**Software Design (International Year 1)**

**XFX1041-1920 Software Design**

Coursework Title **Authentication Manager**

Problem Description

The authentication manager system to be developed allows the user to login into the system by authenticating the user name and password against a list of registered members saved on a file. The system will be able to determine if the login was successful or failed based on the accurate result of the login check. The authentication manager will also allow users to register and add to the list of registered members on the file system. Finally the system will permit the user to reset his password should he choose to do so while logging in using a ‘forgot password’ button. While registering, the user must also provide his mother’s maiden name and the system must store this information that will be required in order for the user to change their password.

Class Diagram

**Class diagram for AuthMgrGui class**

|  |
| --- |
| **AuthMgrGui** |
| -mgr : Manager |
| +Display() |

**Class diagram for Manager class**

|  |
| --- |
| **Manager** |
| -gui : AuthMgrGui  -session : Session |
| +login(user:Details) : void  +Register(user:Details) : void  +requestReset(user:Details): void  +ResetPassword(user:Details) : void |

**Class diagram for Session class**

|  |
| --- |
| **Session** |
| -user : Details  -status : Status  -errors : Error |
| +getStatus() : Status  +setStatus(status:Status)  +ValidateEmail() : void  +ValidatePassword() : void  +ValidateUser() : void  +setUser(user:Details) : void  +createUser(user:Details) : void  +setError(errors:Error) : String  +getError() : Error |

**Class diagram for Error class**

|  |
| --- |
| **Error** |
| -code : int  -message : String |
| +getError() : int  +setError(code:int) : void  +getMessage() : string  +setMessage(msg:string) : void |

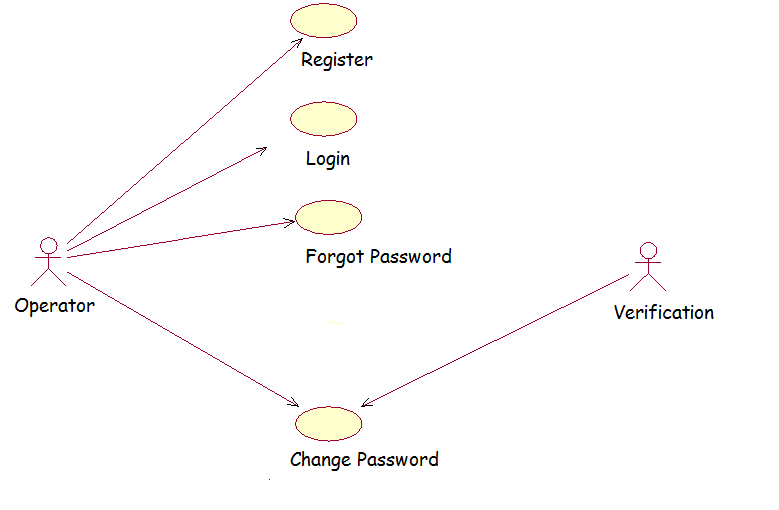
**Class diagram for Status class**

|  |
| --- |
| **Status** |
| -code : int  -message : String |
| +setStatus(code:int) : void  +setMessage(status:string) : void  +getStatus() : int  +getMessage() : string |

**Class diagram for Details class**

|  |
| --- |
| **Details** |
| -UserEmail : string  -UserPassword : string  -MotherMaidenName :string |

Use Case Analysis to determine system requirements



Activity Diagram



Verify Password

Forgot Password

Select Option

Invalid User

Valid User

Authentication

Login

Check User

Change Password

Sequence Diagram

FILE

DETAILS

USER

Register

Fill

Submit

Save

Username and password entered

Check

valid

Login Successful

Algorithm

**Login Algorithm**

Step 1: Start.

Step 2: Enter username and password.

Step 3: If username and password matches then:

Step 4: Login Successful.

Step 5: else: Login Failed

Step 6: Stop

**Registration Algorithm**

Step 1: Start

Step 2: Choose Register option if you are not yet registered.

Step 3: Enter User Email

Step 4: Enter User Password

Step 5: Enter User Mother Maiden Name

Step 6: Save in File

Step 7: Stop

Flow Chart

Enter User Name and Password

Login Successful

Is User Name and Password Correct

False

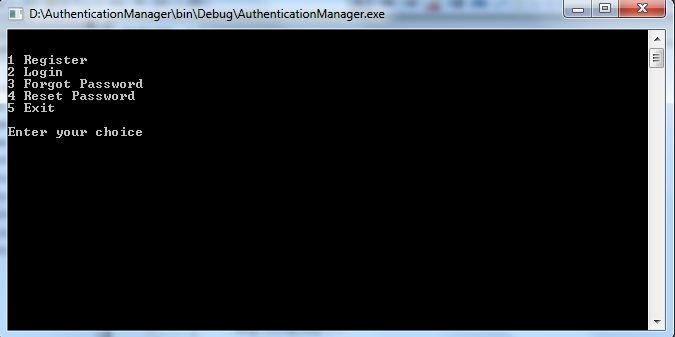
True

Testing

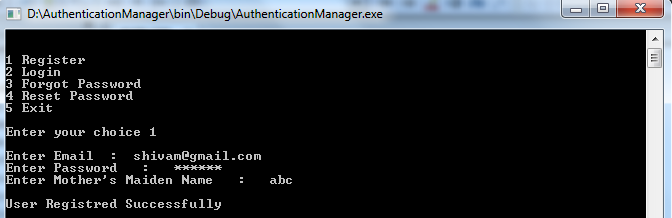
|  |  |
| --- | --- |
| **Test Cases** | **Description** |
| **Test case-1:Login** | User must enter username and password to login. |
| **Test case-2:Registration** | All the fields must be filled by the user during their registration. |
| **Test case-3:Email id** | The duplicate email Id is not allowed. |
| **Test case-4:Password** | Password must be greater than 6 characters |
| **Test case-5:Mother Maiden Name** | Password must be changed after matching user mother maiden name. |

**Output**

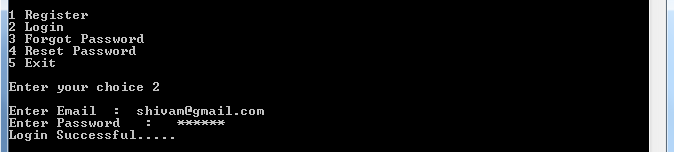
Front Screen



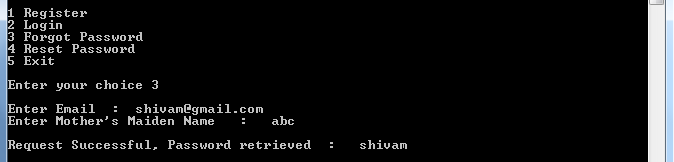
Registration



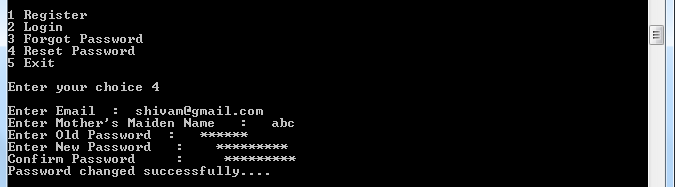
Login



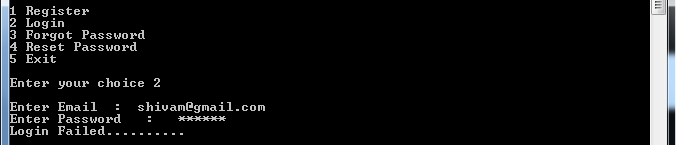
Forgot Password



Reset Password



Login Failed



Duplicate Email Id

