Ricardo Quiroga

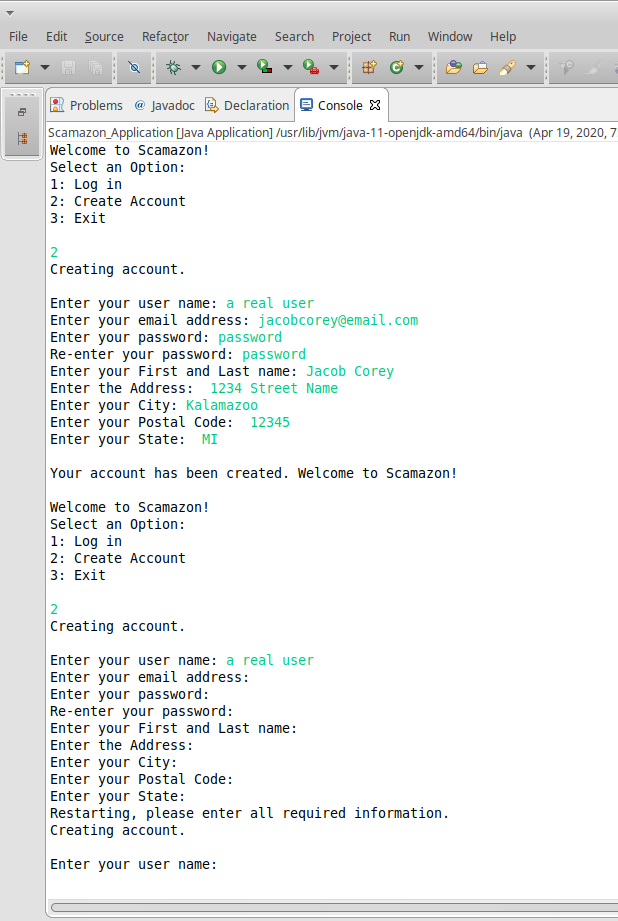
Jacob Corey

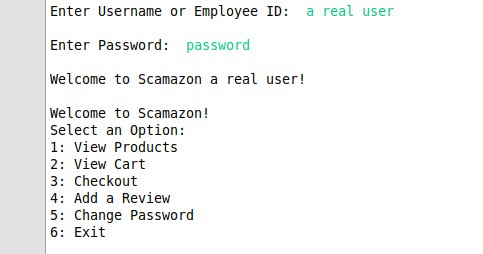
CS 4430 Final Project Report

Our project which we titled “Scamazon” is a text-based database program that resembles product selling websites such as Amazon. We decided to use Java for this, since it was the language that we were most familiar with.

