Design Document

By Hridya Krishna R

10th September 2019

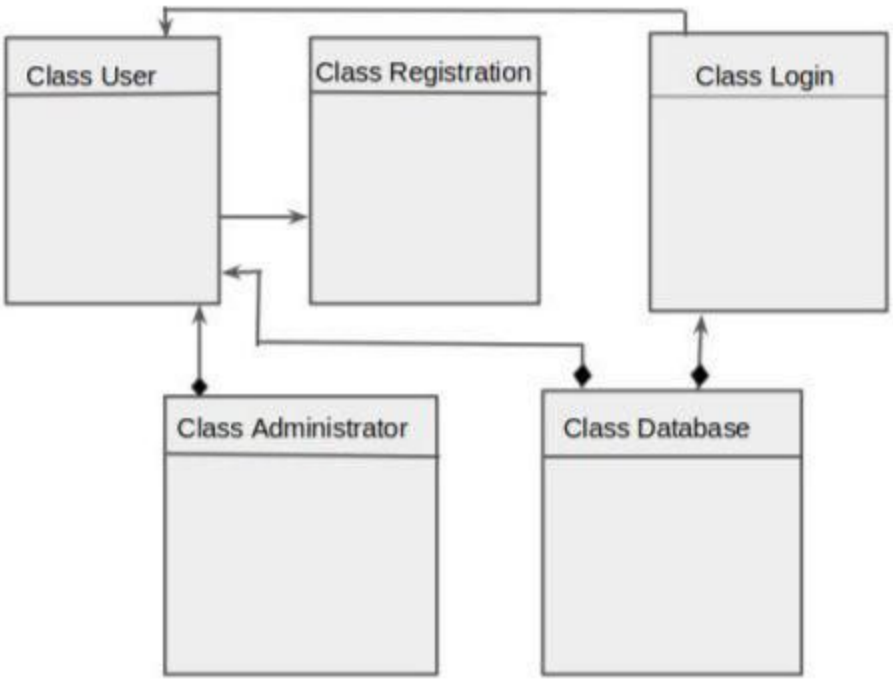
Contents

Introduction

A complete software is deployed only when it is safe and secure. This increases the user-engagement of the product. A secure login page for the user to register themselves with the information being stored in a highly secured secure database would make the users feel comfortable. This guarantees the privacy of a user and the security of the accessible data (as permitted by the user). Registration process helps a person to create account based on the eligibility and a user login page helps an individual to authenticate themselves to gain access.

Product Perspective

Class Registration
Class User
Class Login
Class Administrator
Class Database



Product Function

This project will include the following methods(with their functions) implemented:

1. **login()** : to proceed into the account after authentication.
2. **getRegistered()** : to feed the details of new user and store it into database.
3. **checkCredentials()** : Check the credentials entered during login with the saved credentials.
4. **changePassword()** : to update the old password with a fresh one.
5. **goodPassword()** : to check the password:
    a. Strength - Password should have more than 8

characters and at least 2 digit and 1 special character(!,@,#,$,%,&,*).

i. Very strong: More than 12 characters with at least 3 digits.
ii. Strong: More than 10 characters with at least 2 digits.
iii. Weak: More than 8 characters with at least 2 digits.

b. Not random- Shouldn't be random figures like '123', 'abc', 'ABC', 'QWERTY123' , the name of the user, etc.

6. **resetPassword()** : sends reset password link to the registered email which grants permission to change the password and login with the fresh one.(Forgot password).
7. **rememberMe()** : Makes a user logged in until the user clicks log out, ie, even if the application is closed.
8. **emailVerification()** : sends verification link to the registered email during registration and proceed to the user account only after clicking confirmation through the link.
9. **numberVerification()** : If the user selects mobile verification instead of email verification, a text message with the verification link is sent to the registered mobile number. (Given as an option to the user).
10. **singleSignOn()** : to allow login through third party services

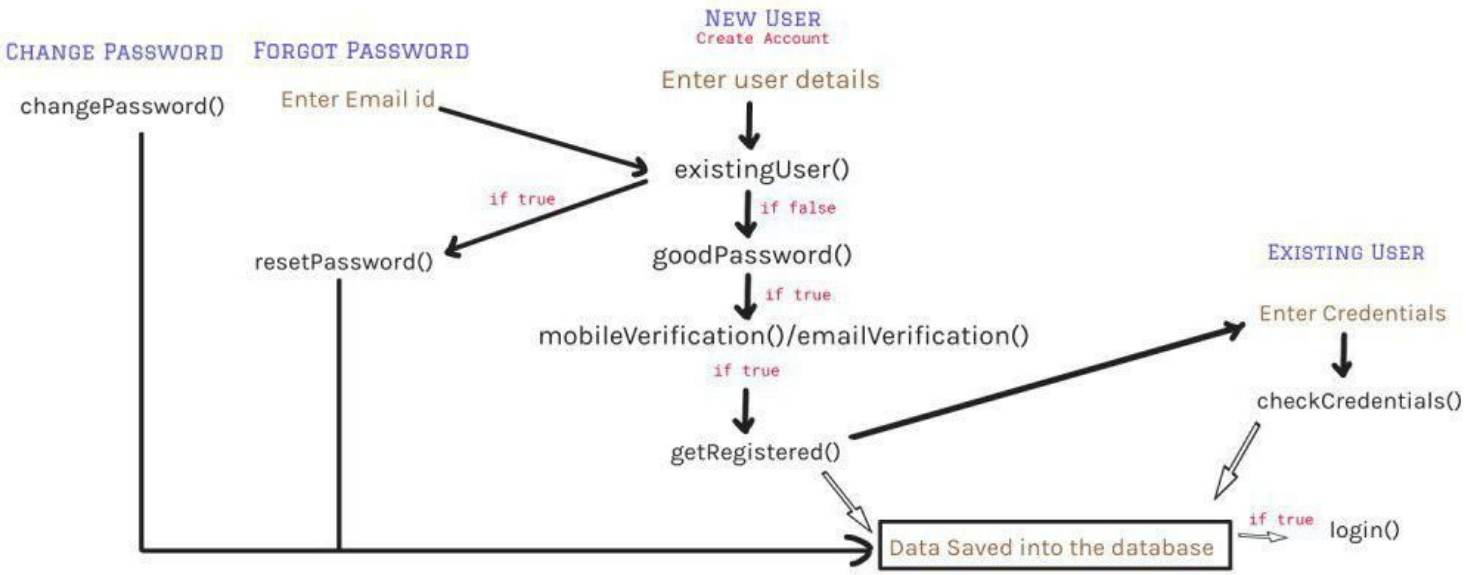
such as Google, Facebook , Github etc.

11. **existingUser()** : Checks if there is already an existing account with the email id provided.

Goals and Assertions

A login page is essential to any application as it provides security and privacy to the user. It makes sure that a person has access to the details or resource of the application. Another important key is that the application will be user-friendly. It will be easily manageable by the user with right authentication. The database will be well maintained and highly secure. The login page also notifies the user, the strength of the password during registration and also prevents unsafe passwords(random guessable passwords). For flexibility, forgot-password option provides email as well as mobile number means of reset.

Flowchart


CHANGE PASSWORD
changePassword()
FORGOT PASSWORD
Enter Email id
resetPassword()
if true
NEW USER
Create Account
Enter user details
existingUser()
if false
goodPassword()
if true
mobileVerification()/emailVerification()
if true
getRegistered()
EXISTING USER
Enter Credentials
checkCredentials()
Data Saved into the database
if true
login()