

Testing Strategy Document

ID 1 and ID 2 Updates: UI and End to End Testing

Our approach with UI testing is using a combination of Jest and the React testing library. We will particularly make sure that essential UI components (e.g. titles, descriptions, links, etc.) are in place, mainly using the *expect* API from Jest. On later deliverables where we have finished implementing all the features for our UI we would use snapshots to make sure that the visual aspect of the webpage remains as expected.

Regarding end to end testing we will use the Playwright Library. We will navigate through the pages testing the main function of them, for example being able to drop files, click buttons or links and ensuring that these work as expected. In this sense we will test the Beap Engine API, making sure that the data uploaded is transformed correctly into a .csv file and that the user can use the available ML models to get their predicted files.

Logging

This section briefly describes the motivation for logging in our project, our tools used, and recommends common best practices for developers and quality assurers.

Logging will be used in our project to address two primary concerns which are to (1) understand common (intended and unintended) use case scenarios from real users and (2) to record failures and help identify faults.

We plan to use Rollbar and Crashlytics. Rollbar is a multilevel logging framework that provides a familiar logging API for several languages and technologies including Java Spring, React, and Node.js. A different logging framework will be needed for R scripts.

Crashlytics will be used to record program crashes. This can be deployed on Digital Ocean. Crashlytic reports include logging that shows a user's path and stack trace leading up to the point of the crash.

Logging Severity Levels

Logging messages are expected to be written by both the dev team and the QA team.

Debug - These logging messages should be placed in most meaningful function and function calls. It is recommended to write debug messages with the same frequency and in similar places as assertions. The message types are usually only used in failure analysis.

Info - Record user events and API calls. For instance, “a controller of your authorization API may include an INFO log level with information on which user requested authorization if the authorization was successful or not” (Kuc, 2023).

Warning - This message type needs to be used anytime something *unexpected* occurs even if no functionality is impacted.

Error - This message type needs to be used when there is a failure that impedes one or more functionality but does not cause the program to crash.

Critical - This message type is used when there is a failure that causes the system to crash or prevents the system from running.

What work needs to be done:

Assertions and Debug messages need to be able to be written in only one line of code. This will promote logging and assertion use.

References for logging level section:

Isaiah, A. (2023, November 23). *Log levels explained and how to use them*. Better Stack Community.

<https://betterstack.com/community/guides/logging/log-levels-explained/>

Kuc, R. (2023, November 24). *Logging levels: What they are & how to choose them*.

Sematext. <https://sematext.com/blog/logging-levels/>

Mendes, E., & Petrillo, F. (2021, December 7). *Log severity levels matter: A multivocal mapping*. arXiv.org. <https://arxiv.org/abs/2109.01192>

Usage Example

source <https://docs.rollbar.com/docs/javascript>

```
// Caught errors
try {
  doSomething();
} catch (e) {
  Rollbar.error("Something went wrong", e);
}

// Arbitrary log messages. 'critical' is most severe;
// 'debug' is least.
Rollbar.critical("Connection error from remote Payments
API");
Rollbar.error("Some unexpected condition");
Rollbar.warning("Connection error from Twitter API");
Rollbar.info("User opened the purchase dialog");
Rollbar.debug("Purchase dialog finished rendering");

// Can include custom data with any of the above.
// It will appear as `message.extra.postId` in the
// Occurrences tab
Rollbar.info("Post published", {postId: 123});

// Callback functions
Rollbar.error(e, function(err, data) {
  if (err) {
    console.log("Error while reporting error to Rollbar:
", e);
  } else {
    console.log("Error successfully reported to Rollbar.
UUID:", data.result.uuid);
  }
});
```

Notes on the use of Mockups:

We have created different Mockups for the pages that we are going to design, so we can be sure that their design is correct and we can ensure usability aspects. These Mockups are part of ID2.

Note on testing of R modules

Since one of the possible features asked to be implemented by the stakeholder was the adding support to upload data from Garmin watches, we may need to test the R module that would interact with this data. This would mainly consist of unit tests done manually. However, this feature would only be done if time allows to develop it.