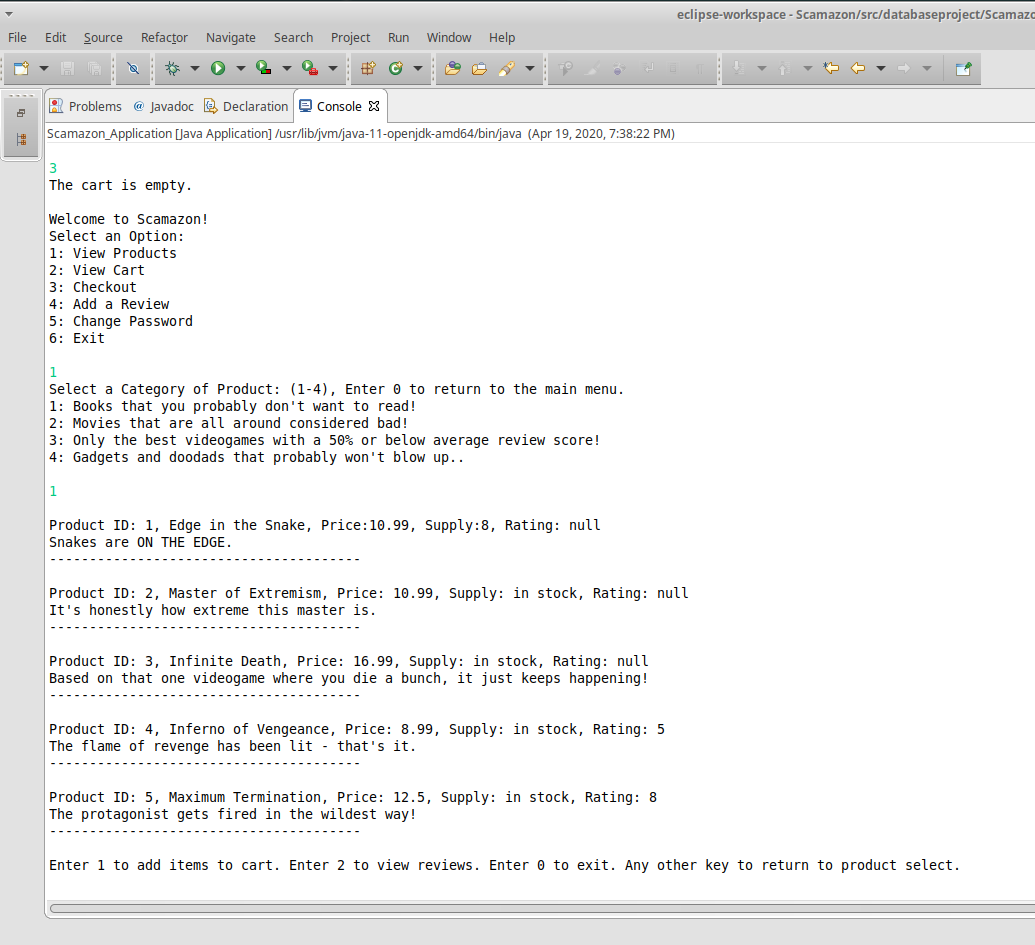
Scamazon opens with a menu allowing the user to either log in to the application or create a customer account in the database. The login method sanitized the user’s input and queries the users table to see if both the userID and password match. If they do it will create a customer class. If not, it will check to see if it is an employee id and password. The user can also choose to create an account. The program will check to make sure the passwords are the same and that the chosen userID does not already exist in the system.

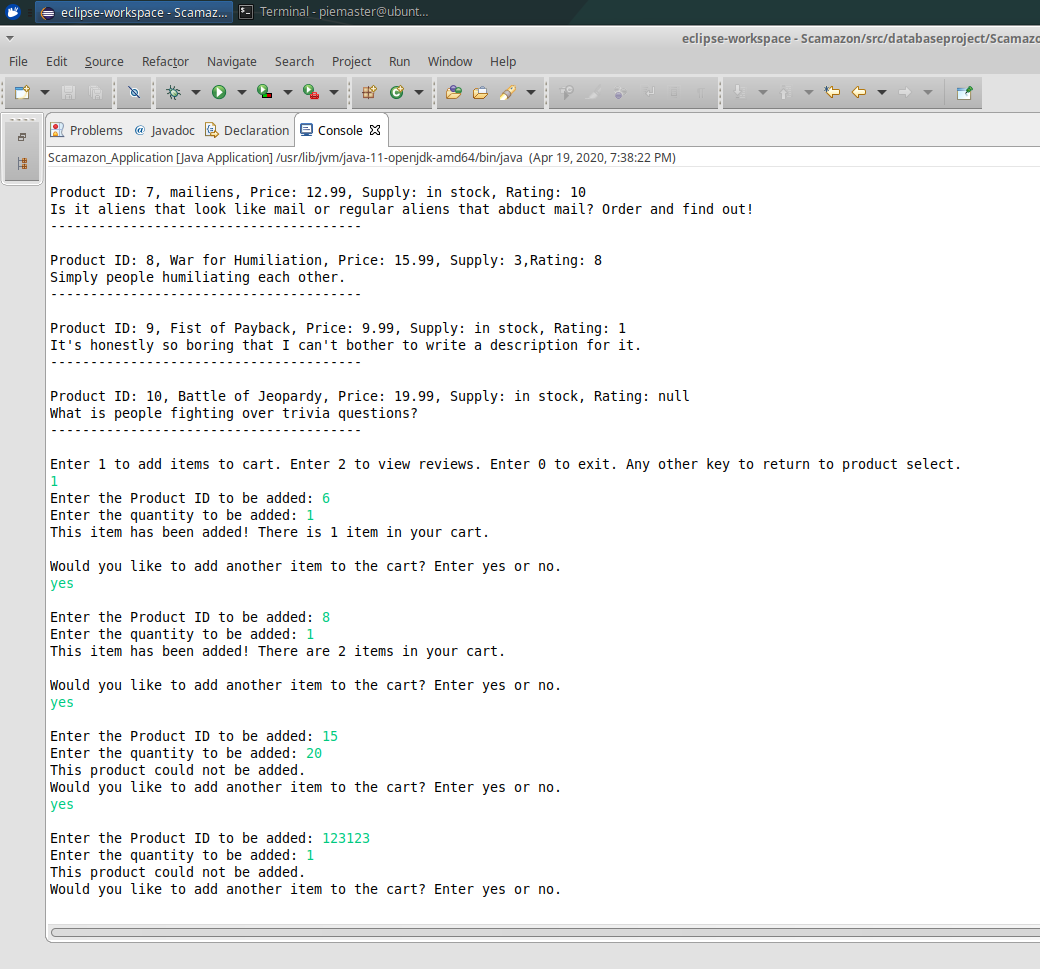
Upon customer login there is a menu allowing the customer to check items, view their cart, checkout, post or edit a review or change their password.





