**Sprint 1 Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **US ID** | **Title** | **Priority** | **Size (Story Points)** |
| 16 | Access System | 1 | 8 |
| 4 | Create Accounts | 2 | 5 |
| 5 | Edit Accounts | 3 | 1 |
| **Total** | | | 14 |

US 16 - *Access System(login/logout)*

Description: As a Member I want to login/logout so I can access the system.

**Functional/System Requirements**

1. Member needs a username.
2. Member needs to have a password set.
3. \*\*Member needs a PIN for two-factor authentication??
4. System must check if username inputted matches a username in the database.
5. If a username exists, system must check whether supplied password matches password inputted.
6. If login successful, display login successful and available options based on username’s tiered status
7. If login unsuccessful, display login unsuccessful.

**Non-Functional Requirements**

\*\*Mailed PIN number??

**Notes**

If the login is unsuccessful, member may be asked to prove they are not a bot or account might become locked.

US 4 - *Create Accounts*

Description: As a Manager I want to create accounts to allow employees(associate/manager) to login into the system and perform different services.

**Functional/System Requirements**

1. No employee other than a manager can create accounts
2. Manager must provide login information again before creating employee account

**Non-Functional Requirements**

None

**Notes**

Description

US 5 - *Edit Accounts*

Description: As a Manager I want to edit an account to correct/modify the information in a user’s account.

**Functional/System Requirements**

1. Manager’s username flagged for ability to edit.
2. Determination of what fields can be edited and flag what should not be edited (while still making it editable).

**Non-Functional Requirements**

Description

**Notes**

Description

***US-16 Access System (login/logout)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Group | * Draw an initial class diagram * Create a Member class |
| 2 | 1 | Group | * Ask member for username * Write an askUsername method |
| 3 | 1 | Group | * Ask member for password * Write an askPassword method |
| 4 | 2 | Group | * Require PIN for two-factor authentication * Write a generatePin method |
| 5 | 1 | Group | * System check to see if username exists * Write a checkUsername method |
| 6 | 1 | Group | * If username exists, then grab password associated with it * System check to see if password matches * Write a checkPassword method |
| 7 | 2 | Group | * Write a loginSuccessful method   + Grabs username’s associated role * Call other methods associated with role from this method |
| 8 | 1 | Group | * Write a logOut method   + When user types “/lo”, they’re prompted “Are you sure you want to logout?” (Answer Y or N) and then logged out |
| 9 | 1 | Group | * Write a loggedIn method |
| 10 | .5 | Group | * Write a loginUnseccessful method   + Starts log in cycle again   + Only allows two more times * Write a accLock - **(undecided)**   + Locks account after 2 or more tries for a period of time |

***US-4 ID Create Accounts***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Group | * Draw an initial class diagram * Create a Manager class |
| 2 | 1 | Zack, Brandon | * If user that’s logged in is a manager, can call method to create account   + Must provide log in again by calling the askUsername method |
| 3 | 1 | Amanda | * If user is already a member/associate/manager, call the askUsername method again, and go to the checkPassword method   + If successful, information on that user appears   + Write a method changeRole to change role variable |
| 4 | 3.5 | Thara | * If the user is not already a member/associate/manager, form appears to create account   + Name, address, phone number, username from name, password chosen by user |

**US 5 - *Edit Accounts***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Group | * Draw an initial class diagram |
| 2 | 1 | Brandon | * If user that’s logged in is a manager, can call method to edit account   + Must provide log in again by calling the askUsername method |
| 3 | 2 | Zack / Amanda | * Write a method called editAccount   + Can change name, address, phone number, or reset password |