



Account Management

**Kshitiz
Prakash**



25Bce10268

Overview

This is a console based python application which allows users to sign up, log in, store notes, view registered accounts, and delete accounts - all in single runtime session. It stores data temporarily.

Features

- Sign up - Create a user account with username, age, gender and password validation
- Log in - Authenticate user with username and password
- Notepad - Store user's note inside the system
- Change password - Update password while logged in
- List accounts - Display all registered users along with their info
- Delete account - Remove an account using correct login credentials
- Exit - Close the program

Functional Requirements

User account creation -

System shall validate that

- Age must be greater than zero
- Gender must be either M or F
- Password credentials should match

User login -

Authentic username and password and raise error in case of authentication failure.

Notepad access -

System shall allow users

- To view their existing notes
- To append new notes

Account listing -

System shall display

- All registered accounts
- Only non-sensitive information

Non-functional Requirements

Performance requirements -

System shall validate that

- The system should respond to user input within 1-2 seconds
- The system should handle multiple user accounts efficiently in memory
- Password credentials should match

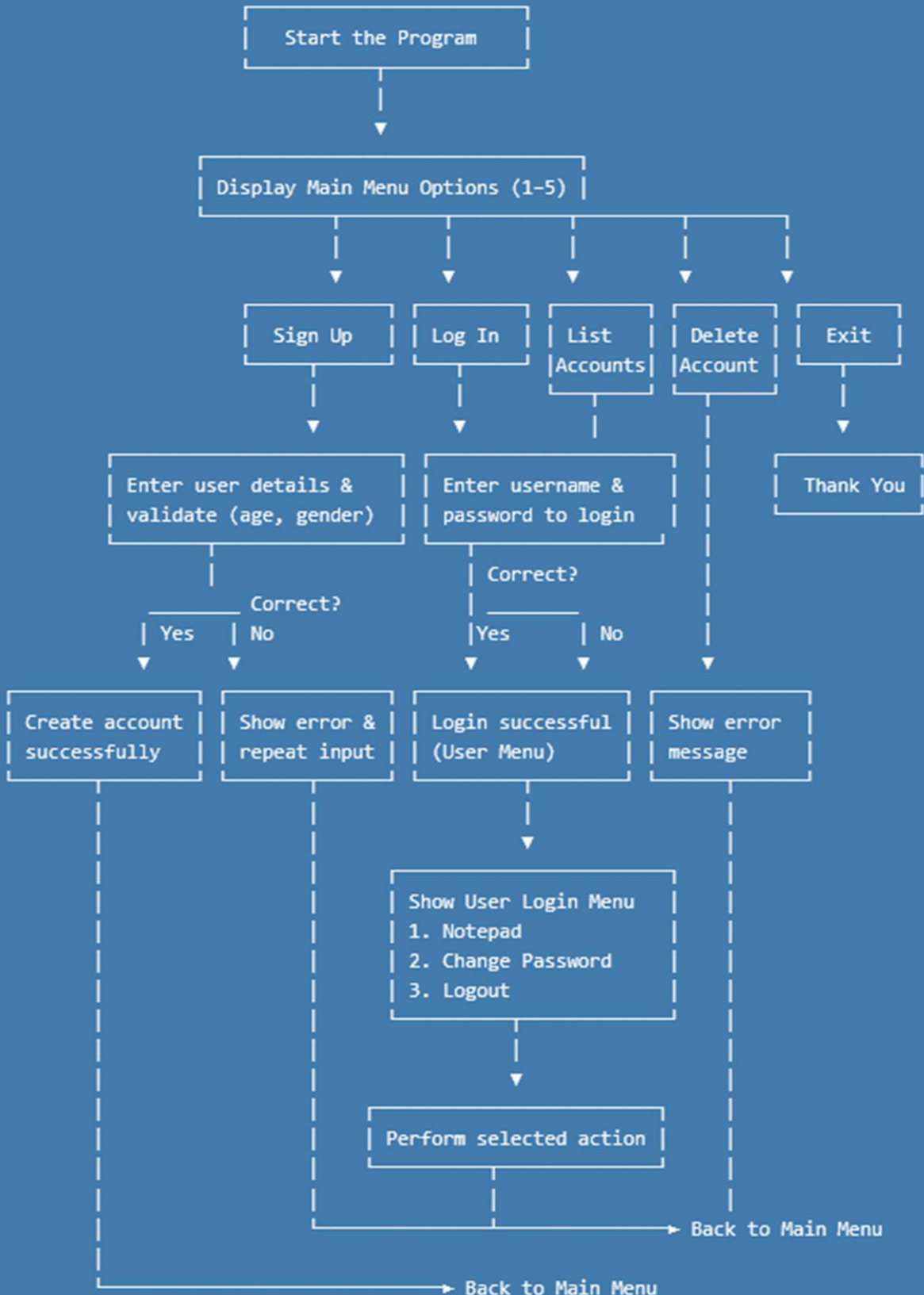
Usability requirements -

- The system should provide clear prompts and step-by-step input instructions
- To append new notes