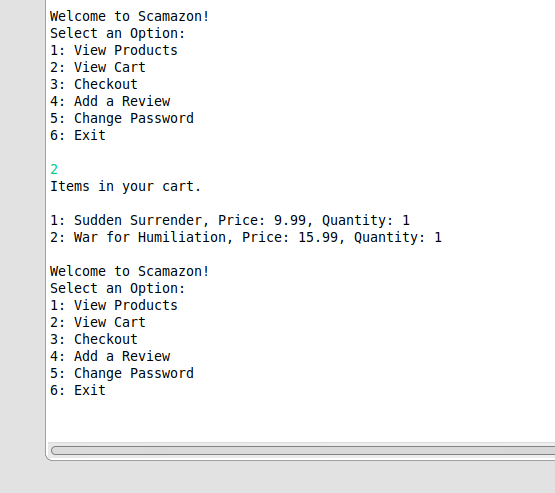
The customer can select from several categories in the database and the program will display the item information. They can then add items to a cart or view reviews for those items.





Items will be added to the cart if they are in the database and the stock is greater than the quantity ordered.

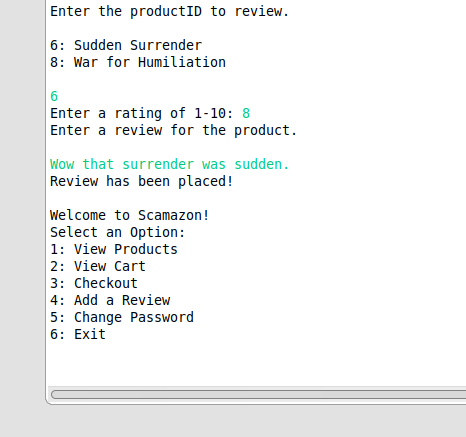
The customer can check their cart, this prints the contents on the screen. The customer will be told if the cart is empty.



