Vanier College

Faculty of Science & Technology

System Development

420-436-VA

Deliverable 06

April 21st, 2023

Purple Team

Nicole Bautista

Mubeen Khan

Deven Patel

Client: Anas & Moustafa

Table of Contents

Executive Overview

The main content of this deliverable is related to the database of our information system. Specifically, the design of a relational database system, entity relationship diagram and their characteristics such as the access speed. Secondly, this document also contains the business problem and a narrative description.

As always, the business problem of our client is related to how they manage their inventory. Consequently, they have issues with keeping track of product expiration dates and overstocking items. In turn, these issues cause a waste of resources, unnecessary effort from employees and mainly a disturbance of the profitability of the Sweemory business.

The narrative description that is written in a narrative structure, explains thoroughly the three different actors which are represented in our database design. To keep it brief, we have an IT specialist, admins, and employees. The IT specialist essentially, manages the accounts of the two other actors mentioned, admins and employees. While the admin is going to be able to view and manage everything related to products, ingredients, recipe and use our messaging system. The employees will only be able to view the data mentioned previously and use the messaging system.

Then in our first appendix, we have a data dictionary that contains three different tables. The first one, represents the entities that represent our database design. We have a total of 9 entities. The second one, represents the attributes that the entities have. The third one, is the relationship table that shows the relationships between two entities.

Moving forward, our second appendix shows us the contents of our first appendix in a visual format. It is being represented as an entity relationship diagram. We have also reproduced our class

diagram from a previous deliverable. Then, explained the similarities and differences between the ERD and the newly produced class diagram.

Continuing, our third appendix we explain the how's and why's, of why we are going to optimize our queries. After that, we show an image representing our database classes normalized to 3NF. Fortunately, our ERD was already in the 3NF form. So, we explained how and why the three forms of normalization were already established.

Lastly, our fourth appendix is all about the accessing speed of our database. We found out through numerous calculation that the size of the Sweemory database would be 424.05 KB for the next few years. By finding the maximum size of each record and then an estimation of the number of records for every table. Lastly, the maximum size of a table. This would eventually give us the maximum size of the database now.

Business Problem

We have determined two key problems that demand addressing in order to answer the client's worries. First and foremost, it is difficult to keep track of product expiration dates because it takes a lot of time and effort from employees. We are putting in place an automatic system that can keep track of expiration dates and alert the owner when a product is getting close to expiration in order to resolve this problem. This approach would streamline the monitoring procedure, save time, and labour requirements, and improve precision and effectiveness.

Second, overstocking wastes resources and reduces the profitability of the store. We would provide staff with real-time stock availability updates via with the website to help them manage inventory levels and avoid overstocking as a solution to this issue. The website would also serve as a central hub for staff to obtain product data, such as costs, specifications, and locations, to increase employee knowledge and boost the effectiveness of retail operations. In order to improve communication, we have also added a messaging system for administrators and staff. Additionally, for the new employed staff, the website would provide recipes of the products available.

In conclusion, the client could decrease food waste, improve the effectiveness of the store, and boost profitability by implementing an automated system to track expiration dates, a messaging system, a list of recipes, real-time updates on stock availability, and a centralised platform for accessing product information.

Narrative Description

In the database design that team purple has created, there are in total three different actors, each one having their own role. The actors are admins, employees, and IT Specialist. The IT specialist is different compared to the two other actors, because he is the one who will give users access to the system and will decide what permissions the user will have. The IT specialist will have permission to add, edit, and delete an admin or employee. The admin and Employee will be interacting with the part of the system that has to do with the products and ingredients. The admin will have permission to add, edit and delete items. However, the employee will not be able to do such things. They will be restricted to only viewing and editing the quantity of the products and ingredients.

IT Specialist

Since this system will not be accessible to any kind of user, we will implement an IT specialist user in the database so that he can then proceed to registering new users. When the IT decides to create the new user, they must first enter the necessary information to register the user into the system and insert the information in the user table. This information would simply be the username, password, status, and the user type which can be either admin or employee. The user type will be used to differentiate the actors. Once the user has been registered, the user table will automatically be updated.

