

Zachary Cheshire User Story

Title:

-As a REGISTERED USER, I want to submit a report on water availability detailing the location, type, and condition of the water so I can contribute to the data in the app.

Tasks:

- Implement a submitReport function
- in submitReport include variables that set the location, type, condition of water
- implement function in controller/activity
- add button to allow user to submit report, which ultimately calls the submitReport function
- add radio buttons to allow user to set location, type, and condition, which will be passed to submitReport function
- Implement back button on the report form to return the user to the previous screen
- Implement a login button and text field for the registered user to enter in their username and password

Acceptance Scenarios:

- Given a registered user is in any activity, WHEN the registered user clicks the add report button, THEN the registered user is shown the form to fill out a report
- Given a registered user is in the report form, WHEN the registered user clicks the submit button, THEN the registered user's form is submitted and added to the reports list
- Given a registered user in the report form, WHEN the registered user clicks the back button, THEN the registered user is returned to the previous activity
- Given a registered user in the login screen, WHEN the registered user fills out the username and password text field and clicks the login button, THEN the registered user is verified and sent to the map view
- Given a registered user in the login screen, WHEN the registered user fails to enter in a valid login, THEN the registered user is prompted with a error message
- Given a registered user is in the report form, WHEN the registered user clicks the submit button and did not fill out all fields, THEN the registered user is prompted with a error message and asked to refill the form

Done Done Criteria:

- Unit test are written or updated to check for allowing a registered user to login, submit a report, backout of a report, and are passed

Anya User Story

Title:

-As a POTENTIAL USER, I want to create a new account so I can begin to use the app.

Tasks:

- implement a button on login screen for potential user to create an account
- take the user to the account creation page
- implement text fields for potential user to enter their username and password
- check their potential username to make sure it's not a duplicate
- new user is added to registered users class
- create segue to home page once new user is created

Acceptance Scenarios:

-GIVEN a potential user is on the home screen, WHEN they choose to create a new account, THEN they are shown the account creation page where they can choose a username and password.

-GIVEN a potential user is on the account creation page, WHEN they press the back button, THEN they are taken back to the log in screen.

Failure Scenarios:

-GIVEN a potential user is on the account creation page, WHEN they try to create an account using a username that already exists, THEN they are asked to choose a different username.

Done Done Criteria

-Unit tests are written to make sure new accounts are created and given the right permissions and no duplicate usernames are allowed

Jerod Ray

Title:

-As a MANAGER, I want to be able to delete user reports when I access the feed so I can keep false data off the app.

Tasks:

- implement deleteMe function to UserReport class
- implement deleteUserReport function to UserReportList class
- add 'Delete' button to Manager's user report view
- implement changes in controller/activity

Acceptance Scenarios:

- GIVEN a manager is viewing a user report, WHEN the professor presses the delete button, THEN the user report will be deleted and removed from the list of user reports
- GIVEN a manager is viewing a user report, WHEN the professor presses the delete button, THEN the manager's view changes to the list of user reports
- GIVEN a manager is viewing a user report, WHEN the professor presses the back button, THEN the user report is not deleted

Done Done Criteria:

- Unit tests are written or updated to include Registered User deletions and are passed

Jack User Story

Title:

- As a REGISTERED USER, I want to log in to an existing account so I can resume activity on an existing account.

Tasks:

- Add current User to model
- Implement login function in model
- Implement a verifyUserExists function to model
- Implement logout function in model
- Add login and logout functionality to controller
- Add login and logout functionality to view

Acceptance Scenarios:

- GIVEN a user is at the title screen, WHEN the user presses the login button, THEN the login form will be displayed to the user
- GIVEN a user is viewing the login form, WHEN the user presses the submit button, THEN the user's view changes to the home screen
- GIVEN a user is viewing the login form, WHEN the user presses the back button, THEN the user's view is returned to the title screen and the information is not stored.

Done Done Criteria:

- Unit tests are written or updated to include User login functionality and are passed
- Unit tests are written or updated to include User logout functionality and are passed

Sammi's User Story

Title:

- -As an ADMINISTRATOR, I want to ban a user from submitting reports so I can keep false reports off the app.

Tasks:

- create a boolean for if a user can post or not
- make changes to Registered User class
- make changes to Model class
- add ban button to administrator's view
- add unban button to administrator's view
- implement changes in controller/activity

Acceptance Scenarios:

- GIVEN an administrator is editing an existing user, WHEN the administrator presses the ban button, THEN the user's posting ability is switched to false, ban button switches to unban, and the changes are reflected in the model.
- GIVEN an administrator is editing an existing user, WHEN the administrator presses the unban, THEN the user's posting ability is switched to true, unban button switches to ban, and the changes are reflected in the model.
- GIVEN an administrator is editing an existing user WHEN the administrator presses the back button, THEN the no changes occur.

Done Done Criteria:

- Unit tests are written or updated to include if users can post or not and are passed