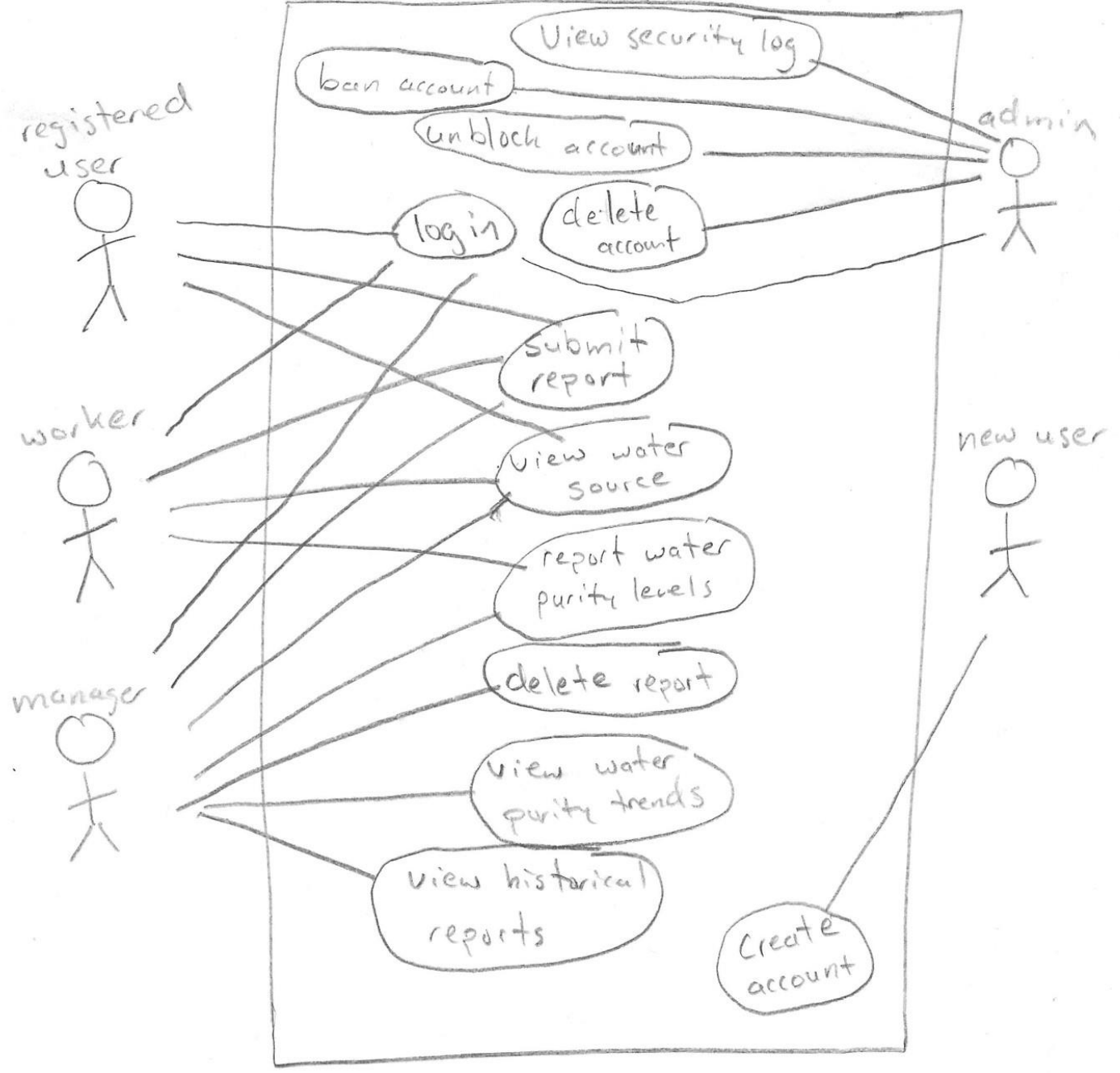


Water Reporter



Noah Rickles

Title: As a registered user, I want to be able to log in with my credentials, so that I can have access to the application.

Tasks:

- create login activity, where registered users enter their usernames and passwords
- create buttons for login, register, help, and cancel actions
- create attributes for length and content of valid usernames and passwords to be entered in the login screen
- implement system to check entered username and password against backend database
- create error messages for invalid usernames and passwords
- allow access into the main application after successful login

Acceptance Scenarios:

- GIVEN a registered user is viewing the login screen, WHEN the registered user presses the back or cancel button, THEN the application returns to the welcome screen.
- GIVEN a registered user is viewing the login screen, WHEN the registered user presses the help button, THEN display the attributes of a valid username and password
- GIVEN a registered user is viewing the login screen, WHEN the registered user enters an invalid username and/or invalid password, THEN display an error explaining what is incorrect and prompt to try again.
- GIVEN a registered user is viewing the login screen, WHEN the registered user enters a valid username and valid password and presses the login button, THEN the username-password pair is checked against the backend database and the user is given or denied access.

