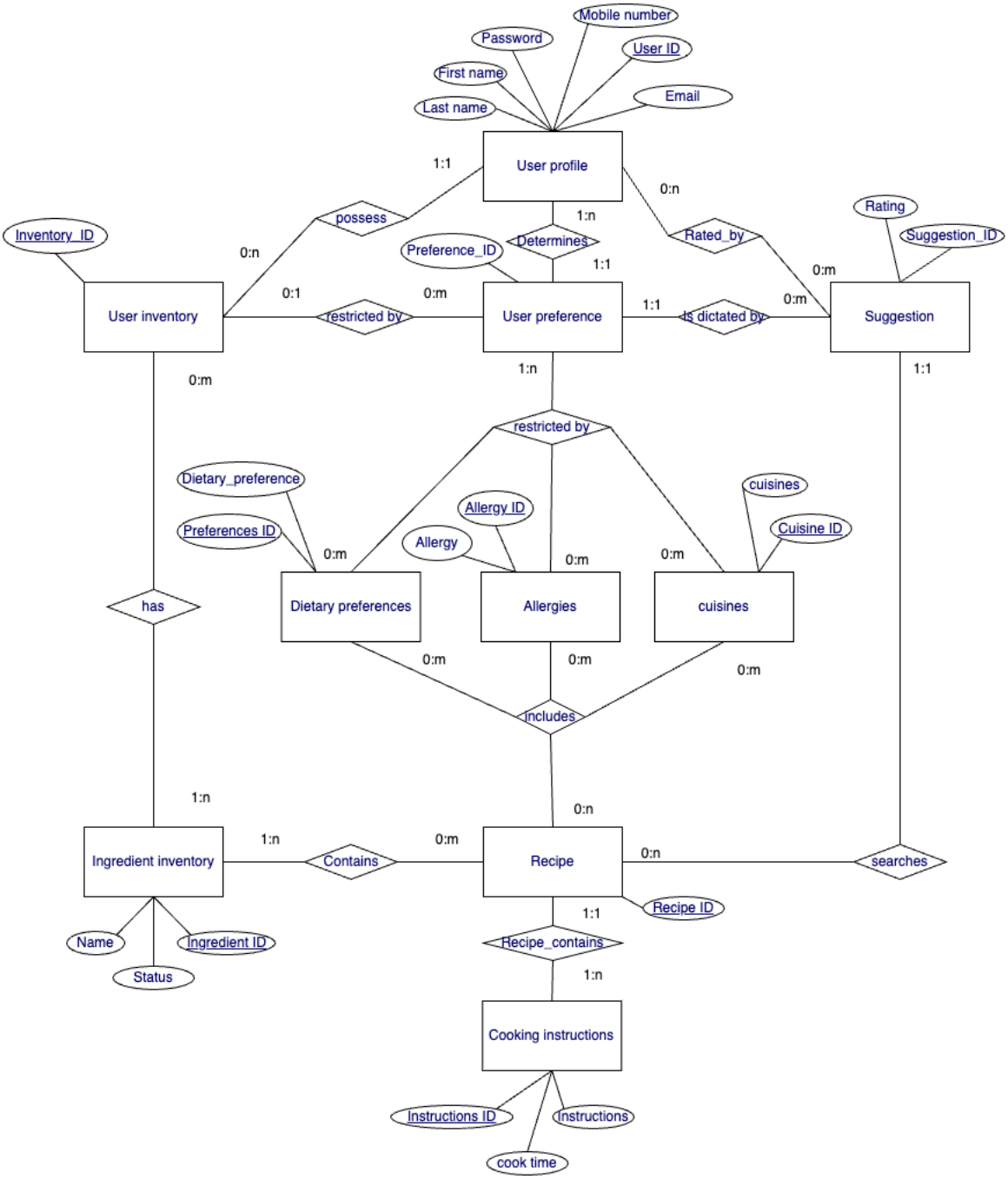
The app's functionality is straightforward yet sophisticated. Users input their dietary preferences, health objectives, and available ingredients. Taste Genius then utilizes this information to generate recipes that are not only tailored to the user’s taste but are also practical, considering what they have at hand. This approach not only promotes healthier eating habits but also reduces food waste, as it encourages the use of ingredients already in the user's pantry.