Now that the user is created, the IT specialist will be able to view their information and edit any information that require change, such as first name**,** last name**,** middle name**,** email**,** and phone number which belong to the profile table, and the username, password, and status which belong to the user table. There will also be a delete feature which will remove the record from both the user table and profile table since the user ID in in the profile table is a foreign key of the user ID in the user table.

Admin

When an admin browses the application, he will be able view, add, edit, and delete products from the product table. When creating a product, he will have to add the name, the picture, the description, the produced date, the expired date, the quantity, and the price for that product. Since the user can add the same product multiple times, but using a different produced date**,** we have made a separate table for produced date, expired date, and quantity which has a product quantity primary key and a product id as foreign key. The admin can also edit a product and delete it if he wishes to remove it from the products table.

Furthermore, just like the products, the admin can also view, add, edit, and delete ingredients from the ingredient table. Adding an ingredient requires the name, arrival date, expired date, quantity, description, picture, and price. The primary keys here are the ingredient ID, arrival date, expired date and quantity will have its own table where ingredient ID is a foreign key, and ingredient quantity ID is a primary key. Editing and deleting the ingredient will be available once it is created and inserted into the database.

The admin will also be able to interact with the recipes. He will be able to view, add, edit, and delete recipes. He will be able to add new recipes to the database by writing the title, description, and the picture. Once the record is added, he can proceed to edit the values of that record or delete.

Employee

When an employee accesses the application, he will only be able to view the products from the product table and edit the quantity from the products quantity table**.** As I previously explained, the product id in the product quantity table will be a foreign key of the product id in the product table. The same will be done for the ingredients. The employee can only view the ingredients from the ingredient table and edit the quantity from the ingredient quantity table, where the ingredient id of ingredient quantity table will be a foreign key of the ingredient id in the ingredient table. The employee can also view the recipes. However, cannot add, edit, or remove any of the records in the recipes table.

Finally, both the employee and admin will have the ability to add and delete the messages. When an employee or admin tries to send a message to another user, they will have to input into the database the receiver information, which will be a user ID, the sender information, which will be the user ID of the user sending the message, the message, and the timestamp. Messages will be sent using the email of a user. The email will then be used to find the user ID that is related to that email. If the user ID exists, it will then be placed into the database as receiver. The users will also be able to delete messages from the database. However, they will not be able to edit the messages.

Appendix 1 – Data Dictionary

Entities

Entity	Description	Aliases	Occurrence
User	A user uses the website.		6 users.
Profile	A profile is set up for the user.		6 profiles
Ingredient	An ingredient is used by employees to make products.	Items	50 ingredients
Ingredient_Quantity	An ingredient's quantity is simply the quantity of ingredients bought on a specific date.		100 ingredient quantity
Product	A product is made by employees to sell to customers.	Final Product	25 products
Product_Quantity	A product's quantity is a batch made on a specific day.		50 product quantity
Notification	A notification alerts the employee and admin.		100 notifications
Message	A message sent to an employee or admin.		100 Messages
Recipe	A recipe is being utilized by User.		25 Recipes

Attributes

Entity	Attribute Name	Description	Constraints	Data type
User	user_id	Uniquely identifies each user	Primary Key	Integer
	user_type	Type of user permissions.	Not null	Text
	username	Full name of staff and a random digit	Not null	Varchar
	password	Password of user	Not null	Varchar
Profile	user_id	Uniquely identifies each user	Primary Key, Foreign Key	Integer
	first_name	First name of user	Not null	Varchar
	middle_name	Middle name of user		Varchar
	last_name	Last name of user	Not null	Varchar
	email	Email address of user	Not null	Varchar
	phone_number	Phone number of user	Not null	Varchar
	status	Status of current user	Not null	Varchar
Ingredient	ingredient_id	Uniquely identifies each ingredient	Primary Key	Integer
	name	Full name of ingredient	Not null	Varchar
	description	Description of ingredient	Not null	Text
	price	Price of ingredient	Not null	Float
	picture	Picture of ingredient		Varchar
Ingredient_Quantity	iq_id	Uniquely identifies each batch of product	Primary Key	Integer
	ingredient_id	Uniquely identifies each ingredient	Foreign Key	Integer

