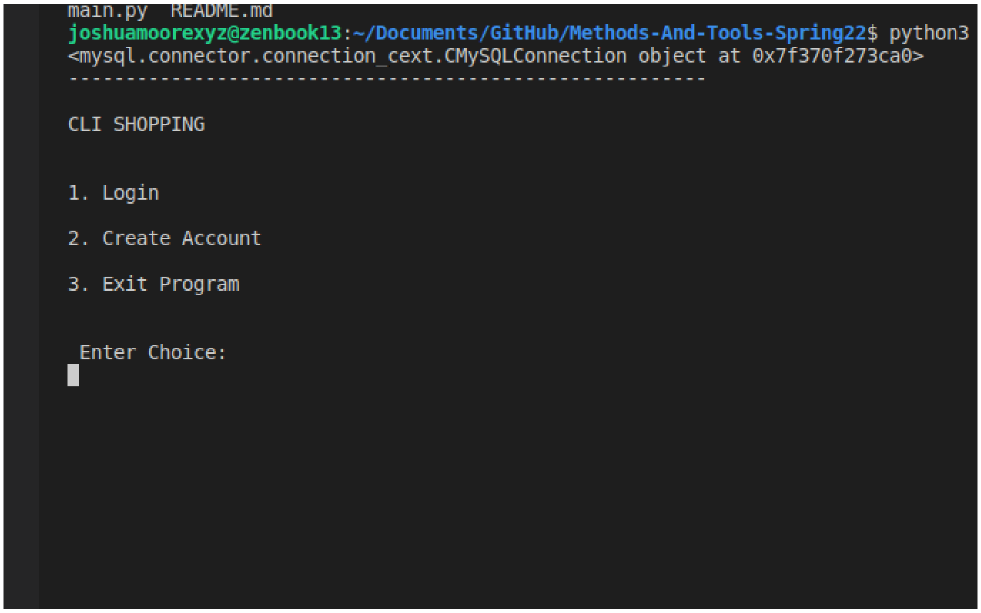
Final Report

Methods and Tools in SW Development



```
main.py  README.md
joshuamoorexyz@zenbook13:~/Documents/GitHub/Methods-And-Tools-Spring22$ python3
<mysql.connector.connection_cext.CMySQLConnection object at 0x7f370f273ca0>
-------------------------------------------------------

CLI SHOPPING


1. Login

2. Create Account

3. Exit Program


 Enter Choice:
```

I. Group Information

Group Number: 9

Group Member names/netIDs:

- **Joshua Moore**
  jjm702@msstate.edu
- **Sharad Rana**
  sr2096@msstate.edu
- **Azim Bazarov**
  Ab4908@msstate.edu
- **Rojal Bishwokarma**
  rb2298@mstate.edu

Group GitHub/GitLab repo link:
https://github.com/joshuamoorexyz/Methods-And-Tools-Spring22

Who was assigned to do what?

For these requirements an ad hoc workflow worked best where problems would be identified by one and then the team would pitch in on things when they needed to be done.

Most commonly:

Programming and Database: Joshua Moore and Rojal Bishwokarma

Documentation and design: Sharad Rana and Azim Bazarov

Tasks divided amongst the group:

Items
View all items in a category
View all items in the user's shopping cart
Add item from a category to their shopping cart
Remove an item from their shopping cart
Checkout the items currently in their cart
Removes items from the user's shopping cart
Edits stock information to lower the stock accordingly
Add an order to the user's order history
View the logged in user's order history
Edit the user's account
Edit the shipping information
Edit the payment information
Delete their account (and all the order history/shopping cart data associated with it)
Logout

II. Updated Detailed Class Diagrams

When creating your project, did any of your classes change?
We didn't add the ItemCategory() class in the final outcome.

Why did they have to change?
We changed because having another table in the database does not make much sense when this functionality can easily be implemented in the Inventory table with another column “Categories”.

Were the changes adding or removing items?

Removing items, such as an anticipated table “Item-Categories” in the backend database.Certain functions were encapsulated within the Inventory class instead.

Original design:

For class diagrams are as follows that we thought we will be implementing in our project:

Account
UserID Password Name Email OrderHistory ShippingInfo PaymentInfo

CreateAccount()
Login()
DeleteAccount()
EditAccount()
Logout()
AddOrderHistory()
ViewOrderHistory()
EditShiipingInfo()
EditPaymentInfo()

- CreateAccount()
  Lets user create an account
- Login()

Allows user to login

- DeleteAccount()
  Allows user to delete their account
- EditAccount()
  Allows user to edit their UserID, Password, and Name
- Logout()
  Allows user to logout
- AddOrderHistory()
  Allows order history of user logged in to be updated
- ViewOrderHistory()
  Allows viewing of users order history
- EditShiipingInfo()
  Allows the editing of shipping info for user
- EditPaymentInfo()
  Allows editing of payment info

Inventory
Item Name Category Price Stock

AddItem()
DeleteItem()
CheckPrice()
EditStock()
CheckStock()
SetCategory()

