Matthew King, Joe Canero, Gary Patricelli, Dan Thielke

Dr. Monisha Pulimood

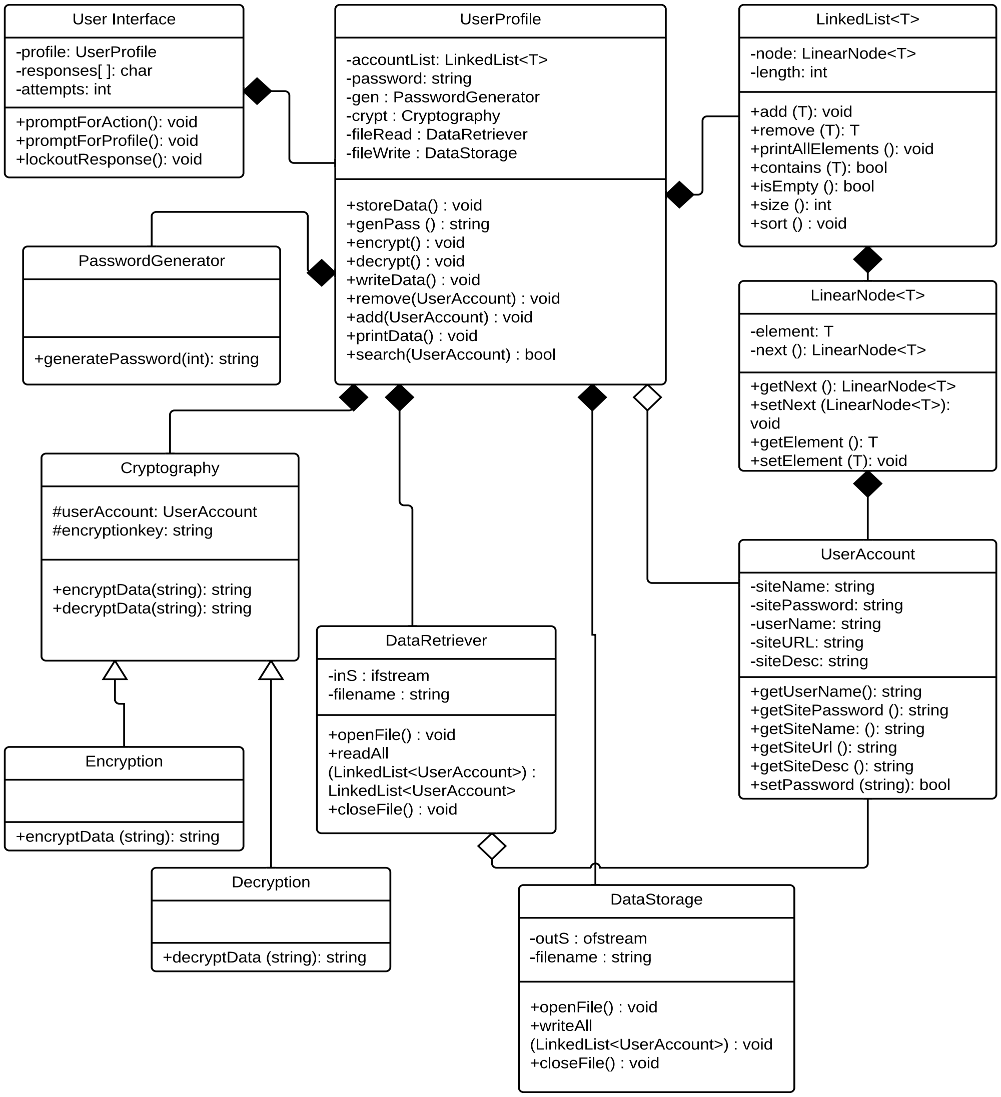
CSC260-02

13 November 2012

Collaborative Project Stage III

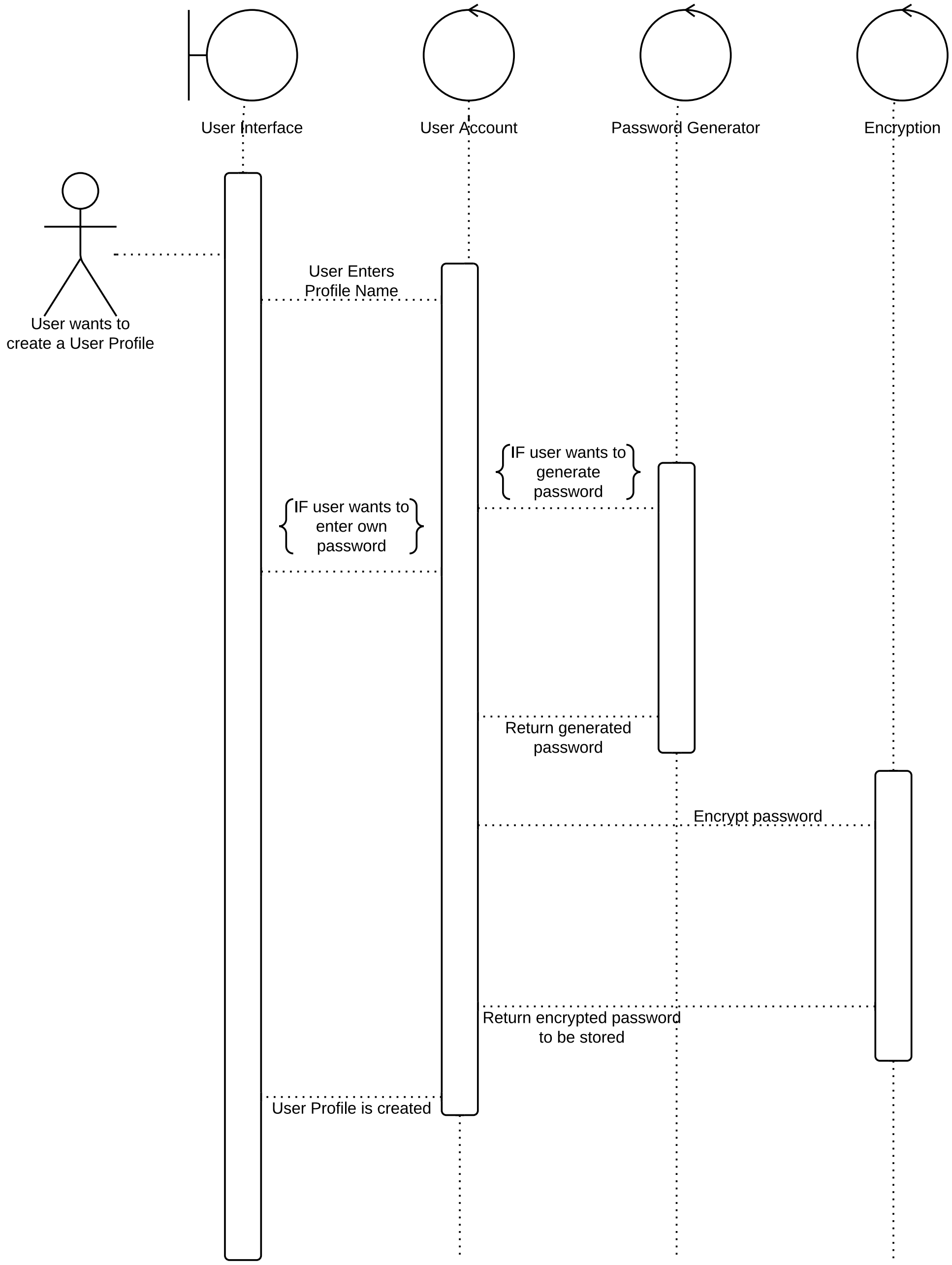
Elaboration of Design

Part 1: Design Class Diagram

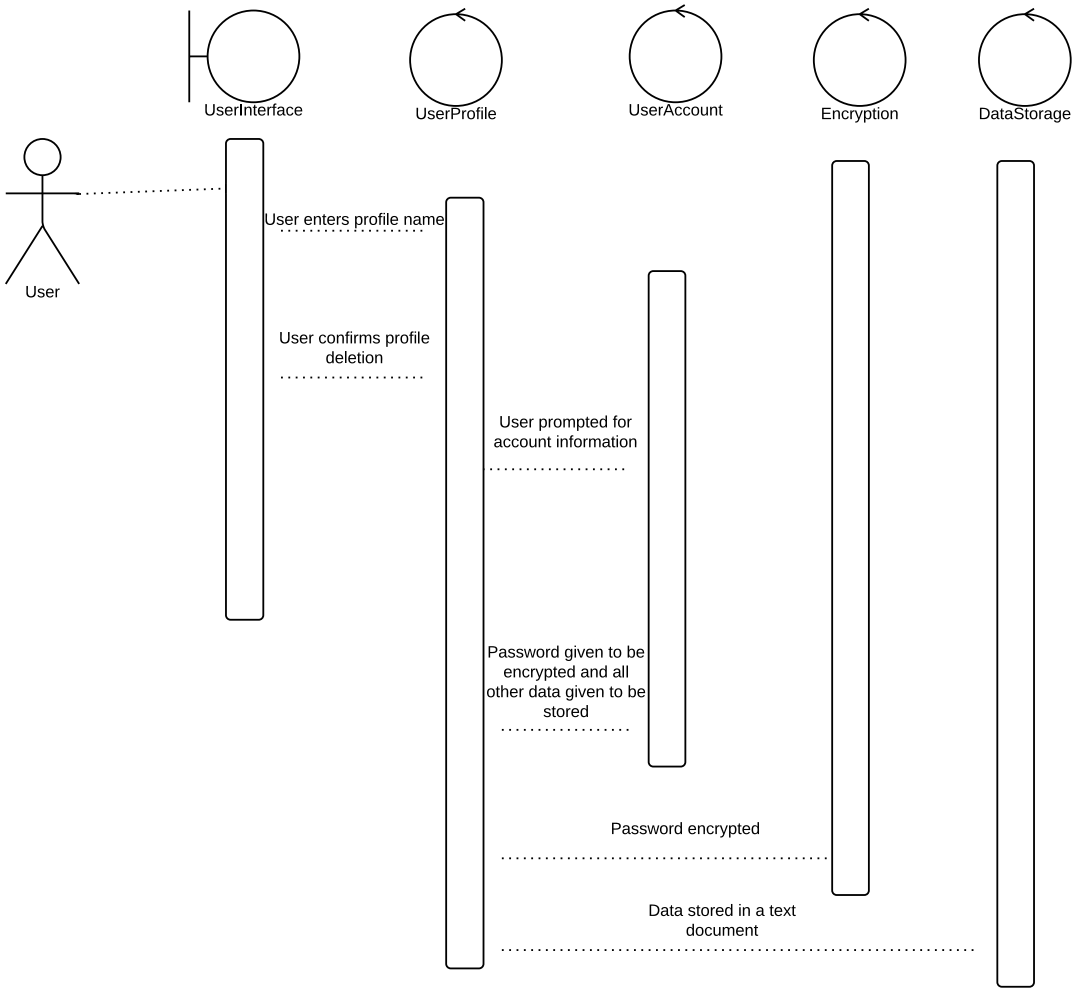


Part 2: System Sequence Diagrams