	produced_date	Date of when the product was produced	Not null	Date
	expired_date	Date of when the product expires	Not null	Date
	quantity	Quantity of the product made	Not Null	Integer
Product	product_id	Uniquely identifies each product	Primary Key	Integer
	name	Full name of product	Not null	Varchar
	description	Description of product	Not null	Text
	price	Selling price of product	Not null	Float
	picture	Picture of product		Varchar
Product_Quantity	pq_id	Uniquely identifies each batch of product	Primary Key	Integer
	product_id	Uniquely identifies each product	Foreign Key	Integer
	produced_date	Date of when the product was produced	Not null	Date
	expired_date	Date of when the product expires	Not null	Date
	quantity	Quantity of the product made	Not Null	Integer
Notification	notify_id	Uniquely identifies each notification	Primary Key	Integer
	notify_type	Type of notification depending if it's low on stock or about to expire	Not null	Varchar
	message	Text	Not null	Text
	timestamp	Date and time of when the notification was sent	Not null	Timestamp
Message	message_id	Uniquely identifies each message	Primary Key	Integer
	receiver	Person receiving the message	Not null	Varchar

	sender	Person sending the message	Not null	Varchar
	message	Message written by sender	Not null	Varchar
	timestamp	Date and Time when the message is sent	Not null	Text

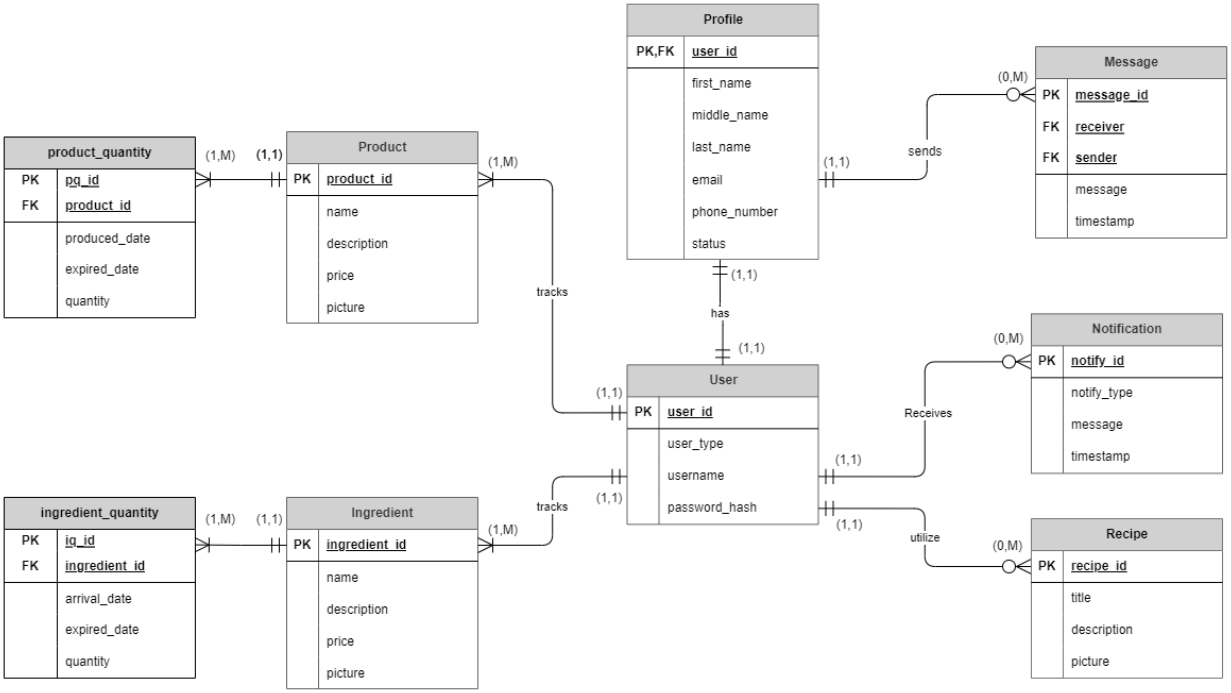
Relationship List

Entity	Multiplicity	Relationship	Multiplicity	Entity
User	1..1	has	1..1	Profile
User	1..1	receives	0..*	Notification
User	1..1	utilizes	1..1	User
User	1..1	tracks	1..*	Products
User	1..1	tracks	1..*	Ingredients
Profile	1..1	sends	0..*	Messages
Product	1…1	has	1..*	Product_Quantity
Ingredient	1..1	has	1..*	Ingredient_Quantity

