

Phase3 Write-Up

1. The tasks that you would like graded for this phase
 - a. Task 3.1
 - b. Task 3.2
 - c. Task 3.5
 - d. Task 3.6
2. For each task you completed in this phase (Required or Additional), a brief description of the changes you made to the provided code, e.g. the names of any classes or methods that were changed, new methods that were created, etc.
 - a. Task 3.1
 - i. `UIInterface`:
 - `start` method: Now includes an option to register a new organization;
 - `registerOrganization` method: Handles the process of registering a new organization, including input validation and error handling;
 - Minor adjustments: Ensuring that error messages are meaningful and guiding the user back to retry if necessary.
 - ii. `api.js`:
 - Added `/createOrg` endpoint to handle new organization creation.
 - Included validation to check for existing login names and return meaningful error messages.
 - b. Task 3.2
 - i. Made a function `DataManager.makeUpdatePasswordRequest` that takes in a org id and a new password and calls the API to update it
 - ii. Made a file `DataManager_makeUpdatePasswordRequest_Test` with test cases for the aforementioned function
 - iii. Made the `/updatePassword` route in `api.js`
 - iv. Made a function `UIInterface.changePassword` that prompts the user for their old password and new password, calls `DataManager.makeUpdatePasswordRequest` if everything is good, and gives descriptive errors if anything goes wrong
 - v. Modified the `UIInterface.start` function, which prompts the user on what action they want to do, to offer the choice of changing the password
 - c. Task 3.5
 1. `UIInterface.java`
 - **New Methods Added:**
 - `register()`: Handles the registration process for a new contributor, prompting the user to input required details and sending them to the server.
 - `start()`: Modified to include an option for registration and to handle the new login and registration flow.

2. DataManager.java

- **New Method Added:**

- `createContributor(String login, String password, String name, String email, String creditCardNumber, String cvv, String expMonth, String expYear, String zipCode)`: Sends the registration data to the server and handles the response.

3. JUnit Tests

- **New Test Class Added:**

- `DataManager_createContributor_Test`: Contains unit tests to achieve 100% statement coverage for the `createContributor` method in `DataManager`.

4. api.js

- **New Endpoint Added:**

- `app.use('/createContributor', ...)`: Handles the form submission to create a new contributor, validating inputs, and saving the new contributor to the database.

d. Task 3.6

- i. `DataManager.java`

- 1. Added method

- `DataManager.makeContributorPasswordUpdateRequest` which allows user who is logged in to change their password.

- ii. `api.js`

- 1. Added `/updateContributorPassword` endpoint

- 2. Renamed `/updatePassword` endpoint to

- `/updateOrganizationPassword` endpoint for code coherence.

- iii. `DataManager_makeContributorPasswordUpdateRequest_Test.java`

- 1. Added sound test case for `updateContributorPassword` function.

3. N/A

4. N/A

5.

- a. Spencer: All of Task 3.2. For Task 3.1, created the

- `DataManager.createOrganization` as well as corresponding test cases, utilizing defensive programming techniques

- b. Rae: All of Task 3.6. Minor revision for Task 3.2 for readability.

- c. Grace: All of Task 3.5. For Task 3.1, added `/createOrg` endpoint to handle new organization creation in the Server-Side, and Included validation to check for existing login names and return meaningful error messages. Also made changes to `UserInterface` to allow a user to start the app without providing login

credentials initially and give them the option to log in or create a new organization.