Initial Design

Classes:

MarketPlace

Buyer

Seller

Item

MarketPlaceAdmin abstracts Buyer & Seller

MarketPlace includes:

METHODS:

Register()

makeID()

viewStock()

login()

updateCreds()

search()

VARS:

Buyer includes:

METHODS:

register()

login()

Buy(item)

updateCreds()

viewMarketplaceItems()

search()

VARS:

Seller includes:

METHODS:

register()

login()

updateInventory()

updateName()

updateEmail()

addInvItem()

delInvItem()

updInvItem()

updateCreds()

search()

VARS:

Item Includes:

METHODS:

registerItem()

Add()

updateInfo()

delete()

notifySeller()

VARS:

(int) quantity

(string) name

(string) information

(String)category

MarketPlaceAdmin includes:

METHODS:

login()

listBuyers()

listSellers()

updateCreds()

listItems(user)

purchanseHistory(user)

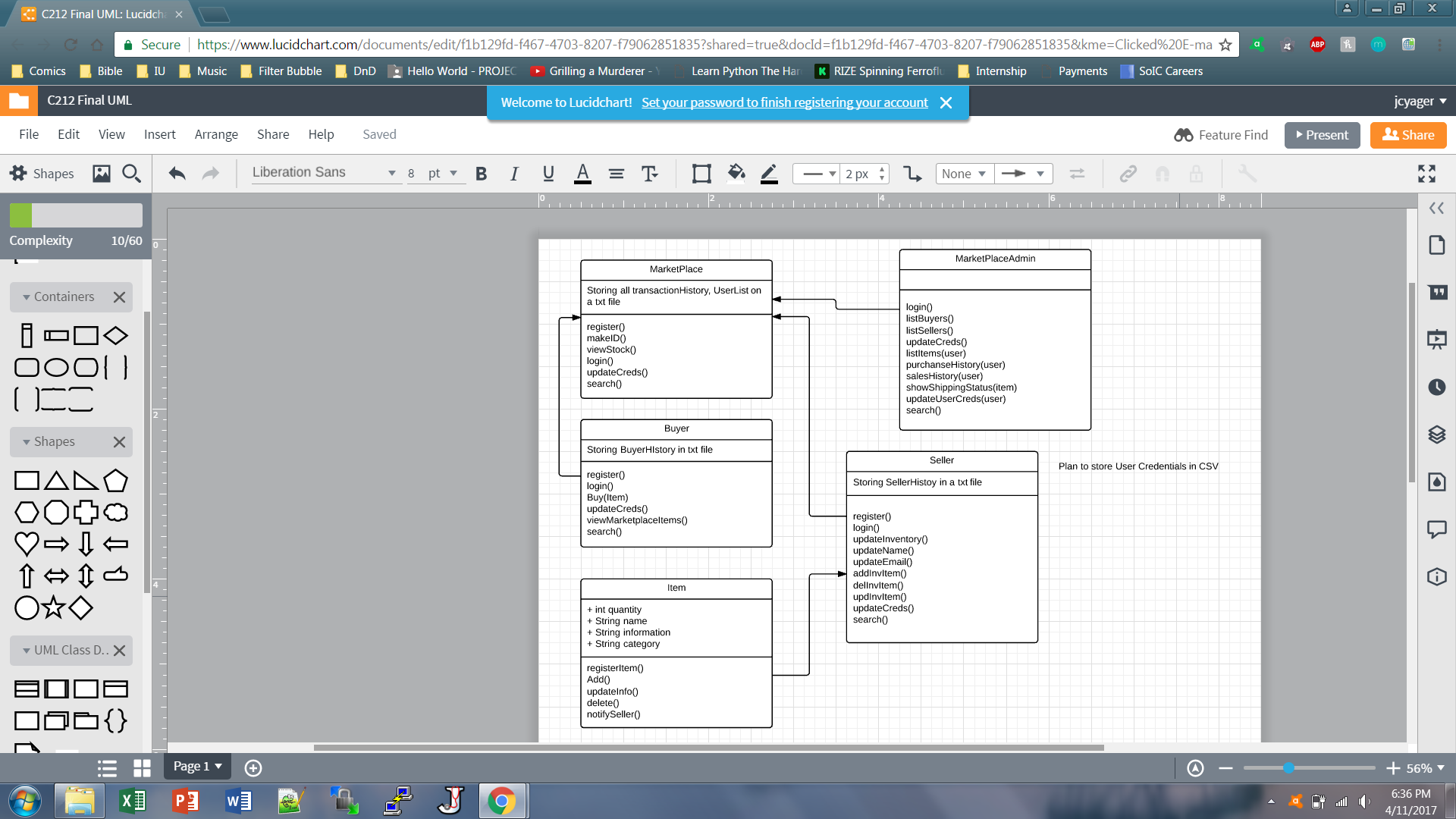
salesHistory(user)

showShippingStatus(item)

updateUserCreds(user)

search()

VARS:



Updated Design

Classes:

MarketPlace

Buyer

Seller

Item

MarketPlaceAdmin

MarketPlace includes:

METHODS:

MarketPalce()

helpMenu()

register()

makeID()

uniqueUsername()

newPassword()

accountType()

newEmail()

login(String, String)

VARS:

loggedin

userGetter

passGetter

emailGetter

typeGetter

updater

currenUsername

Buyer includes:

METHODS:

Buyer()

Buyer(String)

register()

login()