Appendix 2 – ER Diagram

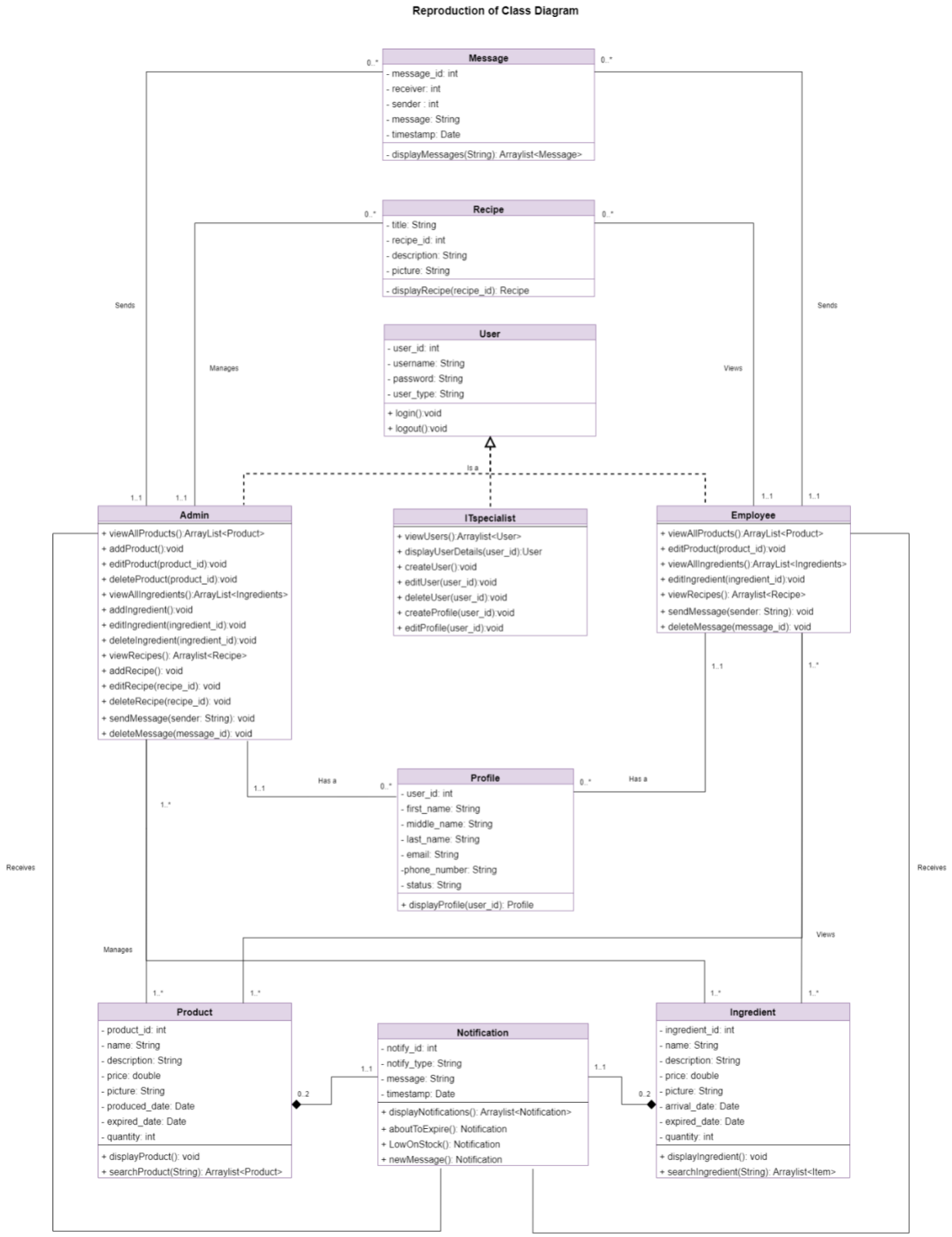
ER Diagram

https://drive.google.com/file/d/1n-kDzu3I_LnyIR4Kxl9rxB4iSA5NIu2r/view?usp=sharing