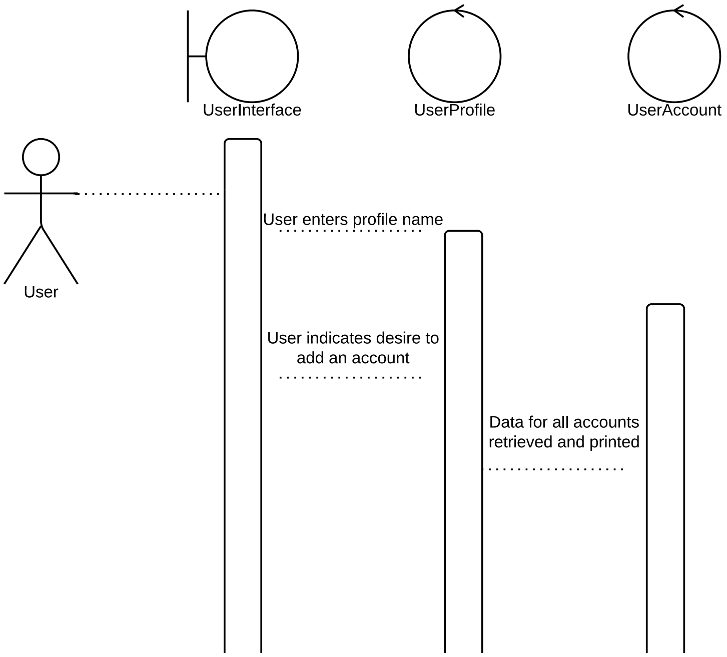
New User Creates New User Profile


User Interface
User Account
Password Generator
Encryption
User wants to create a User Profile
User Enters Profile Name
IF user wants to generate password
IF user wants to enter own password
Return generated password
Encrypt password
Return encrypted password to be stored
User Profile is created