Taste Genius also addresses the challenge of meal planning and preparation in today's fast-paced world. It simplifies these tasks, making it easier for users to maintain a healthy diet and try new recipes, irrespective of their cooking skills or time constraints.

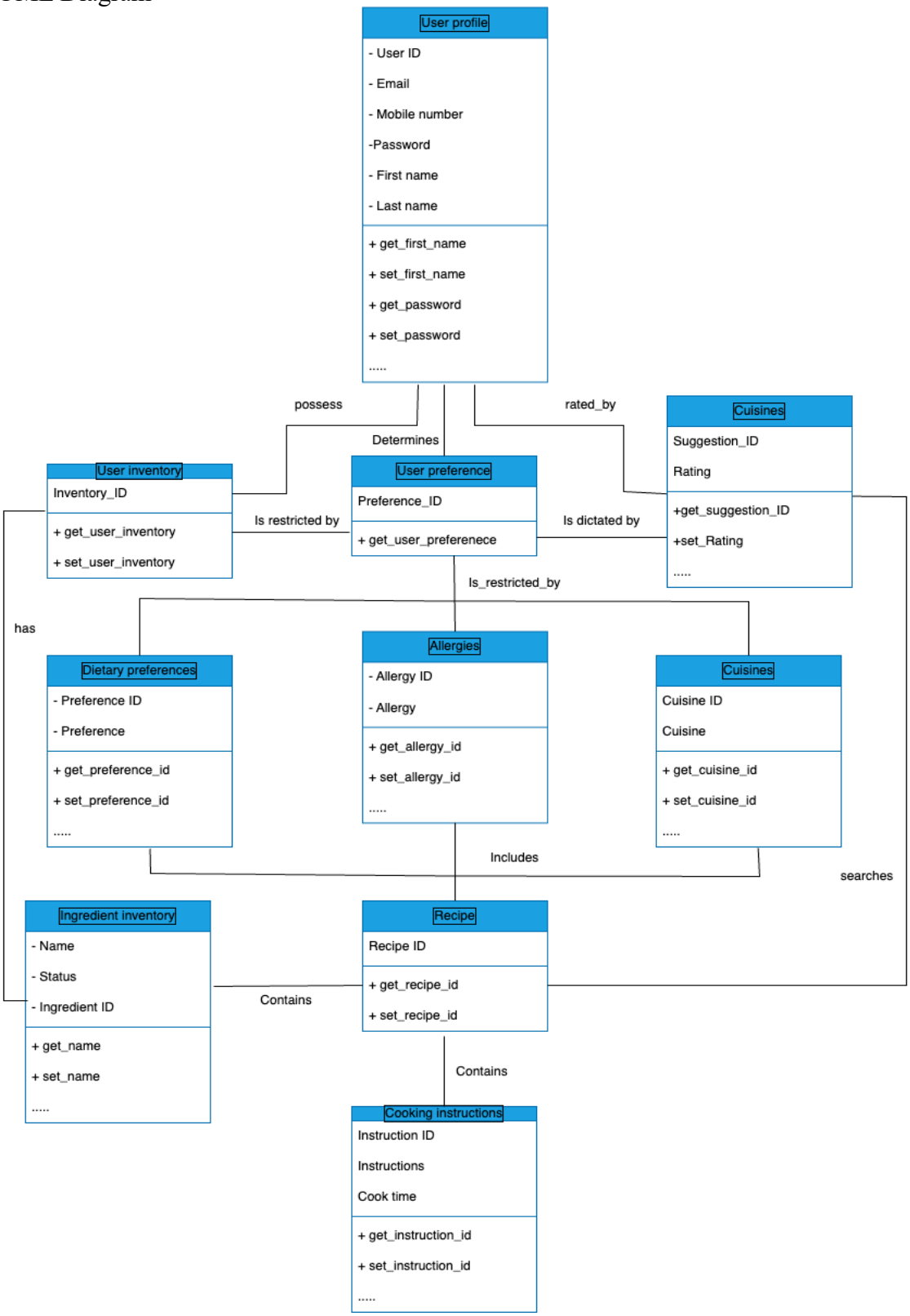
In summary, Taste Genius is more than just an app; it's a comprehensive solution for anyone looking to make their meal preparation easier, healthier, and more aligned with their personal preferences and lifestyle. It's an embodiment of the modern approach to cooking - personalized, health-conscious, and environmentally aware.

