Mandatory Extensions

- Give the admin user the ability to undo every action taken by regular users that can reasonably be undone.
    - <u>Trading</u>: An admin can undo the creation of a trade, regardless of the status of the trade. If appropriate, this will swap items back to the individual accounts.
    - <u>Items</u>: Admins can undo the creation of items through the “Manage Warehouse” view.
    - <u>Accounts</u>: Admins can undo the creation of an account by banning them through the “Manage Accounts” view.
    - <u>Wishlist</u>: Admins can undo wishlist item addition by removing items from someone’s wishlist, this is done through the “Manage Peer Wishlists” view.
- Allow the admin user to adjust all of the threshold values, either from inside the program or by typing into a text file
    - Admins can go to the “Thresholds” view and change the thresholds from there. Accounts can also see to know their restrictions in read-only mode.

- When the user sees something they want to borrow, they can ask the program for a suggestion of what they could lend this user in return:
    - Accounts will see recommended items to lend in green when initiating trades.

- Create a new type of account that allows a user to look at the various parts of the program without being able to trade or communicate with the admin user.
    - ➢ In the login screen the user can select the “Demo Mode” checkbox, in which case other accounts are unable to interact with them. This feature can also be used by existing accounts.

Non-Mandatory Extensions

Note: We implemented 6 features (GUI counts as 2) from the non-mandatory extensions.

1. Locations: In the register screen, accounts are required to specify their location, or select a pre-existing location. They will only be able to trade with people in the same location.
2. Point System: Users who have completed above a certain admin-specified threshold of trades are able to promote themselves to a trusted member through “Account Settings”.
3. Admin undo abilities, in our system admins are able to:
    a. Ban users.
    b. Promote members to trusted community members.
    c. Promote members to a moderator.
    d. See undo abilities above
4. Include a number of different types of accounts or statuses for an account.
    a. Vacation Account: An account on vacation, will not be able to trade.
    b. Trusted Community Member: Able to confirm items from people's inventory.
    c. Banned: An account who is unable to login.
    d. Moderator: Freeze and unfreeze accounts, make other "Trusted Community Members" and is a "Trusted Community Members"
5. Replace your text UI with a Graphic User Interface (GUI).
    - JavaFx

Other Extensions:

1. Supporting different languages: At the start of the program, an account can specify which language they would like to use.
2. Allow users to change their password through Account Setting
3. Security: The program encrypts all users’ passwords so that anyone who got access to the database cannot login on behalf of other users.

Multi User Support

Our program supports multi-users. This means users are accessing the same data set and able to modify data without crashing.

- External storage: The data is stored in a Postgres server on the internet so that anyone who has internet access can use the program concurrently with others.
- Data transfer: Client and the server use JSON to safely transmit data and reject illegal requests to each other. This logic is implemented in Flask (backend) and GSON (frontend).
    - The source code is available under the “backend” folder
    - When save() methods in JsonGateways are called, it files necessarily variables into a JSON using GSON library
    - A POST request containing the JSON will be sent to the server. Server parses the JSON object to verify if it is safe to add the database
        - Save it to the database if the request is validated
    - When populate() methods in JsonGateway are called, it sends a GET request to the server
    - The server fetches all items for the corresponding entity and returns a JSON representation to the client
    - The gateway parses the JSON using GSON library and proceed all the values to corresponding use cases