Note on Spike Prototypes

We would invest in spike prototypes whenever we feel unsure of how the different technologies that we are using would interact between each other. Our first spike prototypes consist of a program that uses React as its frontend component and Java as its backend.

For ID2, we created spike prototypes on the use of playwright and rollbar. We integrated these spike prototypes into our main project.

Note on Smoke Test and Our different kind of testings:

Smoke Tests done for ID1 and ID 2 have basically been manual testing in which we have ensured that the pages that we created are rendered and that one can navigate through the different pages of the app. Additionally, for UI test smokes we rely on the automated tests that we have created using playwright and jest.

Regarding test smokes for functional features we will perform manual tests for example to make sure that we can upload files to the page and then download it as a .csv. In this case the first smoke test would be to ensure that the file that we uploaded gets into our database. In the same way we will test that the predicting ML algorithms actually produce different files of data.

As we have mentioned before, one of our main investments in testability is using a combination of manual and automated testing, which is currently done on the frontend(the main part where we have done changes). We will perform the same thing once we refactor the backend of our program with the new modules that we create. However, we have also invested in making our pipeline safe. Automated testing is part of the CI/CD pipeline because for every time we commit or create a

pull request, all our unit and end to end tests will run. Finally, we have also invested in containerization, since we have moved our complete system to Docker.

ID3 Updates Notes:

Notes on Mocking

Based on our system, we could take advantage of mocking different scenarios. For example, we could mock calls to the database and the R repository of our system. Since calling the database takes a lot of time and we did not write the code that processes the uploaded file into a .csv file, we could mock these components and test that an already predetermined component (which already exists in the database) is correctly transformed into its csv representation and it is shown correctly to the user to be downloaded. In the same way we could mock the call to our R repository by already having the expected file that would be generated by this call.

It is worth mentioning that we are also mocking data and the logging function when we perform our unit tests for our hooks.

Smoke Test Plan for ID 4

For this deliverable our smoke test consisted on running the application, log into it, and make sure that you could navigate through the different pages.

For the next deliverable the smoke test that we will perform will consist of running the application, log into the page, uploading one file and making sure that it is in the database. Depending on development team advancement we may test the functionality of the predicting file page as well.

ID 4 Update Notes:

Testing plan for the Backend

As discussed with Dr. Osgood, we have done some testing on some relevant sections of the backend. For this deliverable we have tested the following files: Access Group Service.java , IncorrectLoginService.java, LoginService.java, MyUserDetails.java, and UserDetails.java. We have performed unit tests on these files.

We created automated unit test cases for all the methods in these files. Specifically we took advantage of mocking to fake the return calls of some functions that were indirectly called within the primarily tested functions. In the next ID we might perform integration testing with some of the backend files.

For ID 5, we will invest in research tools to perform stress and load testing into our system. We will use these tools to generate some test cases and check the performance of our project.

Additionally, we will use the built in profiler of IntelliJ to go through the code of our backend.

Notes on Code Coverage

Two different code coverage reports are generated for both frontend and backend. The frontend makes use of the jest testing library to generate a code coverage report for all overall files, and current files changed. The backend on the other hand makes use of junit and JaCoCo to generate code coverage reports for all overall files and current files changed.

The reports are automatically generated on all open pull requests where both frontend and backend coverage reports are commented. These reports are only generated once all test cases pass in the pipeline. Code coverage reports can also be manually run in the frontend by running **yarn test --coverage --watchAll**. While the backend coverage report can be run using **mvn clean test**.

Our code coverage goal is to reach a minimum of 70% overall coverage for frontend and backend code. This 70% is a metric recommended from CMPT 470 lectures

Note on test Hooks

We have a test hook that we are using in all our tests which allows us to first log in (make the user authenticated) so that the pages load correctly. For our backend code testing we created a Java class called MockFactory which contains several test hooks used to create Mock objects and Fake data. On all of the Jest Unit tests, we used a test hook called renderWithProvider to facilitate the use of Rollbar in our files. Additionally we had one member of the QA to create a playwright test that involves uploading a file and checking that it exists in the database.

