

TEAM D

An electronic trading system

We want to develop electronic trading such that every responsible user can trade their items.

There are three types of users to this system: 1. Super users (SUs), 2. Ordinary users (OUs), and 3. Guests (GUs).

An SU can perform the tasks below:

1. Process applications and decide who can be OU and who should be denied with justifications.
This has not been implemented completely, as the SU can elevate OU to SU status as well through the Django admin interface. The SU can elevate an OU to SU or make an account inactive.
2. Process items the OUs submitted intending to sell, some can be publicly available on the system and some may be denied and communicated to the OU.
A SU can view all items and remove them from the admin interface.
3. Send warnings to OUs who received two complaints or the average grade is lower than 2 with at least 3 evaluators, should an OU receive two warnings this OU should be removed from the system automatically. The SU has the power to remove any OU(s) with justifications at any time.
The SU has the ability to remove an OU at anytime from the admin interface with justifications. They may see the comments and ratings on the admin interface as well.
4. Collect transaction statistics for a certain period (a day or a week) and/or a certain OU(s).
A SU can view the transaction statistics of items and the OU that are selling them in the admin interface, they can see what time and day an item is bid on.
5. Settle disputes between sellers and buyers and apply warnings/removals accordingly.
A SU can remove both sellers and buyers from the database and may send warning messages as well.

An OU can do the following tasks:

1. Features enjoyed by a GU.
Once logged into the site the GU essentially becomes an OU, and can browse, and report suspicious items to the SU.
2. Deposit or withdraw money (symbolically, no real money is involved in the system) from own account. After a transaction, the buyer's money is transferred to the seller. A bidder cannot place a bid with amount more than s/he has in the system.
We attempted to use Stripe but the implementation involved possible real money, so there should've been one test card that is matched against a string of another OU's card balance for the transfer of funds.
3. Submit the bid to buy an available item, the OU who owns the item however decides if this transaction should proceed.
While submitting a bid, the OU must enter an amount greater than the current bid, or otherwise it will not go through.
4. Submit the information (a picture or video, title, key words, time limit, price range) of the item(s) s/he wants to sell.
The OU can enter the Create a Listing Tab and enter title, keyword, starting bid, and a url of the desired image.

5. Sell the item(s) s/he posted, the OU may choose whom to sell, if not the highest bidder was chosen, a reason must be provided.
The OU who is selling the item has the ability to close the auction and sell it to the OU who is bidding on the item(the highest bidder).
6. File complaints to SUs against certain OUs whose item s/he purchased has some problem.
Complaints will show up on the SuperUser/Admin interface as comments/string.
7. Grade an OU *after* buying an item from him/her, 1 being the worst and 5 being the best. Any OU making 3 1 (worst) or 5(best) rating to others, the OU must see the SU to make sure there is no problem, a warning or removal could ensure based on SU's judgment.
An SU may remove an item and can also receive the gradings of OU's as the database is updated in real-time and at their discretion may remove an OU from the site.
8. See his/her own transaction history.
The watchlist data is preserved in the sqlite3 database and is still there after logging out and into the account.
9. Change his/her password, name, address, phone, and credit card number.
Through the admin interface the name, password, and all related information of an OU may be changed.

A GU can do the following task

1. Browse/search available items based on title, keywords, price range, and ratings;
The GU can go through the listings and see the title, keywords, current price and rating for the items.
2. Report to SU about the suspicious items (stolen) s/he spot from the site, if proven true the SU will remove the item and the OU who posted it and generate a report to be sent to police;
Each item has a section for reporting about possible suspicious items that may be filled put and sent to the SU. However only an OU is able to do this as we ran into errors.
3. Apply to be an OU.
The GU has the ability to register on the site and should've been met with a message that they are awaiting approval from the SU admin with the is_active section in the admin interface, when approved the SU may check the box and allow them to sign in. We had errors with the registration screen, and the GU becomes a OU as a result.

Other system constraints.

1. OUs removed from the system by SUs will be blocked for future re-application.
The SU interface is essentially the Django administration interface, and there should've been a blacklist category within the database that will be updated once a SU removes an user from the system accordingly.
2. Items removed from the system by SUs will be blacklisted from system.
The SU interface is essentially the Django administration interface, and the blacklist category should've been made in the database and should've been updated once a SU removes an item, but the blacklist was introducing additional problems.
3. The system can distinguish the interest of an OU during the login session and show different GUI according to the users' browsing or selling/buying history. No adaptive GUI for GUs or SUs.
The GUI is not adaptive, but the OU has the ability to choose a category of interest from the drop-down menu, and as a result they will be able to see a different screen of items based on such interests(categories).

4. GUI is required, could be a local application or web-based one, no preference
The barebones GUI utilizes bootstrap, Django, CSS, HTML and JavaScript. It was deployed on a local web-based server.
5. A creative feature with 10% of the project determined by each team, considerably creative ones (positive outliers) could obtain special bonus.
A Watchlist for saved items of interest is saved in the database with the item_id.
When a Guest User attempts to bid on the item, they will be redirected to the Login page, where they may also apply for registration of an Ordinary User account or if they have an account already they may login.

There were no conflicts, everyone gets the same grade.

Link to Github (Used VS Code):

<https://github.com/mrahman017/EBID-EBUY.git>

Mohammed Rahman

Thierno Dicko

Mahfuzul Islam

John Kim