II. Conceptual Data Modeling

1. EER Diagram



2. UML Diagram



III. Mapping Conceptual Model to Relational Model

Primary Key- Underlined Foreign Key- *Italicized*

- User_profile(User_ID, First_name, Last_Name, password, Mobile_number, Email, *Preference_ID(NOT NULL)*)
- User_preference(Preference_ID, *inventory_ID*)
- Is_restricted_by_dietary_preference(*Preference_ID, Dietary_preference_ID*)
- Is_restricted_by_allergy*(Preference_ID, Allergy_ID*)
- Is_restricted_by_cuisine*(Preference_ID, Cuisine_ID*)
- Cooking_instruction(Instruction_ID, Cook_time, Instructions, *Recipe_ID(NOT NULL)*)
- Recipe(Recipe_ID,*suggestion_ID*)
- Contains(*Recipe_ID, Ingredient_ID*)
- Ingredient_inventory(Ingredient_ID, Name, Status)
- Has(*Ingredient_ID, Inventory_ID*)
- User_inventory(Inventory_ID,*User_ID*(NOT NULL))
- Dietary_preference(Dietary_preference_ID, Dietary_preference)
- Allergies(Allergy_ID, allergy)
- Cuisines(Cousine_ID, Cousine)
- Includes_Cuisine(*Recipe_ID, Cousine_ID*)
- Includes_Dietary_preference*(Recipe_ID, Dietary_preference_ID)*
- Includes_Allergy*(Recipe_ID, Allergy_ID)*
- Suggestion(Suggestion_ID, rating, *Preference_ID (NOT NULL)*)
- Rate_by(*User_ID, rating*)

User_profile Table:
Preference_ID (in User_profile) refers to Preference_ID (in User_preference).

User_preference Table:
Preference_ID (in User_preference) refers to Preference_ID (in Is_restricted_by_dietary_preference).
Preference_ID (in User_preference) refers to Preference_ID (in Is_restricted_by_allergy).
Preference_ID (in User_preference) refers to Preference_ID (in Is_restricted_by_cuisine).
Preference_ID (in User_preference) refers to Preference_ID (in Suggestion).

Is_restricted_by_dietary_preference Table:
Dietary_preference_ID (in Is_restricted_by_dietary_preference) refers to Dietary_preference_ID (in Dietary_preference).

Is_restricted_by_allergy Table:
Allergy_ID (in Is_restricted_by_allergy) refers to Allergy_ID (in Allergies).

Is_restricted_by_cuisine Table:
Cuisine_ID (in Is_restricted_by_cuisine) refers to Cousine_ID (in Cuisines).

Cooking_instruction Table:
Recipe_ID (in Cooking_instruction) refers to Recipe_ID (in Recipe).

Recipe Table:
Suggestion_ID (in Recipe) refers to Suggestion_ID (in Suggestion).

Contains Table:
Recipe_ID (in Contains) refers to Recipe_ID (in Recipe).
Ingredient_ID (in Contains) refers to Ingredient_ID (in Ingredient_inventory).

Ingredient_inventory Table:
Inventory_ID (in Has) refers to Inventory_ID (in User_inventory).