Logs of tests:

Front end:

PASS src/components/LoadingSpinner/LoadingSpinner.test.tsx

LoadingSpinner

✓ T4.8 renders the HashLoader with correct props when loading is true (32 ms)

✓ T4.9 does not render the HashLoader when loading is false (19 ms)

PASS src/shared/hooks/useGetPredictedDataList.test.ts

useGetPredictedDataList

✓ T4.25 should get predicted files successfully (40 ms)

✓ T4.26 should handle getPredictedFiles when it errors (8 ms)

PASS src/shared/hooks/useDownload.test.ts

useDownload

✓ T4.3 should download a file from the database (69 ms)

✓ T4.4 should handle download when it errors (12 ms)

PASS src/App.test.tsx

✓ TID 1.4. renders all links (80 ms)

PASS src/pages/LogoutPage/Logout.test.tsx

Logout

✓ T.14 should call handleLogout on render (26 ms)

✓ T4.15 should navigate to home page after logout (71 ms)

✓ T4.16 should display loading state (21 ms)

✓ T4.17 should display error state (4 ms)

PASS src/pages/LoginPage/LoginPage.test.tsx

LoginPage component

PASS src/pages/LoginPage/LoginPage.test.tsx

LoginPage component

✓ T4.11 renders login form with initial state (119 ms)

✓ T4.12 allows user to type in username and password fields (42 ms)

✓ T4.13 test that button submits form with username and password on click (38 ms)

✓ T4.13 test that the rotator buttons change the text on the screen (43 ms)

PASS src/pages/SignUpPage/SignUpPage.test.tsx

SignUpPage component

✓ T4.18 renders sign up form with initial state (127 ms)

✓ T4.19 allows user to type in first name, last name, username and password fields (65 ms)

✓ T4.20 test that button submits form with first name, last name, username and password on click (63 ms)

✓ T4.21 test that form is not submitted if privacy policy is not agreed to (48 ms)

✓ T4.22 test that form info is not submitted if passwords don't match (48 ms)

✓ T4.23 test that form does not work if an element is missing (51 ms)

✓ T4.24 test that the rotator buttons change the text on the screen (47 ms)

PASS src/pages/LogoutPage/Logout.test.tsx

Logout

✓ T.14 should call handleLogout on render (24 ms)

✓ T4.15 should navigate to home page after logout (66 ms)

✓ T4.16 should display loading state (22 ms)

✓ T4.17 should display error state (5 ms)

PASS src/App.test.tsx

✓ TID 1.4. renders all links (57 ms)

PASS src/pages/FileUploadPage/FileUploadPage.test.tsx

✓ TID 1.1. Renders FileUploadPage components (84 ms)

PASS src/shared/hooks/useDownload.test.ts

useDownload

✓ T4.3 should download a file from the database (92 ms)

✓ T4.4 should handle download when it errors (10 ms)

PASS src/shared/hooks/useGetPredictedDataList.test.ts

useGetPredictedDataList

✓ T4.25 should get predicted files successfully (34 ms)

✓ T4.26 should handle getPredictedFiles when it errors (5 ms)

PASS src/pages/HomePage/HomePage.test.tsx (5.058 s)

✓ TID 1.2. Render Home Page components (174 ms)

PASS src/pages/PredictedDataPage/PredictedDataPage.test.tsx (5.06 s)

✓ Renders PredictedDataPage components (205 ms)

PASS src/pages/FileUploadPage/components/FileDropzoneControls.test.tsx (5.152 s)

✓ Renders FileDropzoneControls components (160 ms)

✓ TID 1.7. Should be able to click Fitbit or Apple Watch radio (276 ms)

PASS src/pages/FileUploadPage/components/FileDropzone.test.tsx (5.57 s)

✓ TID 1.6. Renders FileDropzone components (77 ms)