Security Requirements -

- Password verification is required during login and account deletion
- Sensitive data like passwords are not displayed during account listing

Workflow



```

u=input('username          - ')
ac=0
while ac!=1:
    a=int(input('age          - '))
    if a>0:
        ac=1
    else :
        print('invalid input, try again')
gc=0
while gc!=1:
    g=input('gender (M/F)      - ')
    if g=='M' or g=='F':
        gc=1
    else:
        print('invalid input, try again')

```

```

b41u4f(,7u4a0j7q 7ubn4f' 4u4 0807u.)
6j26:
8c=1

```

```

if s=='1':
    u=input('username          - ')
    ac=0
    while ac!=1:
        a=int(input('age          - '))
        if a>0:
            ac=1
        else :
            print('invalid input, try again')

    gc=0
    while gc!=1:
        g=input('gender (M/F)      - ')
        if g=='M' or g=='F':
            gc=1
        else :
            print('invalid input, try again')

    p=input('password          - ')
    c=input('confirm password - ')
    n=''
    print()

```

```

while gc!=1:
    g=input('gender (M/F)      - ')
    if g=='M' or g=='F':
        gc=1
    else:
        print('invalid input, try again')

p=input('password          - ')
c=input('confirm password - ')
l=[]
n=''
print()
if p==c:
    l=[[u,p,n,g,a]]
    print()
    print('account created successfully')

```

Implementation

- ❑ Ensure `python 3.x` is installed
- ❑ Run the script in `terminal`
- ❑ Follow the `on-screen` menu instructions

Testing approach

- ☐ Sign in a new account
- ☐ Log in to that account
- ☐ Use attached notepad
- ☐ Change password
- ☐ Log out
- ☐ Sign in another account
- ☐ See all registered accounts
- ☐ Exit

Future enhancements

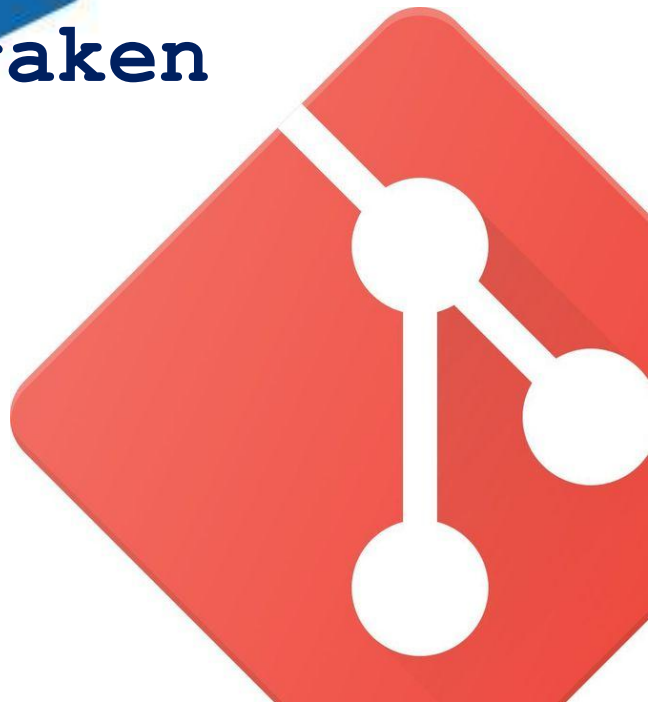
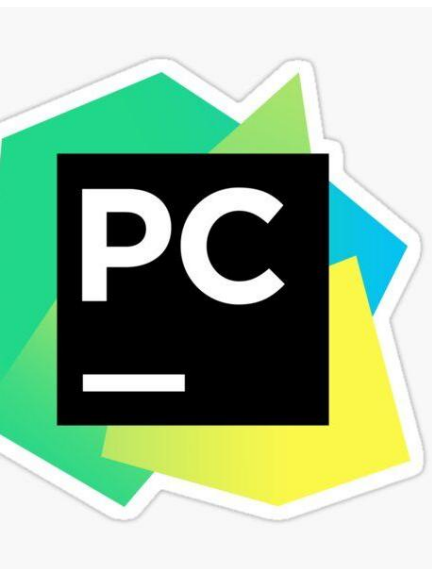
- ❑ Employing database
- ❑ Importing mysql
- ❑ Masking password
- ❑ File handling for permanent storage of information of notepad
- ❑ Replacing with GUI



Technologies used



- ☐ Python
- ☐ Pycharm
- ☐ VS code
- ☐ Git
- ☐ Github
- ☐ GitKraken



Thank you _