Class Diagram

https://drive.google.com/file/d/1Z0PeL5N_k-SaUnwaN5GjkBJHf1QRMUfM/view?usp=sharing



Differences Between ERD & Class Diagram

- Ingredient_quantity and ingredient entities are in one class in the UML called ingredient.
- Product_quantity and product entities are in one class in the UML called product.
- Class diagram shows behavioural features of the system.
- ERD shows structural features of the system.
- Class diagram has parent class User and subclasses for admin, employee, IT specialist.
- ERD has only user entity with user_type attribute that indicates if user is admin, employee, or IT specialist.
- Some ERD entities have same fields because they are foreign keys that are used to join entities.
- Class diagram classes don't have repeating fields because they can be inherited.

Similarities Between ERD & Class Diagram

- Both show relationships between, entities in the ERD or classes in the class diagram.
- Both entities and classes represent our system.
- Both entities and classes indicate the attributes.

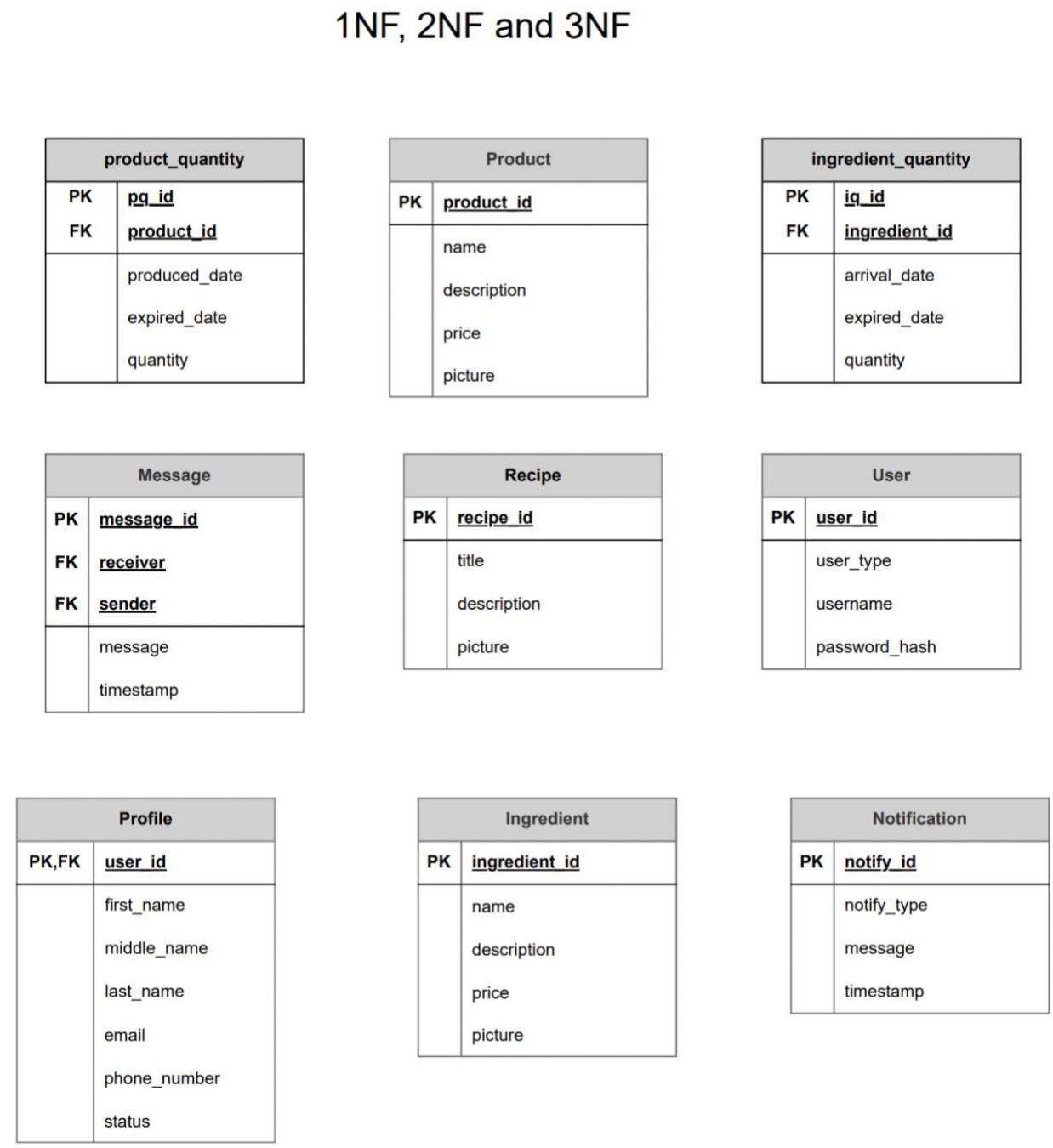
Appendix 3 – Query Optimization and Normalization

