SessionManagement

instance()
init()
login()
logout()
update_activity()

MyPass

init()
auto_lock()
is_weak_password()
register_account()
add_vault_item()
view_vault()
unmask_item()
delete_vault_item()
copy_to_clipboard()
clear_clipboard()
is_valid_email()

Observable

add_observer()
notifying_observers()

EmailNotifier

update()

PasswordBuilder

length()
include_uppercases()
include_number()
include_characters()
build()

MaskingOrUnmasking

get_data()
toggle_mask()

Mediator

notify()

PasswordRecovery

recover_password()

SessionManagement is the first class that uses the singleton pattern to make sure only one instance of the user interaction is there. Then MyPass is the main class that matches all the features into this class. Features such as auto_lock, is_weak_password are available which are described as functions. Observable and EmailNotifier uses the observer pattern to notify the users of the weak and strong password. PasswordBuilder uses the builder pattern and uses different features to find a strong password for the user to use. MaskingOrUnmasking uses the proxy pattern and lets the user unmask the data. Mediator lets the UI interact with each other without any issues. PasswordRecovery uses the chain of responsibility pattern for more security by asking three security questions.
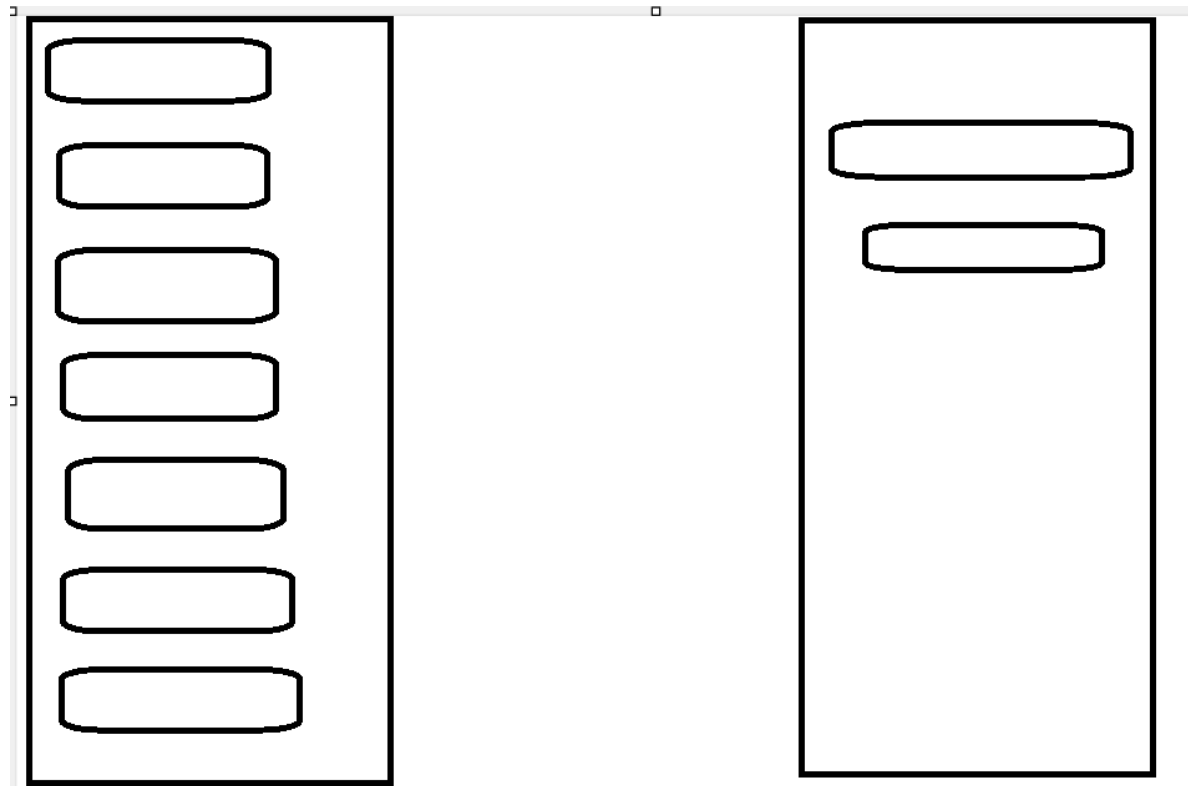
The User Interface shows the first tab where the 7 options match the options I have in the code. The second tab is for user after they input register.