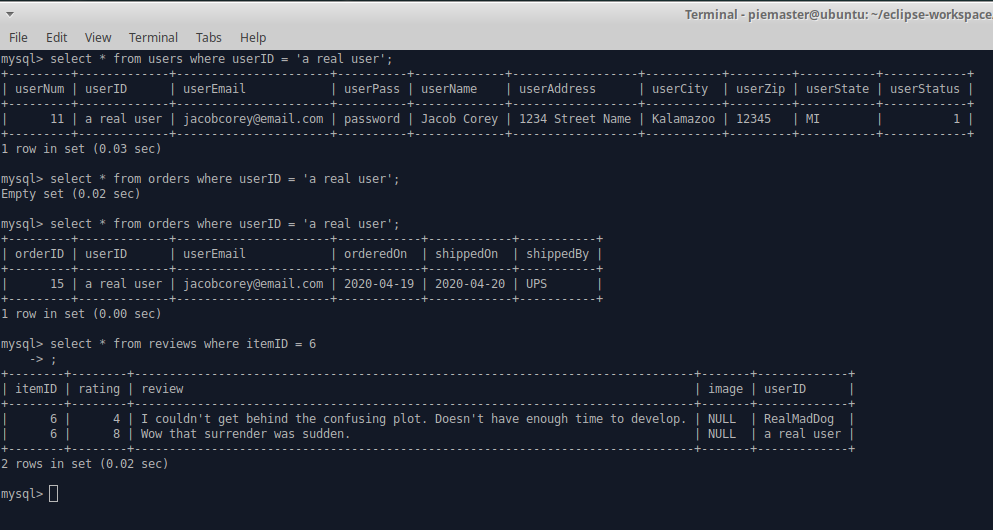
The customer can checkout which will “purchase” items in their cart. They will be asked if their address information is correct and will either update the address or process the transaction. If there is nothing in the cart it will notify the customer. The database will be updated with the order details if the order is processed.

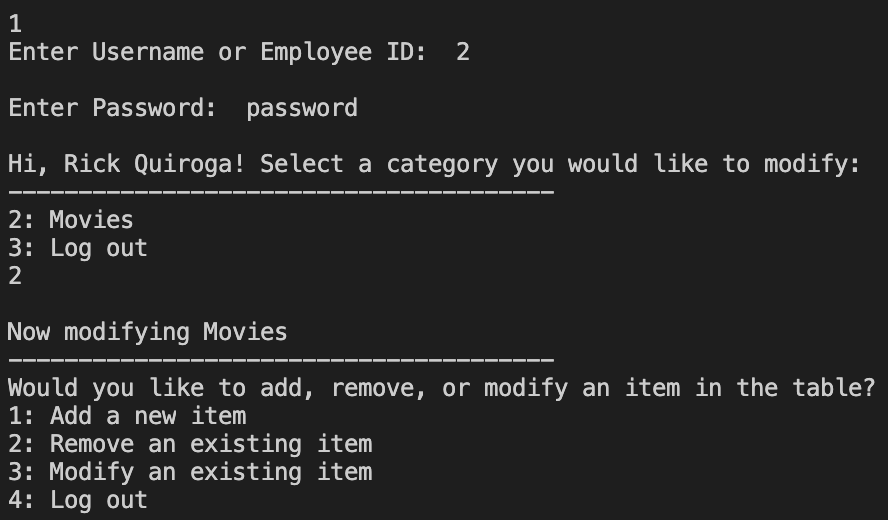
The program will place the order on the current day and ship it the day after.

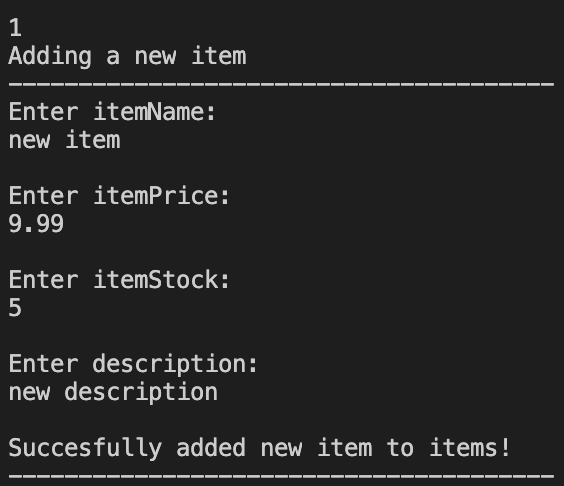
The user can then place a review for a product, which will go in the reviews table and update the average product rating in the database.