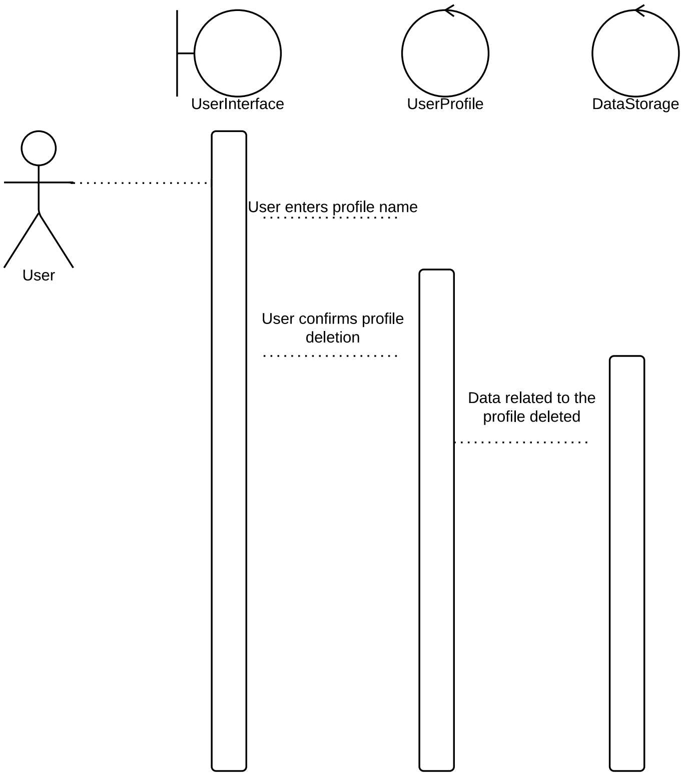
Existing User Adds an Account from System Ready State


UserInterface
UserProfile
UserAccount
Encryption
DataStorage
User
User enters profile name
User confirms profile deletion
User prompted for account information
Password given to be encrypted and all other data given to be stored
Password encrypted
Data stored in a text document