Has Table:
Ingredient_ID (in Has) refers to Ingredient_ID (in Ingredient_inventory).

User_inventory Table:
User_ID (in User_inventory) refers to User_ID (in User_profile).

Includes_Cuisine Table:
Cuisine_ID (in Includes_Cuisine) refers to Cousine_ID (in Cuisines).

Includes_Dietary_preference Table:
Dietary_preference_ID (in Includes_Dietary_preference) refers to Dietary_preference_ID (in Dietary_preference).

Includes_Allergy Table:
Allergy_ID (in Includes_Allergy) refers to Allergy_ID (in Allergies).

Suggestion Table:
Preference_ID (in Suggestion) refers to Preference_ID (in User_preference).

Rate_by Table:
User_ID (in Rate_by) refers to User_ID (in User_profile).

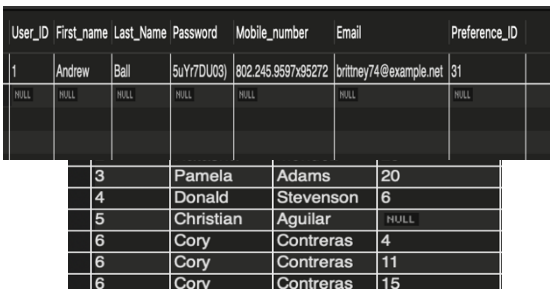
IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:
The database was created in MySQL and the following queries were performed:

1. Retrieve all columns from the User_profile table for users with a specific User_ID.

   SELECT * FROM User_profile WHERE User_ID = 1;
2. Retrieve all users and their associated inventories, including users without inventories

   SELECT User_profile.User_ID,
   User_profile.First_name,
   User_profile.Last_Name, User_inventory.Inventory_ID
   FROM User_profile
   LEFT OUTER JOIN User_inventory
   ON User_profile.User_ID = User_inventory.User_ID;



User_ID	First_name	Last_Name	Password	Mobile_number	Email	Preference_ID
1	Andrew	Ball	5uYr7DU03\	802.245.9597x95272	brittney74@example.net	31
NULL	NULL	NULL	NULL	NULL	NULL	NULL

3	Pamela	Adams	20
4	Donald	Stevenson	6
5	Christian	Aguilar	NULL
6	Cory	Contreras	4
6	Cory	Contreras	11
6	Cory	Contreras	15

3. Retrieve users who have rated a recipe higher than the average rating.

```
SELECT
   User_profile.User_ID,
   User_profile.First_name,
   User_profile.Last_Name,
   Rate_by.rating
FROM User_profile
INNER JOIN Rate_by ON User_profile.User_ID =
Rate_by.User_ID
WHERE Rate_by.rating > (
   SELECT AVG(rating) FROM Rate_by
);
```

User_ID	First_name	Last_Name	rating
2	Natasha	Mendez	4
4	Donald	Stevenson	4
5	Christian	Aguilar	5
6	Cory	Contreras	4
9	Mackenzie	Finley	5
10	Courtney	Cline	5
15	Melissa	Shields	4
18	Albert	Morrison	4
19	Wesley	Lee	4
22	Matthew	Wyatt	5

4. List users who have at least one associated dietary preference

```
SELECT User_profile.User_ID, User_profile.First_name,
User_profile.Last_Name
FROM User_profile
WHERE EXISTS (SELECT 1 FROM User_preference
WHERE User_preference.Preference_ID =
User_profile.Preference_ID);
```

	User_ID	First_name	Last_Name
	2	Natasha	Mendez
	3	Pamela	Adams
	4	Donald	Stevenson
	5	Christian	Aguilar
	6	Cory	Contreras
	7	Randall	Griffin
	8	Christine	Garcia
	9	Mackenzie	Finley
	10	Courtney	Cline
	11	Thomas	Sanders
	12	Christopher	Kennedy