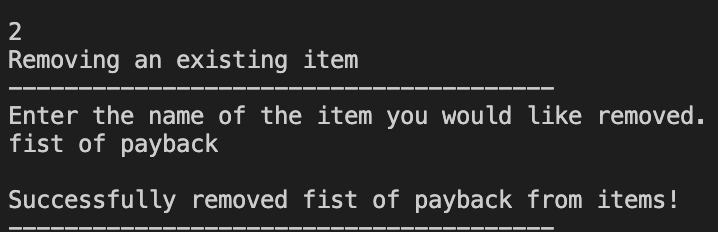


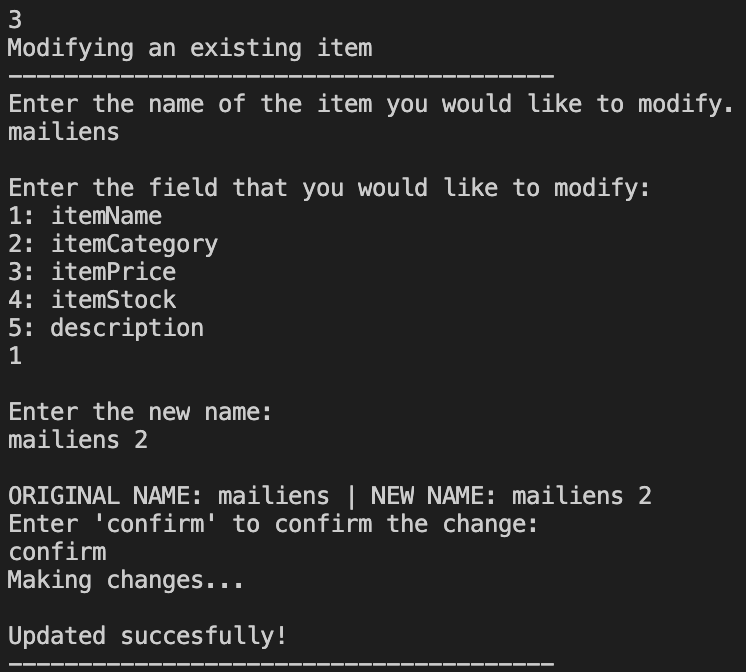
A screenshot of the database after a user was created, an order was placed, and a review was made.

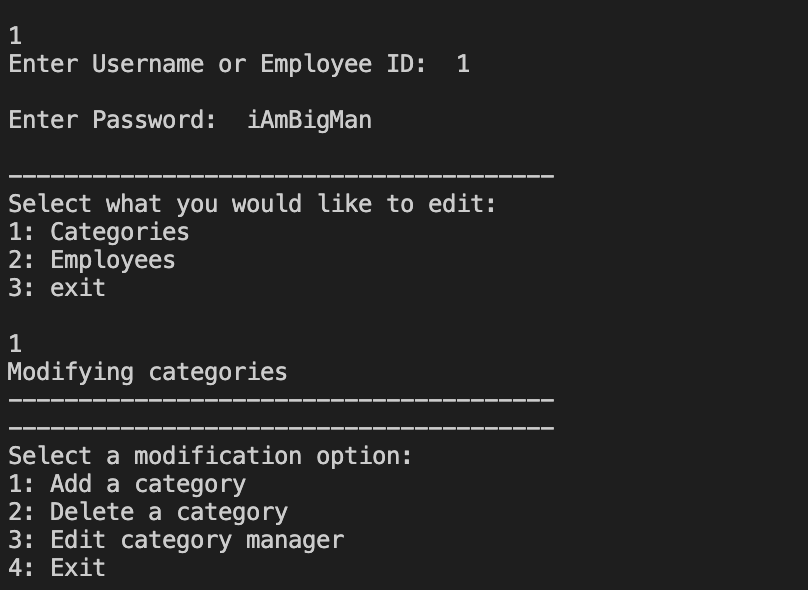
.

