

Amelia Chapman UML diagrams & Pseudocode

Person
Name: str Age: int Favorite animal: str Favorite color: str Friend list: [str] or [Person] Blocked list: [str] or [Person]
Change name Change age Change favorite animal Change favorite color Remove/block friend View friends View messages Send message

Social Network
List of people: [Person] (see Person above for attributes of people) List of usernames: [str]
Create account (See Person for editing details, viewing friends, blocking friends, viewing messages, sending messages)

loginMenu

- Have create new account, login, and quit as options
- If user selects create new account, go to createNewAccountMenu
- If user selects login, get input to enter their username and check if their username is in usernameList
 - If username entered IS in the list, bring up manageAccountMenu and set p1 to the index of their username but for listPeople
 - If the username entered is NOT in the list, tell them that it is wrong & bring up the loginMenu again
- If user selects to quit: end program
- Else: tell user their input is invalid and show loginMenu again

createNewAccountMenu (def create_account in social_network_classes.py)

- Have user enter username, name, age, favorite color, and favorite animal

- Add this to the list of people and the username to the list of usernames (they will be the same index which will be used later)
- Afterwards put them on the manageAccountMenu

manageAccountMenu

- Have edit details, manage friends, view/send messages, logout
- If user selects to edit their details, bring up editDetailsMenu
- If user selects to manage friends, bring up manageFriendsMenu
- If user selects to view/send messages, bring up messagesMenu
- If user selects to logout, bring up login menu
- Else: tell user that their input is invalid and show manageAccountMenu again

editDetailsMenu

- Have edit name, edit age, edit favorite color, and edit favorite animal as options
- If user select to edit name, ask for input for new name:
 - Save new name in the str name variable and go back to manageAccountMenu
- If user select to edit age, ask for input for new age:
 - Save new age in the int age variable and go back to manageAccountMenu
- If user select to edit favorite color, ask for input for new favorite color:
 - Save new favorite color in the str favColor variable and go back to manageAccountMenu
- If the user selects to edit favorite animal:
 - Save new favorite animal in the str favAnimal variable and go back to manageAccountMenu
- Else: tell user their input is invalid and go to manageAccountMenu

manageFriendsMenu

- Have add a new friend, view friends, and block a friend as options
- If user selects to add a new friend, ask for input for their name
 - Save name in a list of their friends
- If user selects to view their friends, print the list of their friends
 - If they do not have any friends added (the list is empty), print that they do not have any friends added
- If user selects to block a friend:
 - Print a list of their existing friends
 - Ask for input on which friend they would like to remove
 - Remove that friend from their friends list
- If user selects to go back, go to manageAccountMenu
- Else: tell user their input is invalid and show manageFriendsMenu again

messagesMenu

- Have a list of all of their friends and ask for input on who they would like to contact
 - Bring up a chat with that person and ask them to enter a message
 - Save it to the recipient's inbox

- They can also view their inbox which just prints `p1.inbox`
- They can also have the option to go back in which case just go to `manageAccountMenu`
- Else: tell user their input is invalid and show `messagesMenu` again