Query Optimization

Our information system will contain many queries. Therefore, using numerous different processes to optimize our queries, will consequently bring several advantages for us. Some of these advantages are the following. When queries are optimized, the resource consumption is reduced. As well as the response time of the query being decreased. Lastly, we can identify poor query performance. Now for the techniques that we will use to optimize our queries, here are the ones we plan on implementing.

- Using SELECT with column names instead of SELECT * - Avoids retrieving unnecessary data.
- Creating indexes on frequently used columns - Helps retrieve information faster/efficiently.
- Create JOINS with INNER JOIN (not WHERE). - Creates a more efficient execution plan.

Normalization



Our database's entity relationship diagram was fortunately already in 3NF form. Therefore, nothing was modified. 1NF is established because we can't have multiple values in a single column. The two tables named product_quantity and ingredient_quantity ensure that when the business adds more stock for an ingredient or make more of a particular product, a new record is added into them.

2NF is established since we don't have any partial dependencies. The two tables called ingredient_quantity and product_quantity have their respective iq_id and pq_id primary keys as well as their respective ingredient_id and product_id foreign keys. These two keys are required for the non-key attributes of these two tables to exist.

Lastly, 3NF is established because there are no transitive dependencies. This means that a non-key attribute does not depend on another, in that table.

Appendix 4 – Access Speed

For the current project, we are estimating for the website to be around 424.05 KB for the next few years. These results were bases on the following calculations.

Client Questionnaire

Q: How many ice cream flavours do they have in total?

A: We currently have around 22 to 24 ice cream at this time.

Q: How many ingredients do they use?

A: Usually, we use around 43 to 47 ingredients to make our products.

Access Speed Required

Due to our client's business being relatively new and their current staffing of only one employee, the frequency of database access is projected to be minimal. Furthermore, besides ourselves, the only individual with authority to modify the database is our client. Once the initial data entries, such as user profiles and product and ingredient information, have been added, only our client will be accessing the database, intermittently inserting new product information.

Response Time

As any good website, we are aiming for the response time to be around 100 ms to 1000 ms. We will try to do so as we optimize the website.

Optimizing Our Website

During the initial phase of creating the Entity Relationship Diagram (ERD), our team generated more tables than required. For instance, we designed a user table that extended to IT Specialist, Employee, and Admin tables. However, we later recognized that these extended tables were redundant and occupied essential space. Furthermore, the User table already contained a user type, rendering the extension tables unnecessary. Consequently, we removed the redundant tables to optimize space and enhance access speed.

Apart from optimizing the ERD, we intend to optimize our website in other ways. Firstly, we aim to limit the number of redirects to reduce loading time and ensure consistent views. Secondly, we plan to minimize the amount of HTTP requests as they contribute to the website's loading time. For example, we

will reduce the size of stylesheets, scripts, and additional CSS or JavaScript. Finally, we will avoid using large image sizes that can impede the website's performance.

Database Access

Due to our client's nascent business and their current staffing of only one employee, the frequency of database access is projected to be minimal. Furthermore, besides ourselves, the only individual with authority to modify the database is our client. Once the initial data entries, such as user profiles and product and ingredient information, have been added, only our client will be accessing the database, intermittently inserting new product information.