Buy()

listItems()

newPassword()

newEmail()

notifySeller(String,String,int,int)

history(String, String)

search(String)

VARS:

input

reader

currentUsername

updater

passGetter

reader3

reader4

Seller includes:

METHODS:

seller()

seller(String)

register()

login()

addInvItem()

delInvItem()

updInvItem()

updateCreds()

search()

notifySeller()

currentUsername()

newPassword()

newEmail()

clearMOTD()

motd()

VARS:

intemList

nameScan

infoScan

categoryScan

quantityScan

input

currentUsername

updater

passGetter

emailGetter

reader3

reader4

MarketPlaceAdmin includes:

METHODS:

marketPlaceAdmin()

login()

listBuyers()

listSellers()

updateCreds()

listItems(String)

purchanseHistory(String)

salesHistory(String)

updateOther(String)

newPassword()

newEmail()

search(String)

VAR:

input

currentUsername

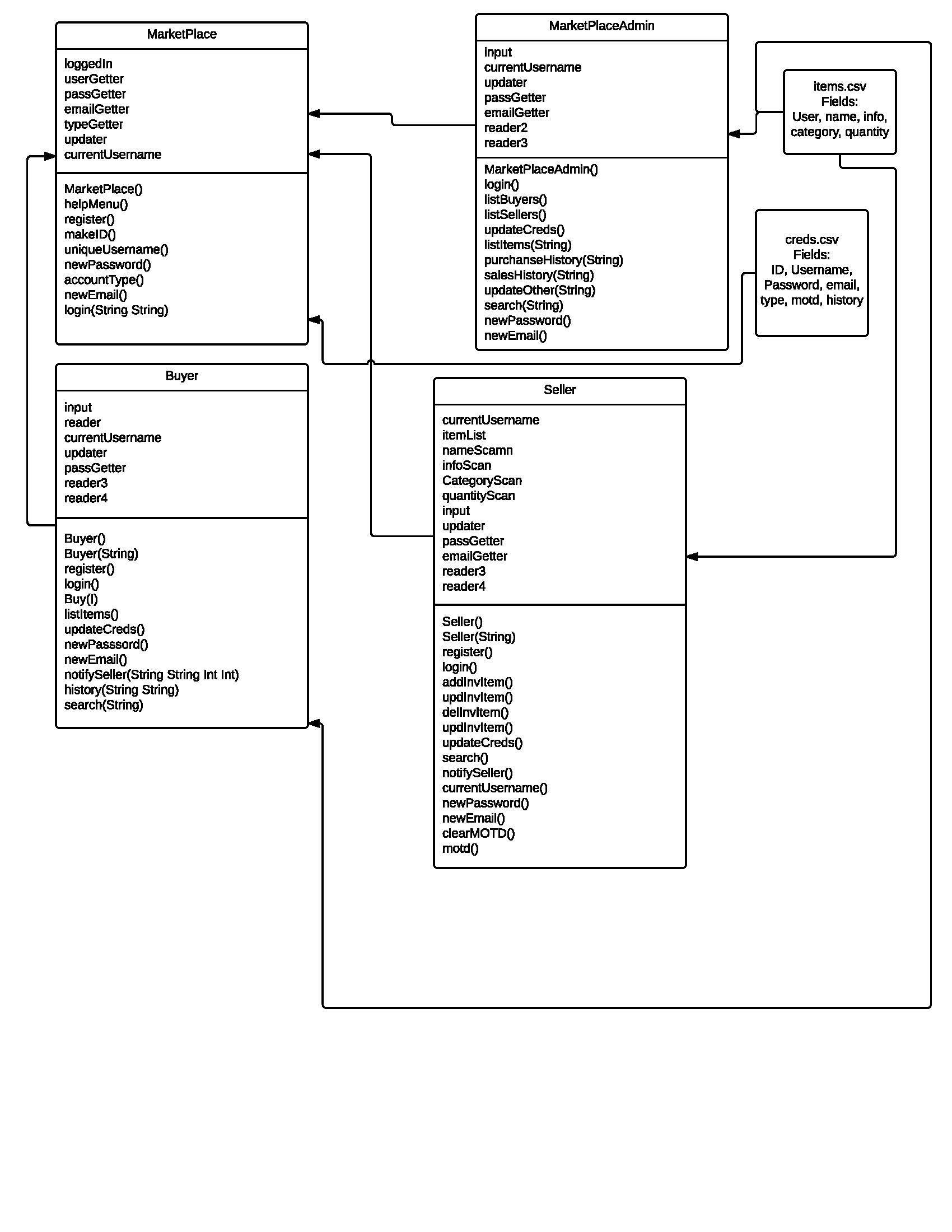
updater

passGetter

emailGetter

reader2

reader3



Discussion

Our largest change was that we destroyed the items class. We decided that interacting with items using a CSV file was more efficient and better for memory. Our two CSV files are how the program maintains state persistence. Everything that happens while the program runs is essentially recorded in a CSV. Another large change we made was to focus the creation of new users, using register(), happens in the MarketPlace. Creds.CSV is where all users are recorded.

Testing Activities

How testing works:

* run MarketPlace in the council
* move through all methods
* in each method we intentionally run an error where possible
* run the method successfully
* repeat that process with Seller.java
* repeat that process with Buyer.java
* repeat that process with MarketPlaceAdmin.java

At the bottom of the Testing.dox file we pasted the data from both the creds.CSV and items.CSV. That data holds our preexisting users and their items, as well as the new users and items that were created during testing.