5. Retrieve the count of recipes for each user.

```
SELECT User_ID, First_name, Last_Name,
      (SELECT COUNT(*) FROM Recipe WHERE
Suggestion_ID = User_profile.Preference_ID) AS
RecipeCount
FROM User_profile;
```

	User_ID	First_name	Last_Name	RecipeCount
	2	Natasha	Mendez	0
	3	Pamela	Adams	1
	4	Donald	Stevenson	1
	5	Christian	Aguilar	1
	6	Cory	Contreras	2
	7	Randall	Griffin	2
	8	Christine	Garcia	1
	9	Mackenzie	Finley	1
	10	Courtney	Cline	1
	11	Thomas	Sanders	0
	12	Christopher	Kennedy	1
	13	Samantha	Clayton	2
	14	Jackson	Patterson	1
	15	Melissa	Shields	2
	16	Ashley	Cohen	1

6. Retrieve All Ingredients from User Inventory and All Ingredients Used in Recipes:

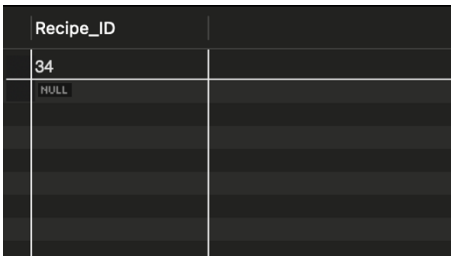
```
SELECT Name FROM Ingredient_inventory
UNION
SELECT i.Name FROM Ingredient_inventory i
JOIN Contains c ON i.Ingredient_ID = c.Ingredient_ID
JOIN Recipe r ON c.Recipe_ID = r.Recipe_ID;
```

Name
upon
church
instead
black
land
tonight
let
service
speech
especially
cup
although
similar
host

7. Find recipes that include ingredients from the user's inventory

```
SELECT Recipe.Recipe_ID
FROM Recipe
```

```
WHERE Recipe.Recipe_ID IN (
 -- Subquery to get recipes with ingredients from user's
inventory
SELECT DISTINCT r.Recipe_ID
FROM Recipe r
JOIN Contains c ON r.Recipe_ID = c.Recipe_ID
JOIN User_inventory ui ON c.Ingredient_ID =
ui.Inventory_ID
WHERE ui.User_ID = (SELECT User_ID FROM User_profile WHERE First_name = 'Donald')
);
```



NoSQL Implementation:

Two tables:

User_profile(User_ID, First_name, Last_Name, password, Mobile_number, Email, Allergy_ID, Cousine_ID, Dietary_preference_ID)

Recipe_details(Recipe_ID, Cook_time, Instructions)

One relation:

Restricted_by(Dietary_preference_ID, Allergy_ID, Cousine_ID, Recipe_ID)

have been created in Mongodb playground. The following queries were done:

1. Retrieve all users with a specific dietary preference

```
db.Users.find({ Dietary_preference_ID: 2 });
```

Result:

