Task1 Fixes:

Issues:

Aesthetics Fixes:

-fixing method and attribute to united format

- fixing todos

-change set to lists

-concurrency (check if automatically in Django)

Store:  
-Filter by features: A user may filter by price range, product rating, store rating, and category

-Accesses need to hold who nominated you + each user needs to know the people he nominated

Alternatively, accesses hold an AccessState abstract object that extends to Founder-Manager-Owner and each object AccessState (that will hold permissions aside from roles) knows to answer if an operation is legal.

-adding to accesses, there need to be an extra option for permissions

-Option to change permissions and authorities

-Close store: by store founder, makes the store ‘inactive’. A closed store can be reopened. While closed – it can’t give information for any member who isn’t an owner\founder\admin. When closed – the owners and managers get a notification, but their authorities stay the same

User:

-member\_to\_guest: Logout as a member make you a guest

-User extends Guest (change architecture): change the structure of the code accordingly

Service:

-add getLogger

-Better descriptions for logger

-change the output of the functions be either Boolean or exception. Not both

- authentication

StoreFacade:

-Split external services

-something with purchaseCart and systemCheck

-Make External services Singleton (so for example we could check the payment and supply from a cart method)

-supply approval at purchase

-after open Market, there needs to be a call to supply service

-after open Market, there need to be a call to payment service

-list of online users not just guests in StoreFacade (opt)

Repo Fix:

Fix use cases

Fixing terms alphabetically and add links  
  
Tests:  
-Fix all tests + add assertion also in fail scenarios (that the system stayed the same)

-Add guests tests

-concurrency Tests: give a scenario and check several time (in a loop) if no error occurs

-Add tests about member gets his previous cart  
-We can do different setups

-TESTS SHOULD HOLD AN INSTANCE OF SERVICE

-Finish Integration Tests

-Fill the new tests

Task2 Issues:  
creating service data types – service data types

Messages with observers

Mapping the web socket to specific user

Password validator goes through Django

Service:

-Service objects

Store:

-Discount policies

-Lottery

-Auction

-Purchase Policies: lottery auction…

Tests:

-Add purchasePolicy tests

Gui:  
  
- תהליך אפשרי למינוי תפקיד/הרשאות - בדיקה על אם המבקש יש לו הרשאות של למנות-> הוספת תפקיד למי שבוקש עליו -> מסך נוסף של שאלת איזו הרשאות תרצה למי שמינית עכשיו - ושליחת בקשה ליצירת access