# CS374 – Intro to Database Management

# Database Project

# Rubric for Project Write-up

## Group Member #1: McKenna Galle (The Great Generator)

## Group Member #2: Tersa Almaw (The Awesome Advocate)

## Group Member #3: Vina Le (The Lifeline Leader)

**TEAM G-AL-LE**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Requirements | Points | Awarded |
| Description of Application | * System requirements (e.g. hardware, DBMS, other software) * A detailed description of your application, including how each aspect of your application will satisfy your client’s needs   + There should be enough detail that others could implement your application * Are there design goals that were not met? What are they, and why won’t you fulfill them? | 15 |  |
| Class Concepts | * Discussion of concepts from class that were applied to your project * Discussion of concepts from class that did not apply to your project | 10 |  |
| Personal Contributions | * Contributions listed accurately and fairly * Team discussions that took place over design and implementation * Discussion of alternative designs and algorithms * Discussion about how/whether the project made concepts discussed in class more concrete | 15 |  |
| Grammar | * Spelling and grammar * Follow rules from the Penguin handbook on writing * References as appropriate (e.g. if you are modeling your application after an existing application, make note of that) | 10 |  |

**Description of Application**

For a system requirement, we have our database implemented in Microsoft Access and embedded SQL code using Visual Studios. The user is able to choose from a given menu on what queries can be viewed when accessed. Due to the fact that this was originally created to be out in a mobile application, Eclipse was unable to process Access’ new packet software. With the database already completed at the time, the group has decided to create the queries using C++ instead. The user has the options to view all items that is currently in their refrigerator, view specific allergy-based or quality-based items that are in the fridge. In addition to that, the user also has the option to view what is in their shopping list when the fridge only has one item left in the fridge or when a specific item is expiring soon. This is an aid that provides the user with less stress when they forget the grocery list and wants to know what items are already in the fridge. It was decided that the database, and the embedded SQL, should have categories catered to those with allergies or dietary restrictions, therefore the database includes categories users can mark (check box) whether a specific item is vegetarian/vegan-friendly, contains nuts, soy, dairy, or eggs. This is especially helpful for users who have family members or themselves that are sensitive to certain foods. The largest design goal that was not met was getting the program to implement in Eclipse and using JavaScript to embed it into a mobile application. With such a wide variety of food selection, by creating the program in an app, it allows non-local contributions from other parties like stores or family members. This will pave the way for future improvements, as well as have contributions from local supermarkets that is able to categorize and aid their shoppers in finding items easier. Eventually, the database is able to update itself without user input, especially with tags that store locations are able to dictate.

**Class Concepts Applied**

Specific concepts that was applied in this project was the to use the most efficient non-crashing code based on Complex SQL, where instead of grouping multiple queries into one schema, we have decided to expand and call in functions. However, we did not use any nested loops due to the reason our database was large even without adding on more clauses to specifically grab a certain item. Due to this being a database about a refrigerator, most entries are not just one item – since there are different varieties and types that each item can consist of.

**Personal Contributions**

McKenna: Took over the embedded SQL portion of the work and making sure that all queries are executed correctly.

Tersa: Inputting entries in the database and researching all items and locations that are inserted in the DBMS. Wrote the queries in English.

Vina: Inputting entries into the database wrote the queries and all documentations.

All team discussions are made together when it comes to design and execution. We decided together what needs to be in certain DBMS and how the DBMS categories are executed.