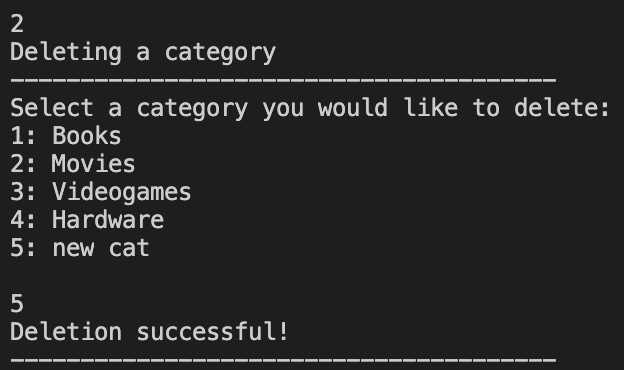
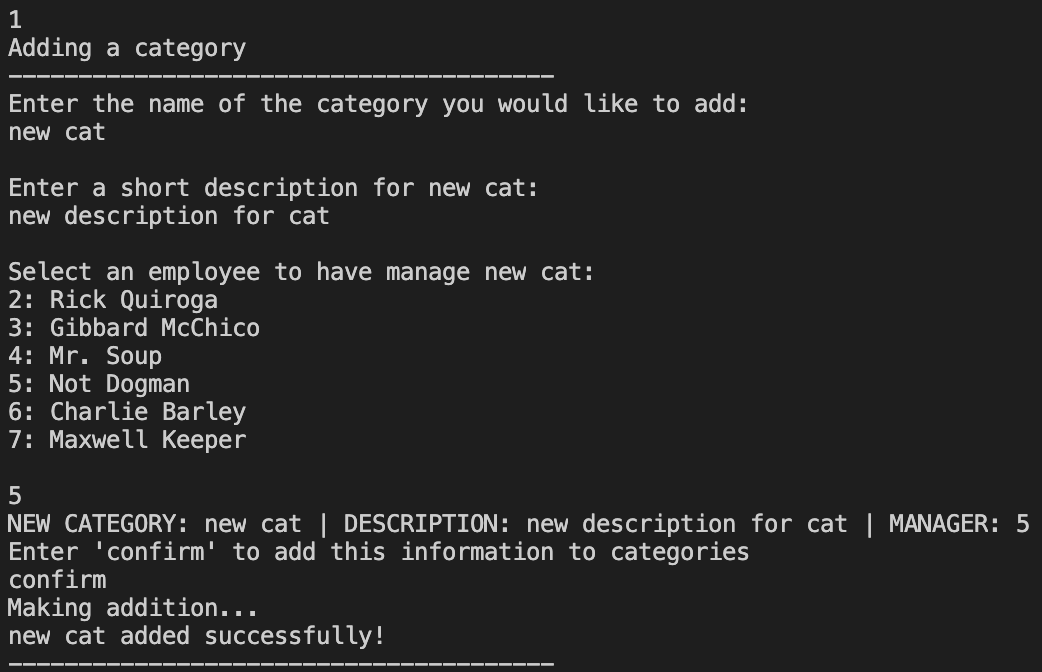
**For employee usage,** once the employee has logged in, they are given a list of the categories that they manage so that they may modify them. When one of the categories is chosen, they are provided with a choice to add an item to the category, remove an item, or update a current item’s information. Employees are not allowed to modify categories that they do not manage.

When adding a new item, the employee is prompted to provide the necessary information for the new item to be stored. Once this info is given, the item is pushed to the database. For removing, the employee is prompted to enter the name of an item in their category. Once they give correct input, that item’s stock is set to 0, making it unavailable.