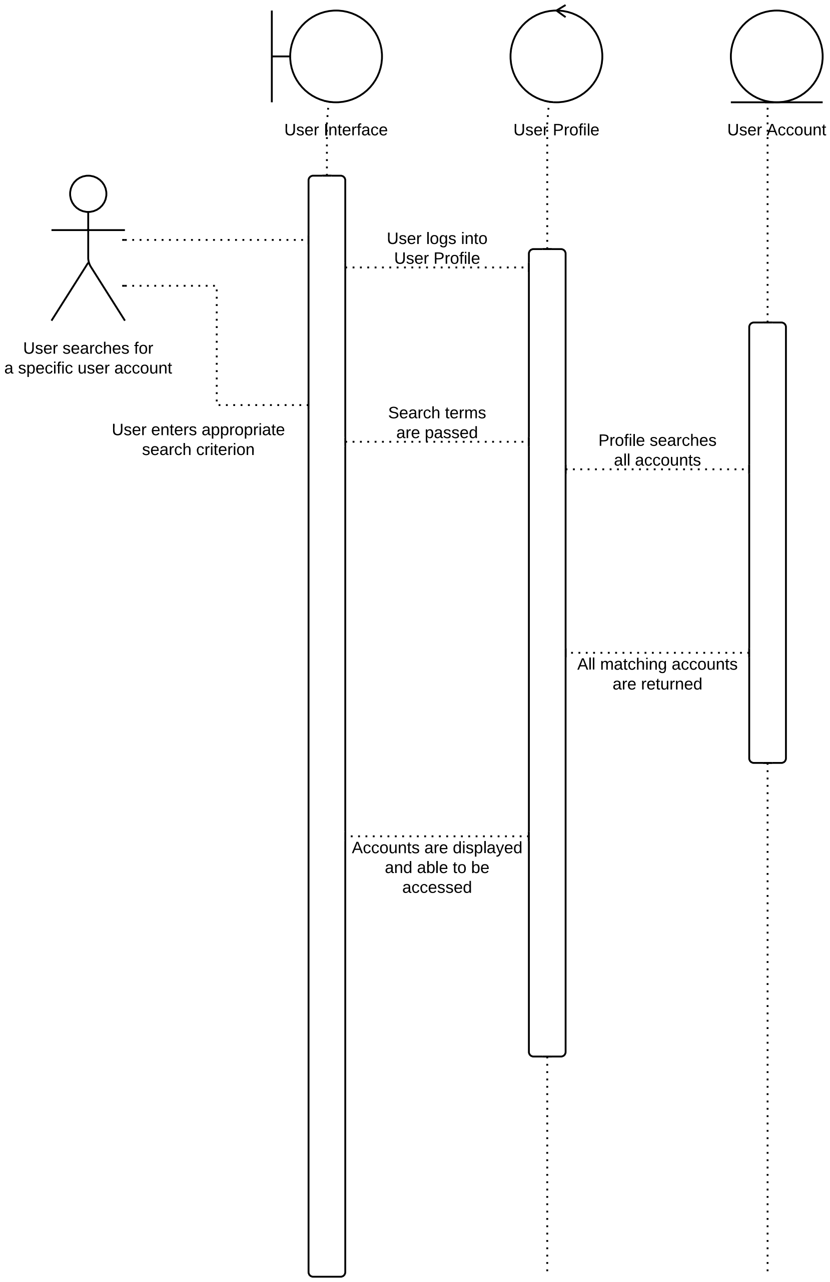
Existing User Selects the Option to Print All Data


UserInterface
UserProfile
UserAccount
User
User enters profile name
User indicates desire to add an account
Data for all accounts retrieved and printed

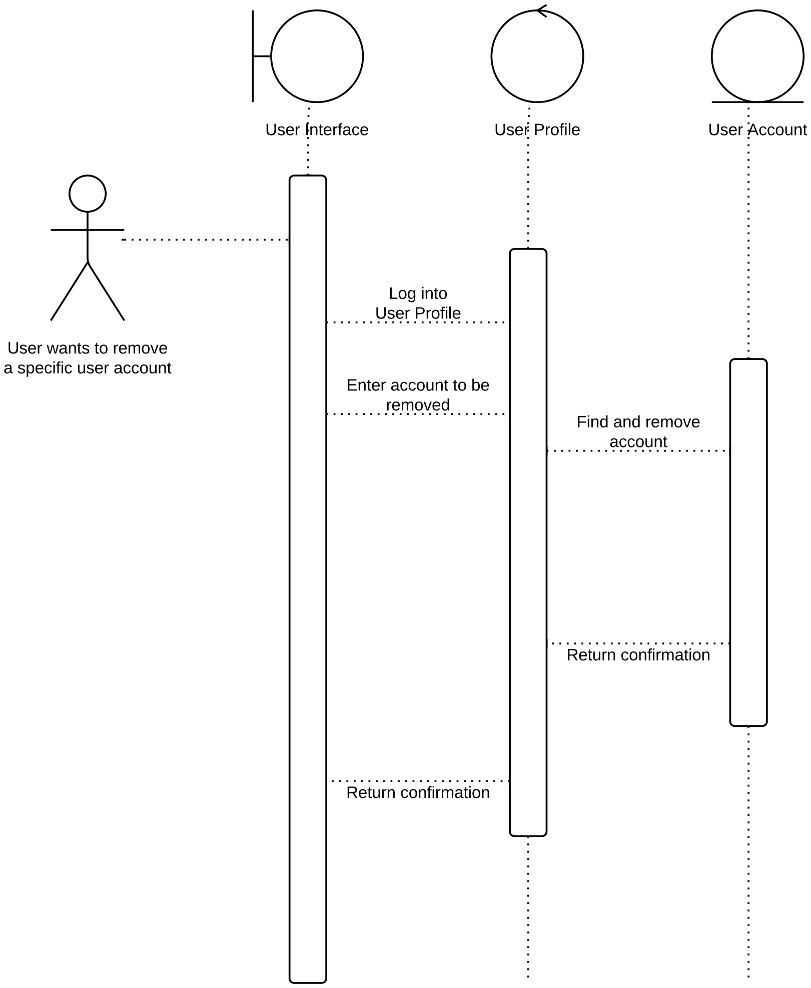
Remove User Profile


UserInterface
UserProfile
DataStorage
User
User enters profile name
User confirms profile deletion
Data related to the profile deleted

