|  |  |
| --- | --- |
| Shopping Cart Project  COP 4045 | Dr. Jaramillo | Abstract  Creating a shopping cart program in Python and implement it using Sqlite3 and GUI. The program will allow the user to login and register for an account. There are two types of user: regular and admin. The regular user can shop through the program and the admin user can make changes in the inventory.  Author: Tri Nguyen |

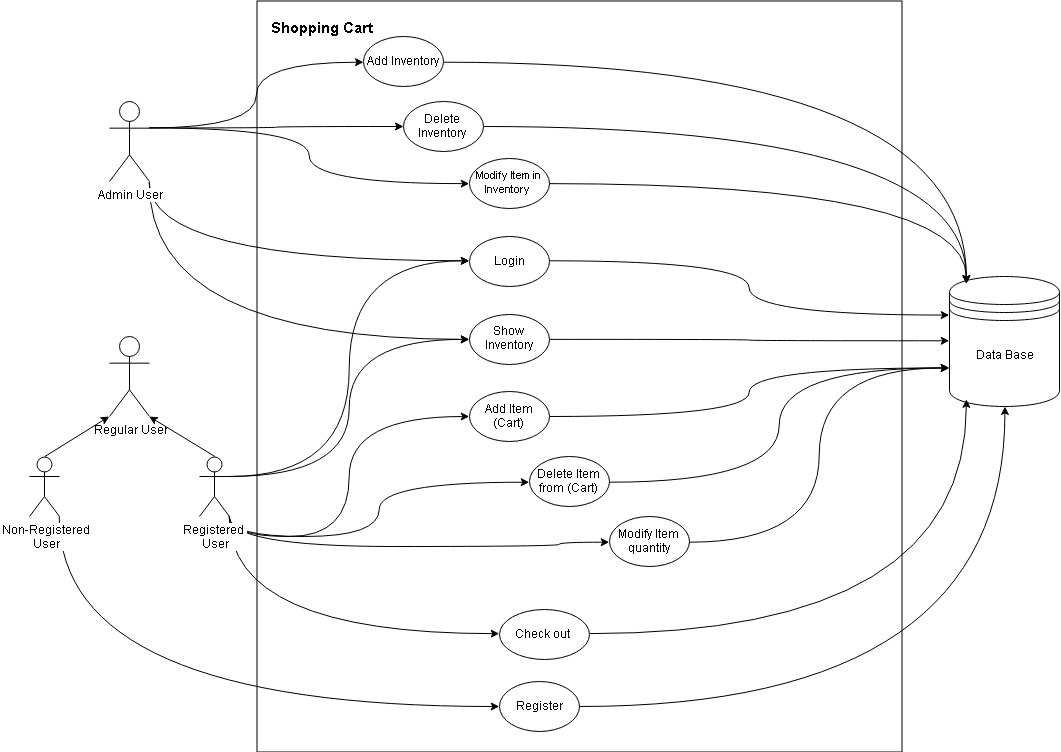
1. **INTRODUCTION**

The Shopping Cart program was based on the application from Chapter 16 in the textbook. To enhance the program, it needed a database and a better user interface. Sqlite3 for database and GUI for user interface was chosen as the enhancement. The program also includes the login and register functionalities. For the register function, a new user needed to create an account to be able to use the program. However, it is only a regular user account. The only way to get an admin account it that the account will be push straight to the database instead of the normal registration. The regular user can add item to their cart, delete item, or modify the item. The admin user can modify the Inventory of the program such as add new item, delete item, or modify existing item.