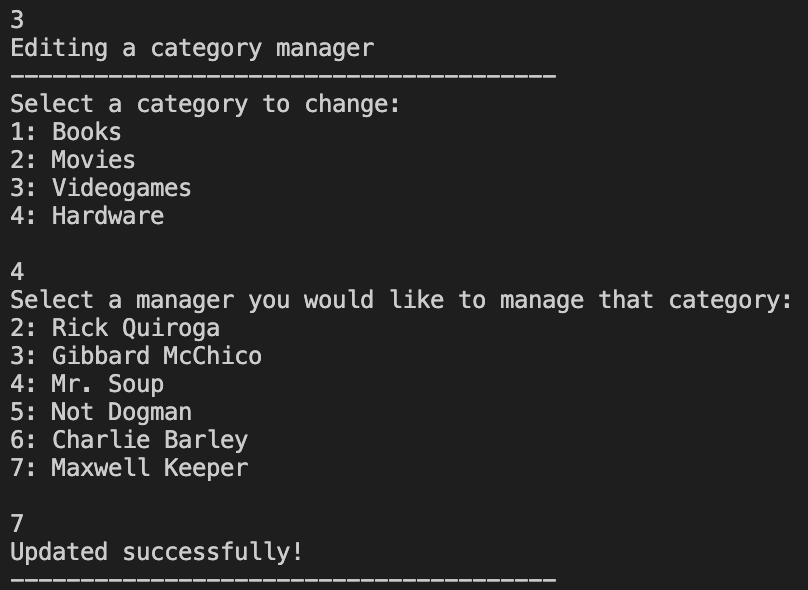


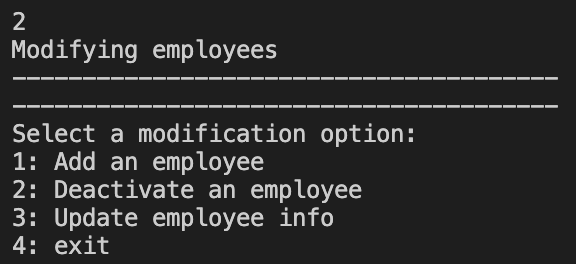
When modifying an item, the user is prompted to give the name of the item. When given, a list of options of what to edit is given. Once they give correct input, they are prompted to provide the new information. Once done, the program will show both the old and new information side by side and ask the employee to confirm the change. If confirmed, the new information is pushed to the database.

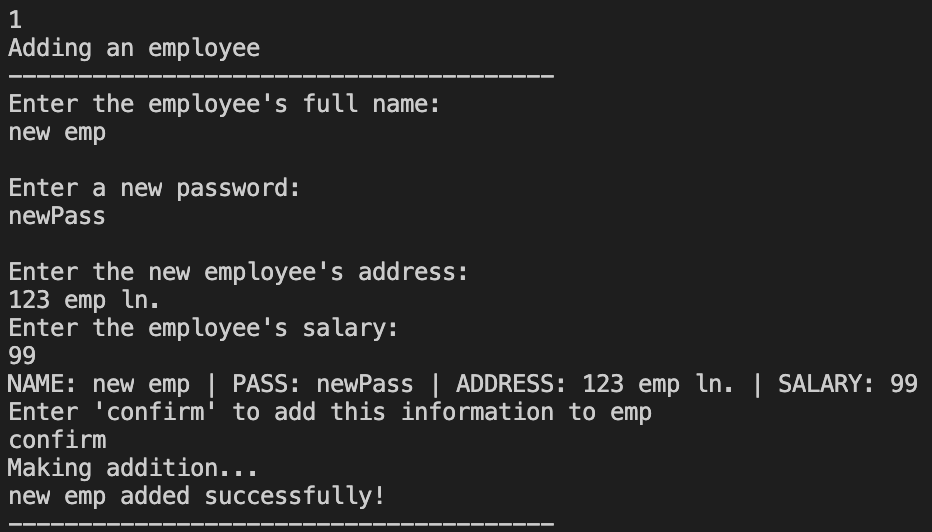
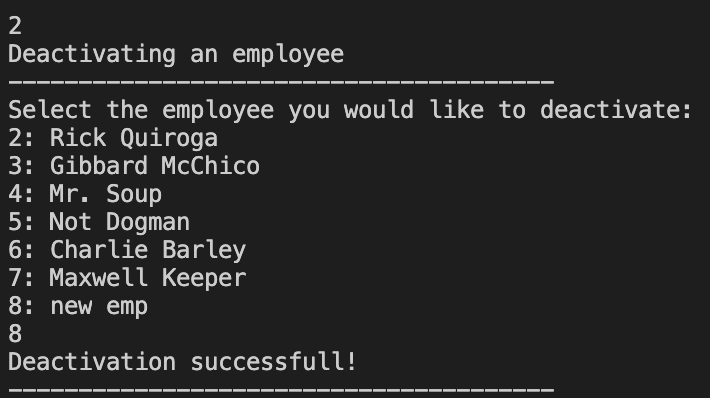
**For admin usage**, once the admin has logged in, they are provided the option to either modify categories or employees. Admin essentially ‘oversees’ what employees do.