User Searches for a Specific Account


User Interface
User Profile
User Account
User searches for a specific user account
User logs into User Profile
User enters appropriate search criterion
Search terms are passed
Profile searches all accounts
All matching accounts are returned
Accounts are displayed and able to be accessed

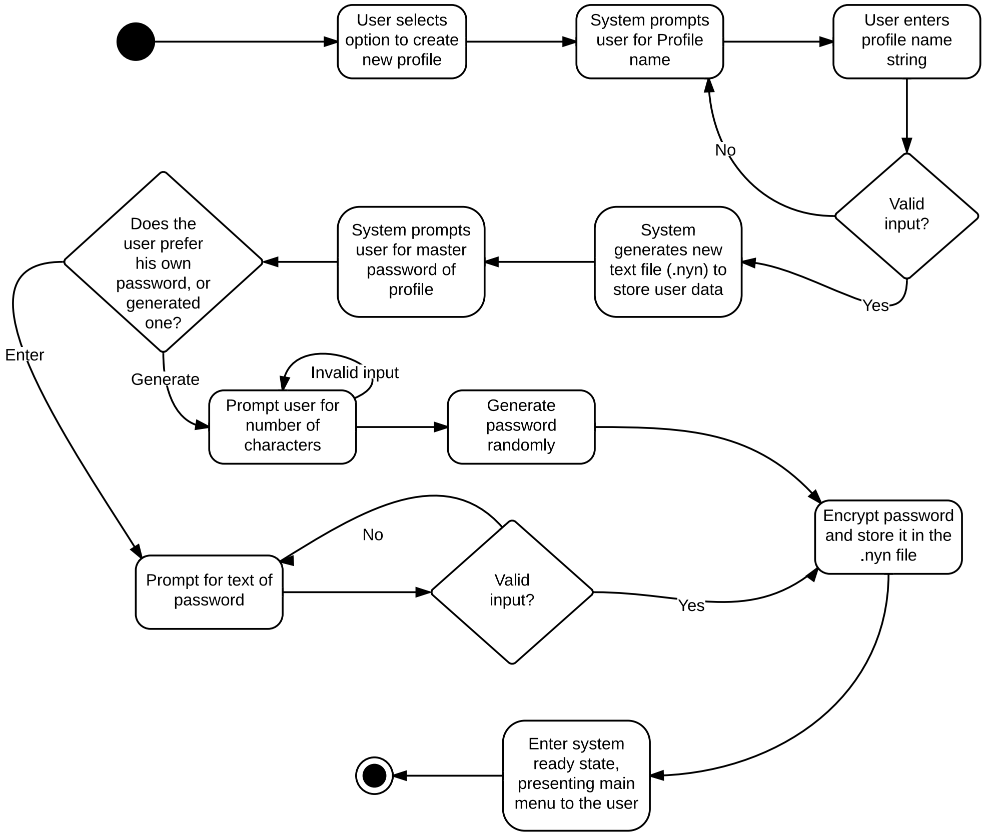
User Decides to Remove an Account



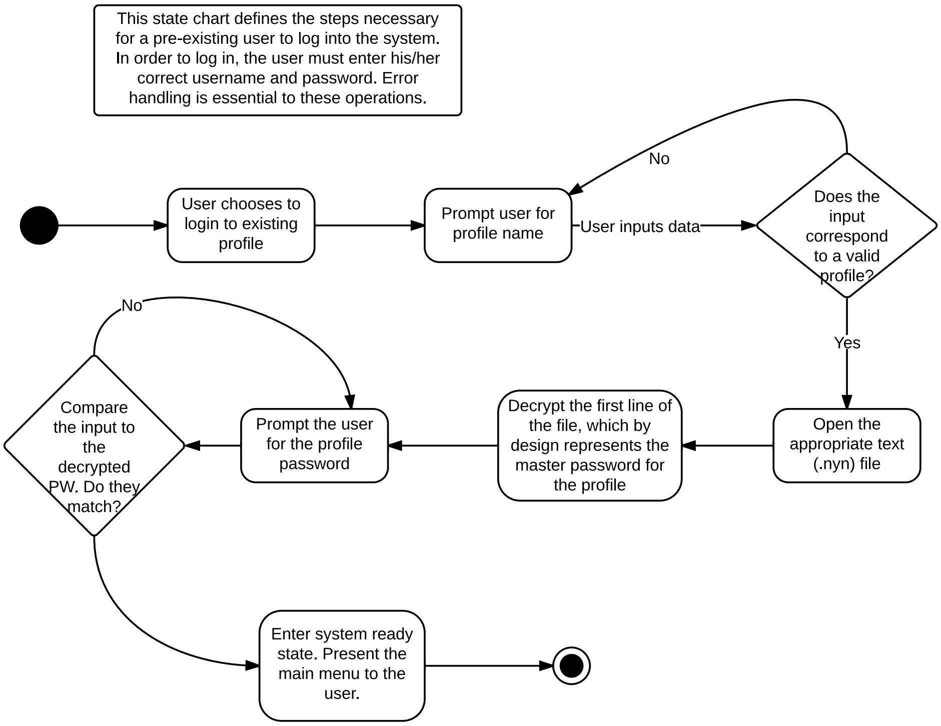
Part 3: Detailed System State Charts

New User Creates a User Profile

This state chart describes the steps necessary to create a new User Profile for a new User. The user must define what his master username and password will be. This profile will store all the data that the user wishes to enter into the system in a single text file.