Maximum size of each record (in bytes)

User Table	
Field Name	**Field Size**
user_id	3
user_type	12
username	50
password	72
Record Size	**140 bytes**

Profile Table	
Field Name	**Field Size**
user_id	3
first_name	50
middle_name	50
last_name	50
email	80
phone_number	10
status	8
Record Size	**254 bytes**

Recipe Table	
Field Name	**Field Size**
recipe_id	3
title	50
message	1000
timestamp	15
Record Size	**1068 bytes**

Message Table	
Field Name	**Field Size**
message_id	3
receiver	3
sender	3
message	150
timestamp	15
Record	**174 bytes**

Notification Table	
Field Name	**Field Size**
notify_id	3
notify_type	8
message	100
timestamp	15
Record Size	**126 bytes**

Product Table	
Field Name	**Field Size**
product_id	3
name	50
description	128
price	4
picture	72
Record Size	**257 bytes**

Product_quantity Table	
Field Name	**Field Size**
pq_id	3
product_id	3
produced_date	15
expired_date	15
quantity	5
Record Size	**41 bytes**

Ingredient_quantity Table	
Field Name	**Field Size**
iq_id	3
ingredient_id	3
produced_date	15
expired_date	15
quantity	5
Record Size	**41 bytes**

Ingredient Table	
Field Name	**Field Size**
Ingredient_id	3
name	50
description	128
price	4
picture	72
Record Size	**257 bytes**

Maximum Number of Records Per Table

The maximum number of records are based on the client questionnaire and in addition we have multiplied the number by a factor of 2 to make space for margin of error and future proofing

Product Table	
Max Records	**50**

Recipe Table	
Max Records	**100**

Ingredient Table	
Max Records	**100**

Notification Table	
Max Records	**100**

User Table	
Max Records	**50**

Product_Quantity Table	
Max Records	**100**

Profile Table	
Max Records	**50**

Ingredient_Quantity Table	
Max Records	**200**

Message Table	
Max Records	**150**

Maximum Size of the Table

Maximum Table Size = (**Max Record Size**(bytes) * **Max Records**) / **1000** (kilobytes)

Product Table	
Max Records	**12.85 KB**

Profile Table	
Max Records	**12.7 KB**

Ingredient Table	
Max Records	**25.7 KB**

User Table	
Max Records	**7 KB**

Message Table	
Max Records	**26.1KB**

Recipe Table	
Max Records	**106.8 KB**

Notification Table	
Max Records	**12.6 KB**

Product_Quantity Table	
Max Records	**8.2 KB**

Ingredient_Quantity Table	
Max Records	**4.1 KB**

Maximum Size of the Database now

Seeing how the database is currently empty, we estimated the amount considering the size of database right now and the max size of the database.

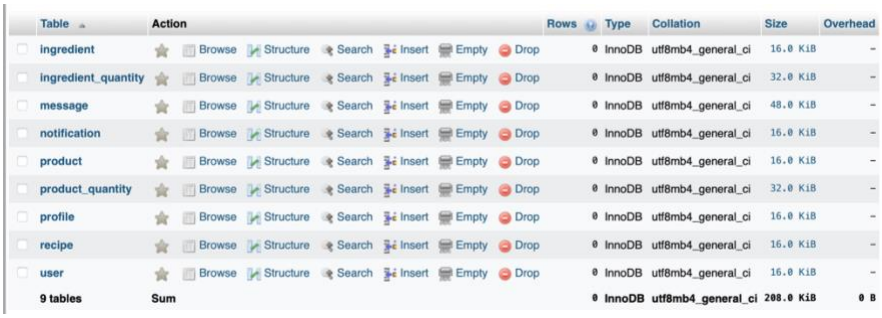


Table	Action	Rows	Type	Collation	Size	Overhead
ingredient	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
ingredient_quantity	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
message	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
notification	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
product	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
product_quantity	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
profile	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
recipe	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
9 tables	Sum	0	InnoDB	utf8mb4_general_ci	208.0 KiB	0 B

Maximum Database Size Now → Sum of All Tables (Max Table Size) + Sum of All (Empty Tables)

Maximum Database Size Now = 216.05 KB +208.0 KB

Maximum Database Size Now = 424.05 KB (or 0.42405 MB

)

Works Cited

HADŽIĆ, A. (2021, June 18). *14 Website Speed Optimization Tips: Techniques to Improve Performance and User Experience*. Retrieved from Sematext: https://sematext.com/blog/improve-website-performance/

Holowczak, R., Begg, C., & Connolly, T. M. (2008). *Business Database Systems* (Vol. 1st Edition).