

# Explanations

## Design Document - Password reset

By Isaac Murisa 201534328

The functional feature is to allow the user to reset his/her password. On sign up a user needs to enter a security question and provide an answer to the question. The question will be used as a security feature to allow the user to enter a new password. The feature will use the USERS database table to retrieve and as well set a new password. The desired sequence of pages is as follows (inorder):

- Forgot Password?
- Enter User Name
- Answer The Question
- Reset Password
- Back to Login

If stage number 2 fails the users is prompted to retry his/her username. If stage number 3 fails the user needs to provide the username first therefore he is taken to the previous page.

The feature is going to use three html files which are `security.html`, `question.html` and `enter_password.html`. Both files will extend the homepage. `security.html` file is the page where the user is prompted to enter a username. The `question.html` file is the page where the user has to provide an answer to a question. Lastly, the `enter_password.html` file is the page where the user will enter a new password.

All the above files are passed to flask through functions or methods inside the `flaskr.py` file. The methods are:

- **security()** - the function can handle a GET or POST request. A GET request is sent when the user clicks on `forgot password` and then it sends back the page prompting the user for username. A POST request is sent by the browser when the user enters a username which is then used to retrieve the specific information from the database. The information will be retrieved using an interface module defined as `passwordReset.py`. The module contains a method called `getUserInfo(username)` and another one to be used later called `setPassword(newPassword, Username)`. The user name provided is then passed to the `getUserInfo(username)` as a parameter. The function is then going to return the data in the database entry for the user name provided.
- **question()** - the method handles a POST request and in the form of an answer. When a user provides a correct username the `question.html` file is rendered with the user's security question. When the user answers the question it is sent to the to this method. Again the `getUserInfo()` method in the interface module is called again this time to return data but targeting the answer set in the user's database. When the answer is returned it is sent to the `enter_password.html` file where it is compared with the provided answer by the user. When the answers match the user is prompted to enter a new password, else he informed to retry.
- **newPassword()** - this is the final method of the feature. It gets the new password provided by user and then sends it to the `setPassword(newPassword, username)` method in the interface module to

be updated in the database. When done the method returns the a redirect command to the login page.

The interface `passwordReset.py` is there as well to provide a clear clarity and persistance between the feature and the database as well as the main file. The feature works fine and as expected. It was verified based on the requirement.