{ "_id" : ObjectId("65700ef980b390799552a59e"), "User_ID" : 1, "First_name" : "John", "Last_Name" : "Doe", "password" : "password123", "Mobile_number" : "1234567890", "Email" : "john.doe@example.com", "Allergy_ID" : 1, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65700ef980b390799552a5a1"), "User_ID" : 4, "First_name" : "Bob", "Last_Name" : "Williams", "password" : "bob123", "Mobile_number" : "1237894560", "Email" : "bob.w@example.com", "Allergy_ID" : 2, "Cousine_ID" : 1, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65700ef980b390799552a5a5"), "User_ID" : 8, "First_name" : "Sam", "Last_Name" : "Brown", "password" : "sam456", "Mobile_number" : "6543217890", "Email" : "sam.b@example.com", "Allergy_ID" : 2, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65700f063da578ee1c3625db"), "User_ID" : 1, "First_name" : "John", "Last_Name" : "Doe", "password" : "password123", "Mobile_number" : "1234567890", "Email" : "john.doe@example.com", "Allergy_ID" : 1, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65700f063da578ee1c3625de"), "User_ID" : 4, "First_name" : "Bob", "Last_Name" : "Williams", "password" : "bob123", "Mobile_number" : "1237894560", "Email" : "bob.w@example.com", "Allergy_ID" : 2, "Cousine_ID" : 1, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65700f063da578ee1c3625e2"), "User_ID" : 8, "First_name" : "Sam", "Last_Name" : "Brown", "password" : "sam456", "Mobile_number" : "6543217890", "Email" : "sam.b@example.com", "Allergy_ID" : 2, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65763abe10689f081c78d75d"), "User_ID" : 1, "First_name" : "John", "Last_Name" : "Doe", "password" : "password123", "Mobile_number" : "1234567890", "Email" : "john.doe@example.com", "Allergy_ID" : 1, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65763abe10689f081c78d760"), "User_ID" : 4, "First_name" : "Bob", "Last_Name" : "Williams", "password" : "bob123", "Mobile_number" : "1237894560", "Email" : "bob.w@example.com", "Allergy_ID" : 2, "Cousine_ID" : 1, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65763abe10689f081c78d764"), "User_ID" : 8, "First_name" : "Sam", "Last_Name" : "Brown", "password" : "sam456", "Mobile_number" : "6543217890", "Email" : "sam.b@example.com", "Allergy_ID" : 2, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }

2. Find users with Allergy_ID: 1, Cousine_ID: 3 and Dietary_preference_ID: 2 and have their first name starting with J or have a phone number containing 555.

```
db.Users.find({
  $and: [
    { Allergy_ID: 1 },
    { Cousine_ID: 3 },
    { Dietary_preference_ID: 2 },
    { $or: [
      { First_name: { $regex: /J/i } }, // Users with first name starting with 'J' (case-insensitive)
      { Mobile_number: { $regex: /^555/i } } // Users with mobile number starting with '555'
```

```
    ]}
  ]
});
```

Result:

{ "_id" : ObjectId("65700ef980b390799552a59e"), "User_ID" : 1, "First_name" : "John", "Last_Name" : "Doe", "password" : "password123", "Mobile_number" : "1234567890", "Email" : "john.doe@example.com", "Allergy_ID" : 1, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65700f063da578ee1c3625db"), "User_ID" : 1, "First_name" : "John", "Last_Name" : "Doe", "password" : "password123", "Mobile_number" : "1234567890", "Email" : "john.doe@example.com", "Allergy_ID" : 1, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }
{ "_id" : ObjectId("65763abe10689f081c78d75d"), "User_ID" : 1, "First_name" : "John", "Last_Name" : "Doe", "password" : "password123", "Mobile_number" : "1234567890", "Email" : "john.doe@example.com", "Allergy_ID" : 1, "Cousine_ID" : 3, "Dietary_preference_ID" : 2 }

3. Calculate the average cook time for recipes in a specific cuisine

```
db.Users.aggregate([
  {
   $lookup: {
     from: "DietaryPreferences_Allergies_Cousines",
     localField: "Dietary_preference_ID",
     foreignField: "Dietary_preference_ID",
     as: "userDetails"
   }
  },
  {
   $unwind: "$userDetails"
  },
  {
   $lookup: {
     from: "Recipes",
     localField: "userDetails.Recipe_ID",
     foreignField: "Recipe_ID",
     as: "Recipes"
   }
  },
  {
   $project: {
     _id: 0,
     User_ID: 1,
     First_name: 1,
     Last_Name: 1,
     Recipes: "$Recipes"
   }
  }
]);
```

Result:

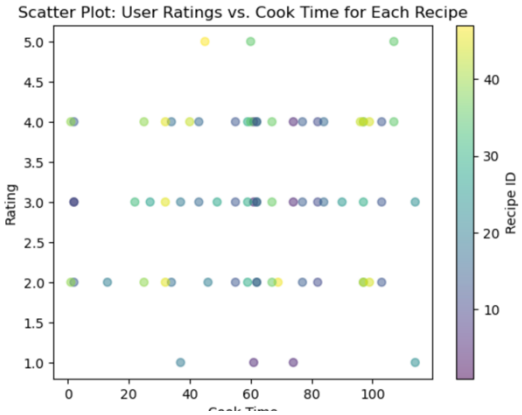
{ "User_ID" : 1, "First_name" : "John", "Last_Name" : "Doe", "Recipes" : [{ "_id" : ObjectId("65700ef980b390799552a5b3"), "Recipe_ID" : 102, "Cook_time" : "45 minutes", "Instructions" : "Saute vegetables and add spices, then simmer for 30 minutes." }, { "_id" : ObjectId("65700f063da578ee1c3625f0"), "Recipe_ID" : 102, "Cook_time" : "45 minutes", "Instructions" : "Saute vegetables and add spices, then simmer for 30 minutes." }, { "_id" : ObjectId("65700f125df714c22e74b276"), "Recipe_ID" : 102, "Cook_time" : "45 minutes", "Instructions" : "Saute vegetables and add spices, then simmer for 30 minutes." }, { "_id" : ObjectId("65700f1f482f21ee96b277f2"), "Recipe_ID" : 102, "Cook_time" : "45 minutes", "Instructions" : "Saute vegetables and add spices, then simmer for 30 minutes." }, { "_id" : ObjectId("65763abe10689f081c78d772"), "Recipe_ID" : 102, "Cook_time" : "45 minutes", "Instructions" : "Saute vegetables and add spices, then simmer for 30 minutes." }] }

{ "User_ID" : 2, "First_name" : "Jane", "Last_Name" : "Smith", "Recipes" : [{ "_id" : ObjectId("65700ef980b390799552a5bb"), "Recipe_ID" : 110, "Cook_time" : "40 minutes", "Instructions" : "Make a classic spaghetti Bolognese with ground beef and tomato sauce." }, { "_id" : ObjectId("65700f063da578ee1c3625f8"), "Recipe_ID" : 110, "Cook_time" : "40 minutes", "Instructions" : "Make a classic spaghetti Bolognese with ground beef and tomato sauce." }, { "_id" : ObjectId("65700f125df714c22e74b27e"), "Recipe_ID" : 110, "Cook_time" : "40 minutes", "Instructions" : "Make a classic spaghetti Bolognese with ground beef and tomato sauce." }, { "_id" : ObjectId("65700f1f482f21ee96b277fa"), "Recipe_ID" : 110, "Cook_time" : "40 minutes", "Instructions" : "Make a classic spaghetti Bolognese with ground beef and tomato sauce." }, { "_id" : ObjectId("65763abe10689f081c78d77a"), "Recipe_ID" : 110, "Cook_time" : "40 minutes", "Instructions" : "Make a classic spaghetti Bolognese with ground beef and tomato sauce." }] }
Type "it" for more

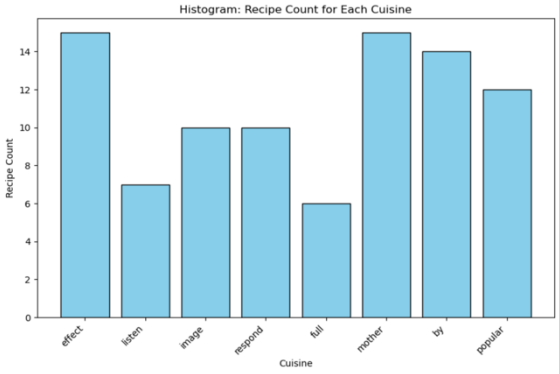
V. Database Access via Python

The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using mysql.connector, followed by cursor.excecute to run and fetchall from query, followed by converting the list into a dataframe using pandas library and using matplotlib to plot the graphs for the analytics.