User selects option to create new profile
System prompts user for Profile name
User enters profile name string
Valid input?
No
Yes
System generates new text file (.nyn) to store user data
System prompts user for master password of profile
Does the user prefer his own password, or generated one?
Enter
Generate
Invalid input
Prompt user for number of characters
Generate password randomly
Encrypt password and store it in the .nyn file
Prompt for text of password
No
Valid input?
Yes
Enter system ready state, presenting main menu to the user

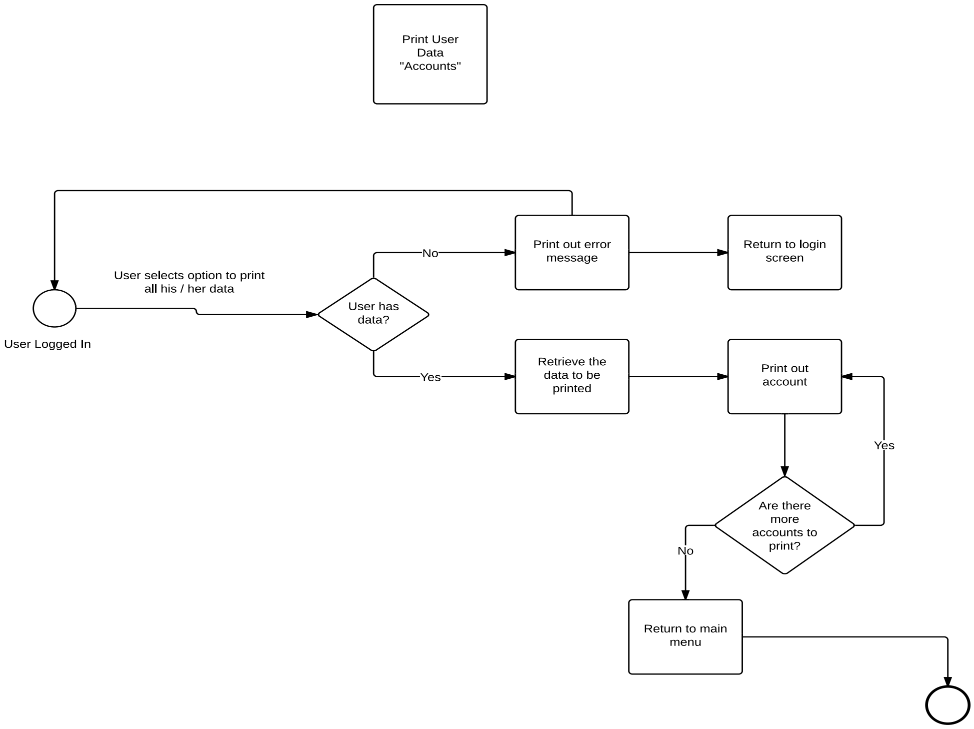
Existing User Logs into System Ready State



This state chart defines the steps necessary for a pre-existing user to log into the system. In order to log in, the user must enter his/her correct username and password. Error handling is essential to these operations.

User chooses to login to existing profile
Prompt user for profile name
User inputs data
Does the input correspond to a valid profile?
No
Yes
Open the appropriate text (.nyn) file
Decrypt the first line of the file, which by design represents the master password for the profile
Prompt the user for the profile password
Compare the input to the decrypted PW. Do they match?
No
Enter system ready state. Present the main menu to the user.