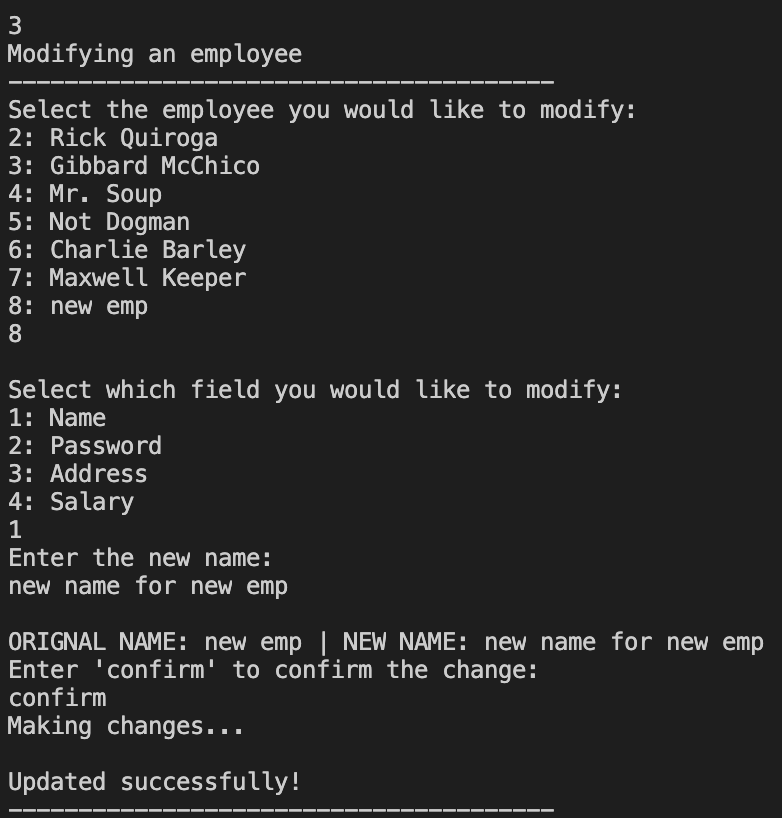
Adding a category asks the admin to provide the necessary info for creating a new category in the database. The admin chooses from the list employees to have manage it. For removing a category, the admin is simply provided the list of categories and chooses which to remove.

Editing category mangers prompts the admin to choose a category to modify, then a new manager to manage that category.



When choosing to modify employees, the admin can add, deactivate, or modify an employee.

Adding works as usual; the admin is prompted to give the necessary info to be pushed. For deactivating, the admin chooses from the list of employees. When chosen, that employees activation field is set to ’n’. Employees are not deleted because there information may still be needed elsewhere in the database.

When modifying employee information, the admin is given the list of employees. When an employee is chosen, the admin is asked to choose which field to modify. Once they have given input, the old and new values for that field are shown and the admin is asked to confirm. When confirmed, the information is pushed.

Lessons Learned:

* Using databases for these types of systems is extremely important. They allow for safe data storage and are easy to update and modify.
* Sanitation of input can be done by using prepared statements instead of solely using statements. This provides more security.
* Using programs to edit sql databases is primarily safer and easier than directly editing the database in a terminal. The program provides a sort of buffer for the user to understand what they are doing with the database, as well as making sure the user provides compatible information for the database.
* One database can be used for handling multiple ‘layers’ of functionality. For example, a user can’t modify tables directly, an employee can’t modify categories outside of their managing, and admin can’t modify the items inside the categories.
* Constraints, while vital to operating an efficient database, can sometimes cause issues in modifications that may seem to be free from foreign key and the like.
* When working with auto-increment keys, it’s best to have some sort of “removal indication” field (such as the empActive field in employees). Deleting something with auto-increment can cause some issues with missing values in a table.
* Often times, many different field types can achieve the same goal. It’s a matter of how the program uses them and how time and space efficient one wants the program and database to be.