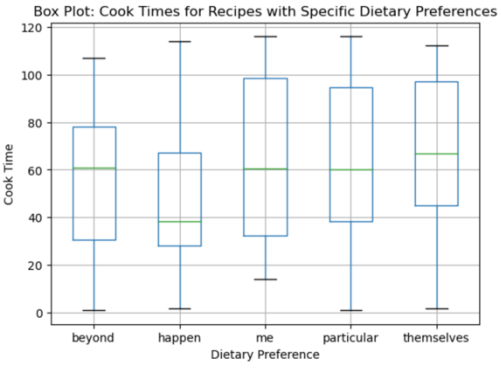
Scatter Plot - Example: User Ratings vs. Recipe ID



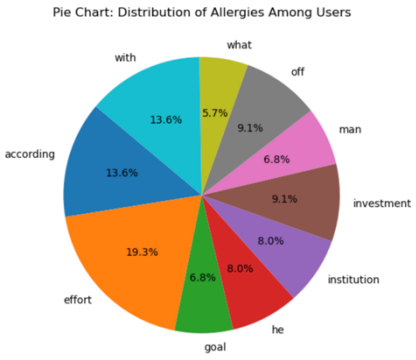
Histogram: Recipe Count for Each Cuisine



Box Plot: Cook Times for Recipes with Specific Dietary Preferences



Pie Chart: Distribution of Allergies Among User



VI:Summary:

Taste Genius is a cutting-edge application designed to revolutionize home cooking by providing personalized recipe suggestions. It caters to the diverse dietary preferences and health goals of users. The app functions as a virtual culinary assistant, tailoring recipes based on individual dietary restrictions, health objectives, and available kitchen ingredients. Its key features include:

- Personalized recipe recommendations aligning with dietary preferences such as vegetarianism or fitness-oriented diets.
- A focus on health and diet consciousness, ensuring each recipe supports the user's lifestyle and well-being.
- Optimization of ingredient usage to minimize waste and enhance meal preparation efficiency.
- An intuitive and accessible user interface.
- A wide array of culinary options to encourage exploration and diversity in cooking.

Recommendations:

1. User Engagement and Community Building: Implement features like social sharing and a community forum where users can share their culinary experiences and recipes. This fosters a sense of community and enhances user engagement.

2. Integration with Smart Kitchen Technologies: To further streamline the cooking process, integrate Taste Genius with smart kitchen appliances. This can provide users with real-time guidance and adjustments during cooking.

3. Expanding the Recipe Database: Continuously update the app's recipe database to include a wider range of global cuisines and accommodate emerging dietary trends. This will keep the app relevant and appealing to a broader user base.

4. Enhanced Personalization through AI and Machine Learning: Implement AI-driven algorithms to learn from user preferences and feedback, enabling the app to offer increasingly accurate and personalized recipe suggestions.

5. Nutritional Information and Health Tips: Include detailed nutritional information for each recipe and offer health and wellness tips. This can appeal to health-conscious users and those following specific dietary regimes.

6. Partnerships with Nutritionists and Chefs: Collaborate with professional chefs and nutritionists to create exclusive content and recipes. This can add credibility and attract users looking for expert advice.

7. Offline Functionality: Develop an offline mode that allows users to access recipes and shopping lists without an internet connection, making the app more versatile and user-friendly.

8. Multi-Language Support: To reach a global audience, offer the app in multiple languages, making it accessible to non-English speakers.

9. Marketing and Brand Partnerships: Engage in strategic marketing and partnerships with food brands, cooking appliance companies, and wellness influencers to increase visibility and user acquisition.

10. Feedback and Continuous Improvement: Regularly collect user feedback and conduct usability testing to continuously improve the app's features and user experience.

In conclusion, Taste Genius is well-positioned to become a leading app in personalized culinary experiences. By implementing these recommendations, it can further enhance its value proposition, broaden its user base, and solidify its position in the market.