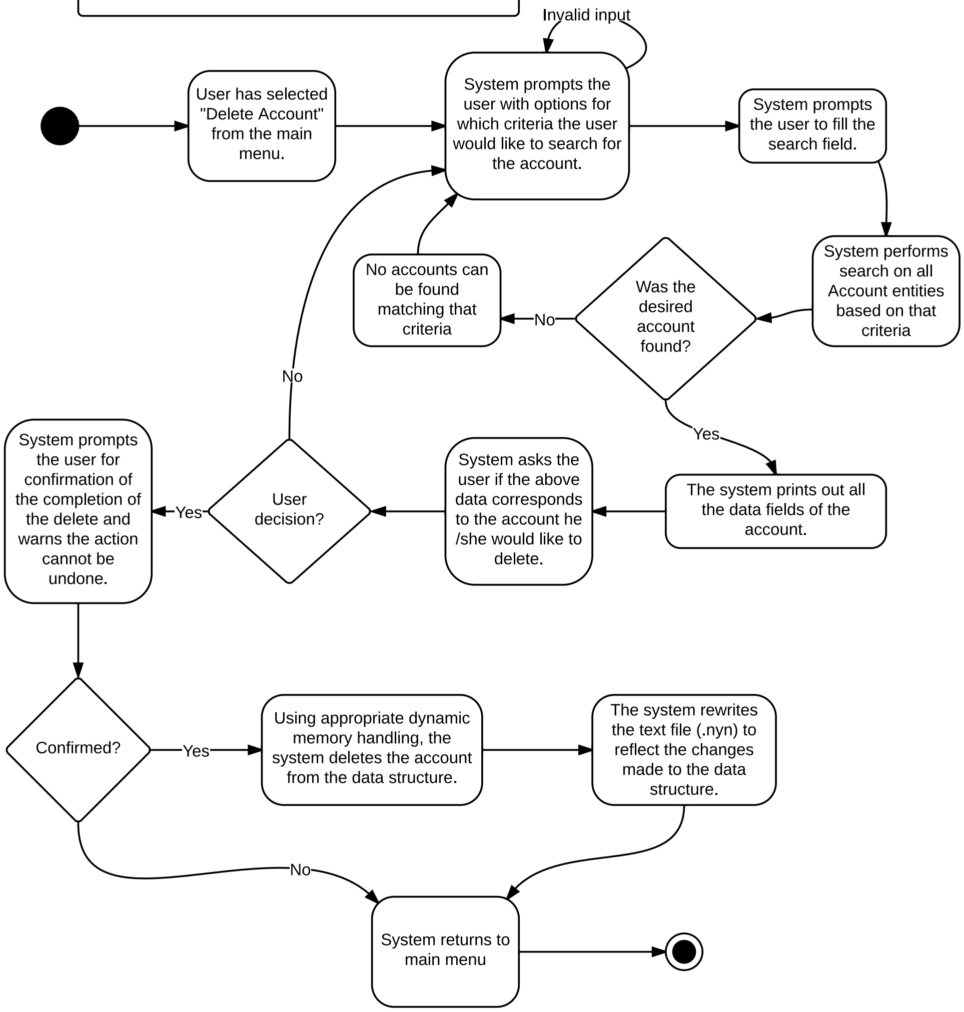
User Decides to Print All Data



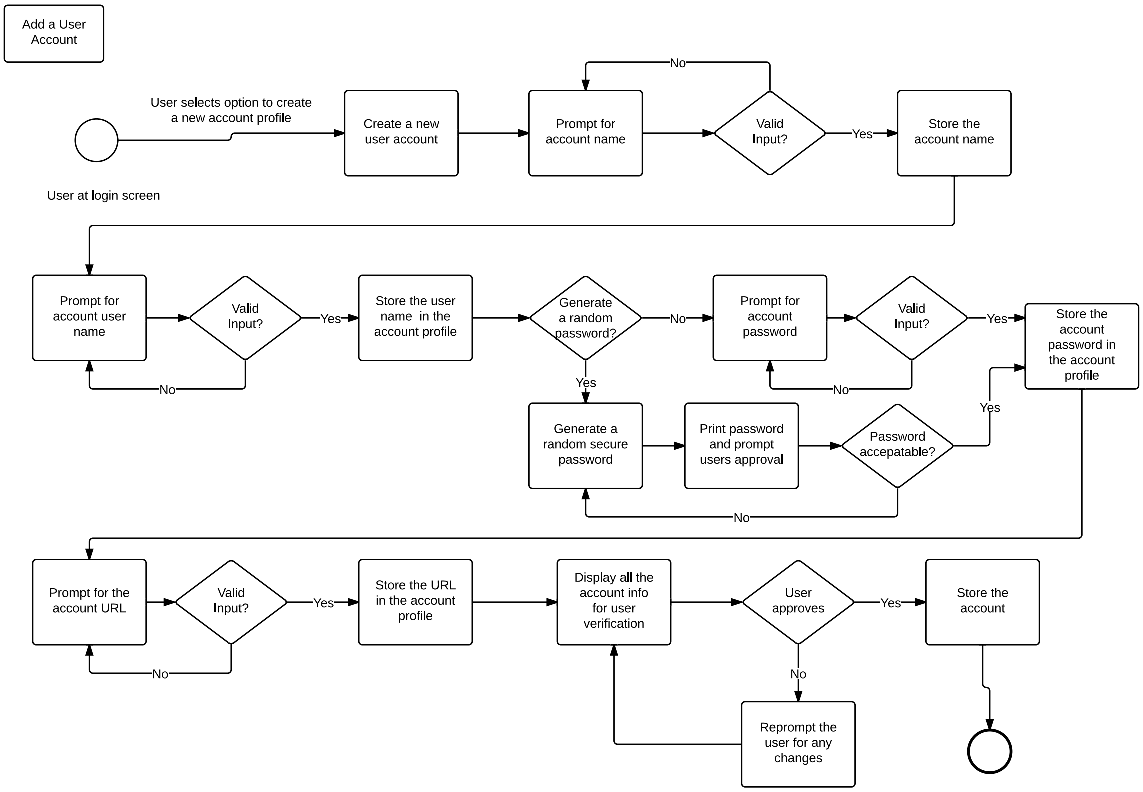
Print User Data "Accounts"
User Logged In
User selects option to print all his / her data
User has data?
No
Print out error message
Return to login screen
Yes
Retrieve the data to be printed
Print out account
Are there more accounts to print?
Yes
No
Return to main menu

Existing User Decides to Remove an Account

This chart defines the steps necessary to delete a user account from the system and the User Profile. In order to determine which account the user would like to delete, the user must search for the specific account to be removed. After finding the proper account, the system should prompt for confirmation.