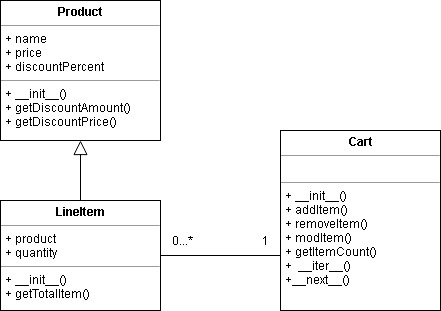
1. **PROJECT DESIGN**
2. Database Design:



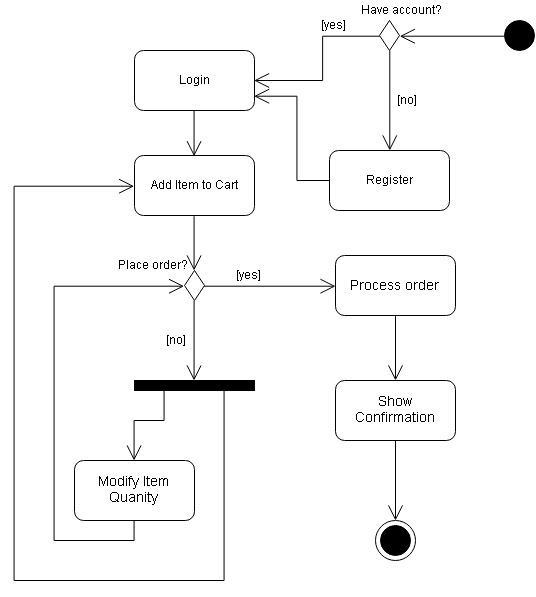
1. Use Case:



1. Class Diagram:



1. Activity Diagram (Regular User):



1. **APPLICATION DESCRIPTION (With Pictures of GUI)**
2. Login and Registration:
3. Login: allow the user to login
4. Register: allow an unregistered user to new account
5. Regular User:
6. List of functionalities:

* cart: Show the cart
* add: Add an item to the cart
* del: Delete an item from cart
* mod: Modify an item in your cart
* check: Checkout
* exit: Exit program

1. Admin User:
2. List of functionalities:

* show: Show items in the Inventory
* addDB: Add an item to the Inventory
* delDB: Delete an item in the Inventory
* modDB: Modify an item in the Inventory
* exit: Exit program

1. **WORK AND CHALLENGES**
2. Work:
3. Created the login and register:

In order to use the program, the user needed to have an account. The account will help save the shopping progress of the user so they can go back next time to continue shopping if they needed. Hence, two functions are introduced into the program:

* Login: check for the username and password. If the username is already existing and wrong password, the program will tell you to login again. If the username doesn’t exist at all, the program will tell you that it doesn’t exists so the user can register a new account.
* Register: ask the user to enter the username and password. It will only allow unique username so if the username already exists in the database, it will ask the user to try to login instead. You can only register as a regular user to shop in the program. The only way to get an admin account it that the account will be push straight to the database instead of the normal registration.

1. Adding a Sqlite3 database:

Enhance the program a Sqlite3 database to stores the Inventory items that available for the user to guy. These items are treated as they have infinite quantity. The database also stores the login credentials of both the regular and admin user. In addition, it also stores the items in the user’s cart that is not check out yet so next time when they login, the items will still be there. This is accomplished by checking the database with the user’s username to see if that person has anything stored or not.

Last but not least, Sqlite3 database is using Foreign Key Constrain so anything happened in the Inventory will affect the Item in the cart of the user. For example, if an item was deleted from the Inventory, the user who has that item in their cart will no longer be there either.

1. Regular and admin user commands in the program:

Except for modify command, the rest of the command for the regular user was from the given code. All I did was enhance them with the database. The modify command will change the items’ quantity based on the user’s inputs. It also has a check that it only allows change value that is greater than zero since it’s would be better if you just delete the item totally.

The newly added admin command was for the benefit of whoever maintaining the program since it will allow the admin to add, delete or modify Inventory. These functions were implemented using database queries.

1. Challenges:
2. Design the DB with Sqlite3 using foreign key:

Sqlite3 is a little bit different than MySQL which I am used to dealing with so I did not realize that the foreign need to be turn on manually for it to work (PRAGMA foreign\_keys = ON). It took me awhile to figure out and delay my progress on the project a little.

1. The “Cart” class in the given code is almost a pure interface class using Python “list” so it doesn’t have anything that deals with the database. I had some problem trying to modify the removeItem function and make the modifyItem function to work.
2. **CONCLUSION**
3. Brief Conclusion
4. Future work