✓ invoke onDragEnter when dragenter event occurs (356 ms)

PASS src/components/LoadingSpinner/LoadingSpinner.test.tsx

LoadingSpinner

✓ T4.8 renders the HashLoader with correct props when loading is true (30 ms)

✓ T4.9 does not render the HashLoader when loading is false (5 ms)

PASS src/shared/hooks/useGetProcessedDataList.test.ts

useListGetProcessedFiles

✓ should get processed files successfully (39 ms)

✓ should handle getProcessedFiles when it errors (6 ms)

PASS src/pages/ProcessDataPage/ProcessedDataPage.test.tsx (5.867 s)

Processed Data Page

- ✓ T3.16 Should properly display the processed data page (854 ms)

PASS src/shared/hooks/useIsUserLoggedIn.test.ts

useIsUserLoggedIn

- ✓ T3.8 returns false and clears storage and cookie when expiresAt is in the past (53 ms)
- ✓ T3.9 returns true when user is logged in and expiresAt is in the future (3 ms)

PASS src/components/HelpPopup/HelpPopup.test.tsx

- ✓ TID 3.1. Renders HelpPopup component (301 ms)

PASS src/shared/hooks/usePredictFile.test.ts

usePredictFile

- ✓ should handle predict successfully (77 ms)
- ✓ should handle predict when it errors (9 ms)

PASS src/shared/hooks/useGetUploadedFiles.test.ts

useGetUploadedFiles

- ✓ T3.19 should get uploaded files successfully (34 ms)
- ✓ T3.20 should handle getUploadedFiles when it errors (66 ms)

PASS src/shared/hooks/useLogin.test.ts

useLogin

- ✓ T3.10 should handle login successfully (67 ms)
- ✓ T3.10 should handle login when it errors (5 ms)

PASS src/components/Navbar/Navbar.test.tsx

- ✓ renders all links when user is authenticated (44 ms)
- ✓ does not render when user is not authenticated (4 ms)

PASS src/shared/hooks/useDeleteFile.test.ts

useDeleteFile

- ✓ T3.17 should handle delete successfully (52 ms)
- ✓ T3.18 should handle delete when it errors (6 ms)

PASS src/shared/hooks/useUpload.test.ts

useUpload

- ✓ should handle uploads successfully (53 ms)
- ✓ should handle errors (5 ms)

PASS src/shared/api/Api.test.tsx

API Tests

- ✓ T3.2 Login: should return mapped data upon successful login (7 ms)

- ✓ T3.3 Login: should throw an error upon failed login (12 ms)
- ✓ T3.4 Logout: should logout successfully (1 ms)
- ✓ T3.5 Logout: should throw an error if fails (1 ms)
- ✓ T3.6 should sign up successfully (1 ms)
- ✓ T3.7 should throw an error if sign up fails (1 ms)

PASS src/shared/hooks/useLogout.test.ts

useLogout

- ✓ T3.12 should handle logout successfully (30 ms)
- ✓ T3.13 should handle logout when it errors (4 ms)

PASS src/shared/hooks/useSignup.test.ts

useSignup

- ✓ T3.14 should handle signup successfully (42 ms)
- ✓ T3.15 should handle signup when it errors (4 ms)

Test Suites: 25 passed, 25 total

Tests: 57 passed, 57 total

Snapshots: 0 total

Time: 8.682 s

Backend:

Console output:

```
2024-03-16 15:58:41 INFO AccessGroupService:90 - in AccessGroupService: get
2024-03-16 15:58:41 INFO AccessGroupService:50 - in AccessGroupService: list
2024-03-16 15:58:41 INFO AccessGroupService:63 - in AccessGroupService: save
2024-03-16 15:58:41 INFO AccessGroupService:50 - in AccessGroupService: list
2024-03-16 15:58:41 INFO AccessGroupService:90 - in AccessGroupService: get
2024-03-16 15:58:41 INFO AccessGroupService:106 - in AccessGroupService:
delete
2024-03-16 15:58:41 INFO AccessGroupService:76 - in AccessGroupService:
Update
2024-03-16 15:58:42 INFO LoginUserService:81 - in IncorrectLoginsService: get
2024-03-16 15:58:42 INFO LoginUserService:110 - in IncorrectLoginsService:
getByUserId
2024-03-16 15:58:42 INFO LoginUserService:42 - in IncorrectLoginsService: list
2024-03-16 15:58:42 INFO LoginUserService:54 - in IncorrectLoginsService: save
2024-03-16 15:58:42 INFO LoginUserService:42 - in IncorrectLoginsService: list
2024-03-16 15:58:42 INFO LoginUserService:81 - in IncorrectLoginsService: get
2024-03-16 15:58:42 INFO LoginUserService:97 - in IncorrectLoginsService: delete
2024-03-16 15:58:42 INFO LoginUserService:67 - in IncorrectLoginsService:
Update
```

2024-03-16 15:58:42 INFO LoginUserService:110 - in IncorrectLoginsService:
getByUserId
2024-03-16 15:58:42 INFO LoginUserService:92 - in LoginUserService: get
2024-03-16 15:58:42 INFO LoginUserService:52 - in LoginUserService: list
2024-03-16 15:58:42 INFO LoginUserService:65 - in LoginUserService: save
2024-03-16 15:58:42 INFO LoginUserService:52 - in LoginUserService: list
2024-03-16 15:58:42 INFO LoginUserService:92 - in LoginUserService: get
2024-03-16 15:58:42 INFO LoginUserService:108 - in LoginUserService: delete
2024-03-16 15:58:42 INFO LoginUserService:78 - in LoginUserService: Update
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:289 - in UserService:
loadUserByUsername
2024-03-16 15:58:42 INFO UserService:117 - in UserService: get
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:269 - in UserService: getUserByUsername
2024-03-16 15:58:42 INFO UserService:63 - in UserService: save
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:269 - in UserService: getUserByUsername
2024-03-16 15:58:42 INFO UserService:315 - in UserService: loadUserDetails
2024-03-16 15:58:42 INFO UserService:52 - in UserService: list
2024-03-16 15:58:42 INFO UserService:63 - in UserService: save
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:269 - in UserService: getUserByUsername
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:315 - in UserService: loadUserDetails
2024-03-16 15:58:42 INFO UserService:315 - in UserService: loadUserDetails
2024-03-16 15:58:42 INFO UserService:52 - in UserService: list
2024-03-16 15:58:42 INFO UserService:117 - in UserService: get
2024-03-16 15:58:42 INFO UserService:289 - in UserService:
loadUserByUsername
2024-03-16 15:58:42 INFO UserService:269 - in UserService: getUserByUsername
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:269 - in UserService: getUserByUsername
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:315 - in UserService: loadUserDetails
2024-03-16 15:58:42 INFO UserService:132 - in UserService: delete
2024-03-16 15:58:42 INFO UserService:148 - in UserService: login
2024-03-16 15:58:42 INFO UserService:269 - in UserService: getUserByUsername
2024-03-16 15:58:42 INFO UserService:104 - in UserService: Update
2024-03-16 15:58:42 INFO UserService:269 - in UserService: getUserByUsername

ScreenShots with tests:

✓ AccessGroupServiceTest	40 ms
✓ testGet	25 ms
✓ testList	2 ms
✓ testSave	5 ms
✓ testListEmpty	0 ms
✓ testGetNull	0 ms
✓ testDelete	6 ms
✓ testUpdate	2 ms
✓ IncorrectLoginsServiceTest	62 ms
✓ testGet	56 ms
✓ testGetByUserId	1 ms
✓ testList	1 ms
✓ testSave	1 ms
✓ testListEmpty	0 ms
✓ testGetNull	0 ms
✓ testDelete	1 ms
✓ testUpdate	1 ms
✓ testGetByUserIdNull	1 ms
✓ LoginUserTest	8 ms
✓ testGet	2 ms
✓ testList	1 ms
✓ testSave	1 ms
✓ testListEmpty	2 ms
✓ testGetNull	1 ms
✓ testDelete	0 ms
✓ testUpdate	1 ms
✓ LoginUserTest	8 ms
✓ testGet	2 ms
✓ testList	1 ms
✓ testSave	1 ms
✓ testListEmpty	2 ms
✓ testGetNull	1 ms
✓ testDelete	0 ms
✓ testUpdate	1 ms
✓ UserServiceTest	39 ms
✓ testValidLoginLocked	6 ms
✓ testLoadUserByUsername	6 ms
✓ testGet	1 ms
✓ testInvalidLoginLocked	1 ms
✓ testSaveUserAlreadyExists	1 ms
✓ testInvalidLoginUnlock	4 ms
✓ testList	2 ms
✓ testSave	1 ms
✓ testInvalidLoginFirstTime	2 ms
✓ testValidLogin	2 ms
✓ testLoadUserDetails	1 ms
✓ testListEmpty	0 ms
✓ testGetNull	0 ms
✓ testLoadUserByUsernameNull	1 ms
✓ testGetUserByUsername	1 ms
✓ testInvalidUsernameAndPasswordLogin	2 ms
✓ testValidLoginWithUnlock	2 ms
✓ testDelete	2 ms
✓ testInvalidLogin	2 ms
✓ testUpdate	1 ms
✓ testGetUserByUsernameNull	1 ms

Front end Code Coverage:



github-actions (bot) commented 2 days ago • edited

...

Frontend Code Coverage Report

St. ?	Category	Percentage	Covered / Total
●	Statements	76.8% (-23.2% ▼)	437/569
●	Branches	57.72% (-42.28% ▼)	86/149
●	Functions	78.1% (-21.9% ▼)	107/137
●	Lines	77.44% (-22.56% ▼)	436/563