- AddItem()
  Allows an item to be added to inventory
- DeleteItem
  Allows and item to be deleted from inventory
- CheckPrice()
  Shows price of item
- EditStock()
  Allows Stock to be edited
- CheckStock()
  Shows current stock of an item category
  All right man
- SetCategory() specifies items category

Shopping Cart
ItemsInCart Total

ViewAllInCart()
AddItemFromCategory()
RemoveItem()
CheckoutItemsInCart()

- ViewAllInCart()
  View all items currently in the cart
- AddItemFromCategory()
  An item can be added from cart from a specific category
- RemoveItem()
  An item can be removed from the cart
- CheckoutItemsInCart()
  Checkout functionality

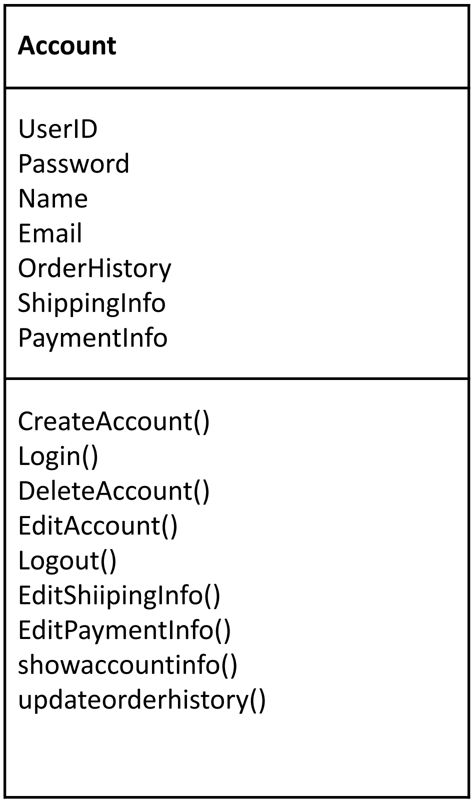
Item-Categories
Electronics Garden Health

ViewallElectronics()
ViewallGarden()
ViewallHealth()
AddCategory()
DeleteCategory()

- ViewallElectronics()
  View all items in the category Electronics
- ViewallGarden()
  View all items in the category Garden
- ViewallHealth()
  View all items in the category health
- AddCategory()
  Add new category
- DeleteCategory()
  Delete category

Updated design for class diagrams:

The following changes were made to our final class diagram from our original diagrams:



Account

UserID
Password
Name
Email
OrderHistory
ShippingInfo
PaymentInfo

CreateAccount()
Login()
DeleteAccount()
EditAccount()
Logout()
EditShiipingInfo()
EditPaymentInfo()
showaccountinfo()
updateorderhistory()

- CreateAccount()
  Lets user create an account
- Login()
  Allows user to login
- DeleteAccount()
  Allows user to delete their account
- EditAccount()
  Allows user to edit their UserID, Password, and Name
- Logout()
  Allows user to logout
- EditShiipingInfo()
  Allows the editing of shipping info for user
- EditPaymentInfo()
  Allows editing of payment info
- showaccountinfo()
  This shows all the account information
- updateorderhistory()
  This helps to update all the oder history

Inventory
Itemid Price Category

printinventoryelec()
printinventorygarden()
editstock()

- printinventoryelec()

  Prints the inventory lec
- printinventorygarden()
- Prints inventory garden
- editstock()

  If we edit stocks

Shopping Cart
Itemid Price Stock Category

AddItem():
printitems()
gettotalforitemsincart()
deleteitemfromcart()
deleteallitemsfromcart()
checkoutitemsincart()

- AddItem()

  This functions allows to add the item

- printitems()

  Helps to print the item

- gettotalforitemsincart()

  This gets the total info from the cart

- deleteitemfromcart()

  This deletes the item from the cart

- deleteallitemsfromcart()

  This deletes all the items from the cart all at once

- checkoutitemsincart()

  This checks out the item in the card individually

** after this point, add any class diagrams for classes that did change. Make sure it's the entire diagram, not just the items that were added/removed/edited -- one class diagram per page

III. Conclusions

If you were to flesh out your project more, what requirements do you think could be added?

Flask or a similar framework could easily be added with some html/css to take the shopping experience to the web.

Are there any overall design choices you wish you could have changed in hindsight? What are they and why?

For a small project like this one, I'm not sure that having an sql database was needed. If the size of the data grew that would be fine, but for a proof of concept next time lists would work fine with implicit indexing of information.

What difficulties did your group have?

One of the difficulties for the group was learning how to use the python sql connector. If you do not know sql syntax and or python well it becomes a hard task for everyone to participate in. The use of mysql-workbench makes it easier, {with some special commands for the snap package to have proper permissions of course}

What did your group learn overall from the project?

Overall, the group has a better outlook on how easy and fast it can be to have a proper backend database that can be tied together with some fast structural code from python.

Python was the perfect language for the project. Porting the code over early was a good decision as all the type casting that happens in this project,would have made it something tricky in the C family of languages.

Also working with a virtual cursor to administer backend SQL commands is a novel approach that ties this together, which we were glad to be exposed to in the project.