Preface

After reading through your team's use case descriptions in relation to the project requirements file provided by the instructor, I must say that I am genuinely amazed by the level of details and planning your team has put into this project. Without a doubt, I can say that your team goes above and beyond in some aspects of the project requirements. For example, the usage of OTP is certainly something that may not be "required" as per say but your use case diagrams file has it as a feature. The level of details behind each use case is also commendable as the interdependence of use cases is clearly and correctly shown in your use case diagram. With all this being said there are still two features that I propose as they can be interpreted to be part of the project requirements.

Feature Request: Delete Account From Database & System

High Level Description:

Use Case: User Account Deletion

Iteration: 1

Primary Actor: User

Goal in Context: Allow a user to delete an account from the database and the system by providing necessary credentials for verification.

Preconditions:

• The user device is powered on.

• The user is logged into their account.

• The user navigates to the profile management/account management page.

• The website is operating without any technical issues.

• The user has a valid email account

Triggers:

• The user selects the 'Delete Account'(Formatted In Red Font) option from the profile management page. .

Scenario:

1. The user selects the "Delete Account" option on the profile management page. .
2. The system prompts the user to confirm if they actually want to confirm with the confirm not being the default option.
3. The user selects confirm.
4. The system prompts the user to input their current password and their associated email account.

5. The system checks if the password and email match what is stored in the database from when the user registered the account.

6. If valid, the system sends an account deletion verification email with a one-time password.

7. Otherwise, users will have to re-enter their password and email until it matches what is stored in the database.

8. The user enters the one-time password into the given prompted textbook.

9. The system deletes all data related to the user from the database.

10. The user is redirected to the sign-in page and they receive an email saying their account has been deleted.

Postconditions:

• The user successfully deletes their account and related data from the database.

• The login credentials(username) are now available to use for new registrations.

• The user cannot login again with the deleted credentials.

Exception:

• The inputted password doesn't match the database password.

• The inputted email doesn't match what is in the database.

• The inputted one time password doesn't match what the system sent out.

Priority: Medium (Dead Accounts and Banned Accounts should be removed from the system as to conserve resources)

When Available: Always accessible on the profile management page.

Frequency of Use: Once per user account.

Channel to Actors: Web browser, system UI, email access.

Secondary Actors: System Administrator.

Channel to Secondary Actors: Database Management/(Auth Manager)

Open Issues:
• Possibly recovering the account after it gets deleted from the database.

Expected Impacts:

The implementation of this design will mostly rely on the authentication and profile management division as they will simply have to incorporate ways to remove data from the database as opposed to the current ways of simply adding data or replacing data(resetting passwords). In implementing this feature, the GUI division will also have to add this feature to their interface. Other than these two minimal changes there isn't any other major changes that may drastically impact the system.

Suggestions for Implementation:

- Add a method named deleteAccount() under the ProfileManager section of the class diagram of the authentication team.
    - This method can have other sub/helper methods like getEmail() and verifyEmail(), getPassword() and verifyPassword() getOTP() and verifyOTP() which all help with backend work of deleting the account.
- This use case description will likely connect to your “user dashboard” use case which will include “Delete Account”. There is also a likely chance that this method will also connect to an already existing use case of OTP verification. You can simply use the code for OTP verification to do one aspect of this use case.