Done Done Criteria:

- a registered user can login and access the application if their credentials are valid and matched with the database
- test cases are written based on the acceptance scenarios and run until they have all been successfully passed
- code is reviewed by another member in the group

James Matthew Lord

Title: As a manager, I want to delete a report, so that bad reports are no longer part of the database of water sources.

Tasks:

- Implement a function in the Manager class for deleting posts
- Add logic to confirm deletion so posts are not accidentally removed
- Add button to manager view
- Implement controller/activity changes

Acceptance Scenarios:

- GIVEN a manager is viewing the reports page, WHEN he/she selects a report, THEN the option to delete the report appears.
- GIVEN a manager chooses to delete a report, WHEN he/she clicks the delete button, THEN a confirmation message appears to prevent unexpected removal.
- GIVEN a manager is viewing the confirmation, WHEN he/she clicks the 'confirm deletion' option, THEN the report is removed and the manager is returned to the reports screen.
- GIVEN a manager is viewing the confirmation, WHEN he/she clicks the 'cancel deletion' option, THEN the report is not removed and the manager is returned to the reports screen.

Done Done Criteria:

- Delete method is implemented such that the proper action is taken when the manager selects a post.

Nikhil Ramesh

Title: As an admin, I want to ban or delete an account, so that I can prevent malicious users from submitting reports or using the application whatsoever.

Tasks:

- Add functionality to visit a user's profile as an administrator.
- Implement buttons on user profile to ban/delete the account.
- Create confirmation dialog upon ban/delete button press to ensure that no accidental deletions occur.
- Update database of users/passwords to reflect banning or deletion of the account
- Keep record in database of banned users so they cannot create water reports and deleted users so that they may not recreate their account.

Acceptance Scenarios:

- GIVEN an administrator is viewing a user profile, WHEN they click on the ban button, THEN they are prompted to confirm the ban of the user.
- GIVEN an administrator is viewing the ban confirmation prompt, WHEN they click yes, THEN the user is added to a database for users who cannot create reports for the app.
- GIVEN an administrator is viewing the ban confirmation prompt, WHEN they click no, THEN the administrator is redirected to viewing the user's profile again.
- GIVEN an administrator is viewing a user profile, WHEN they click on the delete button, THEN they are prompted to confirm the delete of the user.
- GIVEN an administrator is viewing the deletion confirmation prompt, WHEN they click yes, THEN the user is added to a database for users who cannot recreate accounts for the app.

- GIVEN an administrator is viewing the deletion confirmation prompt, WHEN they click no, THEN the administrator is redirected to viewing the user's profile again.

Done Done Criteria:

- All specified buttons, activities, pop-ups and prompts have been implemented into the user interface.
- All possible actions have been implemented that are related to the acceptance scenarios.
- Code is tested and reviewed by other members of the group

Renner Brown

Title: As a new user, I want to create an account, so that I become a registered user.

Tasks:

- create register activity, where unregistered users enter their desired usernames and passwords
- create buttons for cancel, create account, and help
- create attributes for length and content of valid usernames and valid passwords to be entered in the register screen
- implement system to check if the entered username is available in the backend database
- create error messages for invalid usernames and passwords
- allow access into the login screen after successful register

Acceptance Scenarios:

- GIVEN an unregistered user is viewing the register screen, WHEN the registered user presses the back or cancel button, THEN the application returns to the login screen.
- GIVEN an unregistered user is viewing the register screen, WHEN the unregistered user presses the help button, THEN display the attributes of a valid username and password
- GIVEN an unregistered user is viewing the register screen, WHEN the unregistered user enters an unavailable username and/or invalid password, THEN display an error explaining what does not meet the username/password criteria and prompt to try again.
- GIVEN an unregistered user is viewing the register screen, WHEN the unregistered user enters a valid username and valid password and presses the register button, THEN the username-password pair is stored in the backend database and the user is redirected to the login screen.

Done Done Criteria:

- an unregistered user can register and add his registration info to the database of user info
- test cases are written on the basis of acceptable passwords and available usernames and tested until they pass
- code is reviewed by another member in the group

Manoj Kandiakounder

Title: As a registered user, I want to be able to view a water source, so that I can check the details of a given water source.

Tasks:

- create screen to view the list of water sources (prioritization can be based on the different attributes of a water source such as proximity, cleanliness, volume, etc. (excluding ID))
- create interface to select a water source from the list of water sources
- create page that loads attributes of the water source based on
- create button to return to the page with the list of water sources

Acceptance Scenarios:

- GIVEN a registered user is on the home screen, WHEN the registered user pressed the tab for the water sources screen, THEN the application returns the water sources screen
- GIVEN a registered user is viewing the water sources screen, WHEN the registered user presses on a UI element containing the name of a water source, THEN the application returns a screen with the data of that water source
- GIVEN a registered user is viewing a specific water source screen, WHEN the registered user presses on the button either labeled “Back” or displayed with a return logo, THEN the applications returns the water sources screen

Done Done Criteria:

- a registered user can view the names of all water sources, and the data of any given water source
- test cases are written based on the acceptance scenarios and run until they have all been successfully passed
- code is reviewed by another member in the group