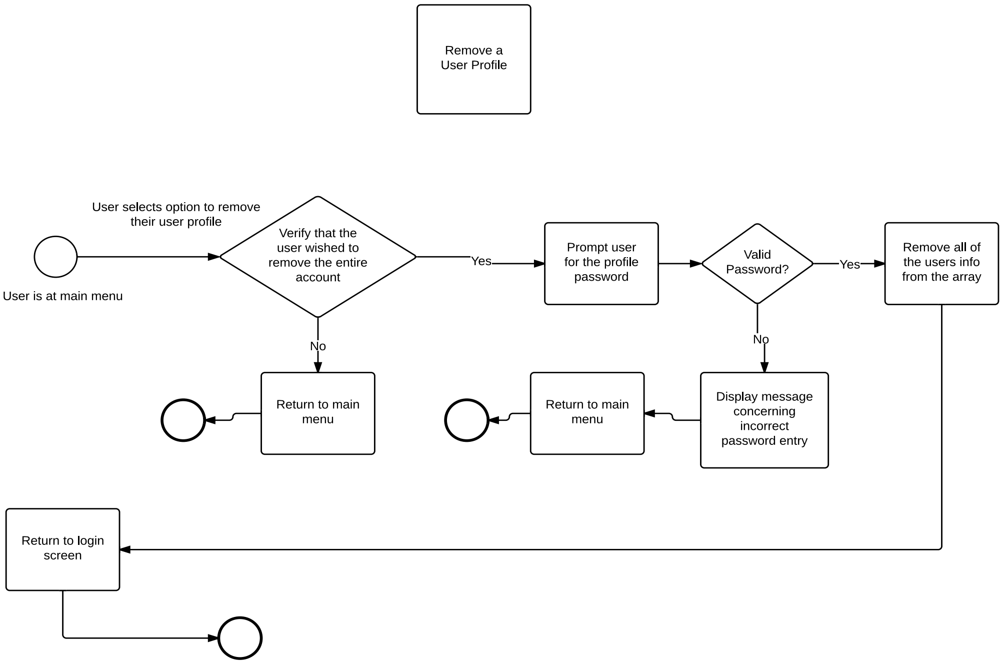

Invalid input
User has selected "Delete Account" from the main menu.
System prompts the user with options for which criteria the user would like to search for the account.
System prompts the user to fill the search field.
System performs search on all Account entities based on that criteria
Was the desired account found?
No
No accounts can be found matching that criteria
Yes
The system prints out all the data fields of the account.
System asks the user if the above data corresponds to the account he /she would like to delete.
User decision?
No
Yes
System prompts the user for confirmation of the completion of the delete and warns the action cannot be undone.
Confirmed?
Yes
Using appropriate dynamic memory handling, the system deletes the account from the data structure.
The system rewrites the text file (.nyn) to reflect the changes made to the data structure.
No
System returns to main menu

Existing User Decides to Add a New Account



Add a User Account
User selects option to create a new account profile
User at login screen
Create a new user account
Prompt for account name
No
Valid Input?
Yes
Store the account name
Prompt for account user name
Valid Input?
Yes
No
Store the user name in the account profile
Generate a random password?
No
Yes
Prompt for account password
Valid Input?
Yes
No
Store the account password in the account profile
Generate a random secure password
Print password and prompt users approval
Password accepatable?
Yes
No
Prompt for the account URL
Valid Input?
Yes
No
Store the URL in the account profile
Display all the account info for user verification
User approves
Yes
No
Store the account
Reprompt the user for any changes

Existing User Removes User Profile (All User Data)



Remove a User Profile
User selects option to remove their user profile
User is at main menu
Verify that the user wished to remove the entire account
Yes
No
Return to main menu
Prompt user for the profile password
Valid Password?
Yes
No
Remove all of the users info from the array
Display message concerning incorrect password entry
Return to main menu
Return to login screen

Part 4: User Interface Design

Our user interface design will be completely text based implemented via CLI and will utilize all of the eight golden rules for designing interfaces. Due to CLI limitations, we must prompt the user for each individual input for each action. After the user logs into the system ready state, either via log in or profile creation, the command line will present several functionality options, each with a corresponding input token. In order to access a particular functionality, the user must enter its input token, which will be of the char datatype. Each submenu will list a back button to access the previous menu, thereby undoing the most recent action by the user.

1. Strive for consistency
    a. The interface will appear the same for every use. There will be a list of possible functions available to the user on startup. For each version change, we intend for the user interface to remain the same to not confuse users.
2. Enable frequent users to use shortcuts
    a. The program will just have the shortcuts designated to access all of the functions. Additional advanced shortcuts will not be available.
3. Offer informative feedback
    a. The program will indicate error messages when things go wrong. It will explain what is acceptable input and how to operate the program. After completing a task, the program will demonstrate that the task has been completed.
4. Design dialogs to yield closure
    a. When a task is completed, the program will say that the task is completed. For example, if a user adds an account, the program will display a message saying that the account has been added.
5. Offer simple error handling
    a. The program will check for all invalid input and make sure the user knows what valid input includes. It will allow users to re-enter data when an error occurs or otherwise when necessary.
6. Permit easy reversal of actions
    a. The program will not exactly have reversal of actions, however it will double check to confirm that the user really wants to perform an action.
7. Support internal locus of control
    a. Every action taken by the program will be directly under the control of the user through the use of text-based commands. The program outputs verification of the user's actions.
8. Reduce short-term memory load
    a. The program accomplishes this by listing all of the commands available and the hotkey required to use them. It will explain how to enter data correctly and will also display a list of accounts that the user owns.