▼ Show new covered files 🌟

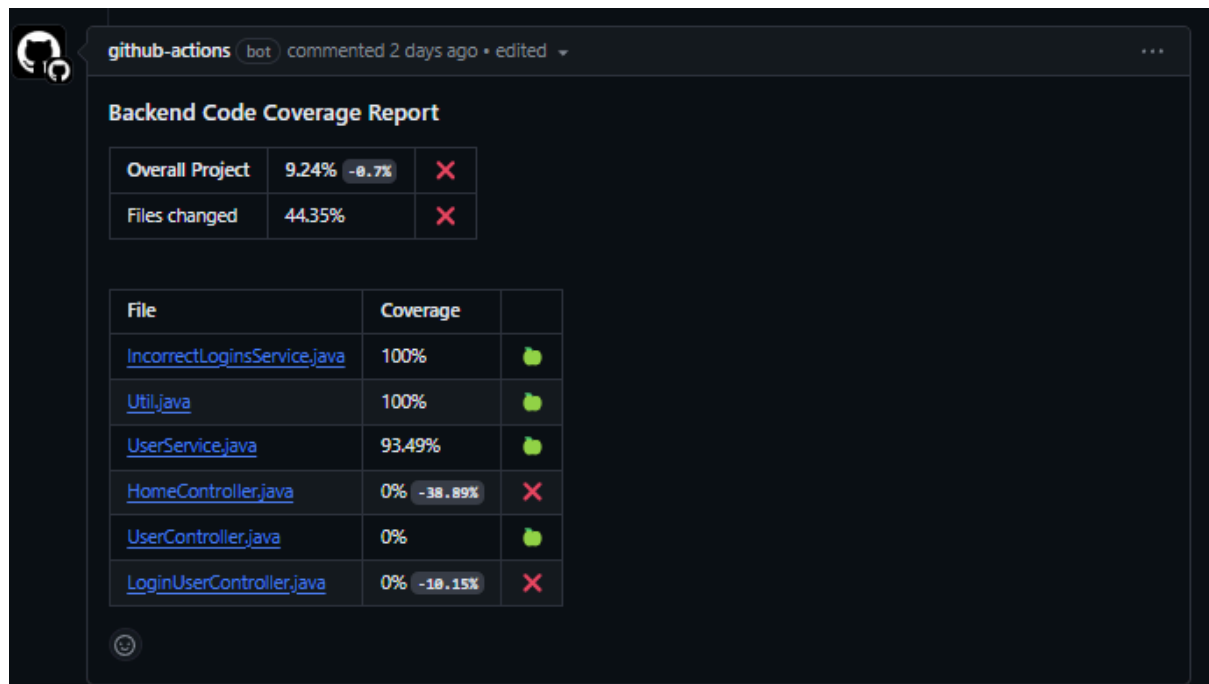
St. ?	File	Statements	Branches	Functions	Lines
●	... / baseapi.ts	100%	75%	100%	100%
●	shared/api/Api.ts	100%	50%	100%	100%
●	... / index.ts	100%	100%	100%	100%
●	... / useLogin.ts	100%	100%	100%	100%
●	... / LoginPage.tsx	74.42%	70%	58.33%	74.42%
●	... / useSignup.ts	100%	100%	100%	100%
●	... / SignUpPage.tsx	80.3%	69.23%	68.75%	80.3%
●	... / useUserLoggedIn.ts	100%	100%	100%	100%
●	... / useAuth.tsx	90%	50%	100%	88.89%
●	... / HelpPopup.tsx	85.71%	56.25%	100%	85.71%
●	... / useDownload.ts	45.16%	100%	60%	48.28%
●	... / ProtectedRoute.tsx	100%	100%	100%	100%
●	... / useGetProcessedDataList.ts	100%	100%	100%	100%
●	... / usePredictFile.ts	100%	100%	100%	100%
●	... / ProcessedDataPage.tsx	64.71%	36.84%	75%	64.71%
●	... / useUpload.ts	100%	100%	100%	100%
●	... / FileDropzone.tsx	56.63%	33.33%	66.67%	58.02%
●	... / useGetPredictedDataList.ts	100%	100%	100%	100%
●	... / useLogout.ts	100%	100%	100%	100%
●	... / LoadingSpinner.css	100%	100%	100%	100%
●	... / LoadingSpinner.tsx	100%	100%	100%	100%
●	... / Logout.tsx	100%	100%	100%	100%
●	... / useGetUploadedFiles.ts	90.91%	100%	83.33%	90.91%
●	styles/navbar.css	100%	100%	100%	100%
●	... / Navbar.tsx	100%	100%	100%	100%
●	... / useDeleteFile.ts	100%	100%	100%	100%
●	... / FileDropzoneControls.tsx	100%	100%	100%	100%
●	... / PredictedDataPage.tsx	50%	35.71%	16.67%	51.72%
●	... / FileUploadPage.tsx	100%	100%	100%	100%
●	... / HomePage.tsx	100%	100%	100%	100%
●	... / GoogleLogin.tsx	0%	0%	0%	0%

Test suite run success

73 tests passing in 28 suites.

Report generated by [jest coverage report action](#) from [af46847](#)

Backend Code coverage:



ID 5 Update Notes:

Testing plan for the Backend

One more time, as discussed with Dr. Osgood, we have done testing on relevant sections of the backend. For this deliverable we focused on files from the Controller layer as well as additional files from the Service layer. The tested files were the following: `AccessGroupController.java`, `UserController.java`, `WatchController.java`, and `RawDataServiceTest.java`. We created automated unit test cases for all the methods in these files. Specifically, we took advantage of mocking to fake the return calls of some functions that were indirectly called within the primarily tested functions.

It is worth noting that we combined the functionality of `AppleWatchController` and `FitbitController` into a single module called `WatchController`, in order to reduce the amount of code that manages this functionality and avoid code duplication. Additionally, for the testing of this file, we created an `HttpSession` object to mock an authenticated session (with a valid user ID) and cover all the possible scenarios that the code could encounter.

Notes on the use of Profiler for the Backend

We attempted to use a profiler on the backend of the project but we ran into some blockers. We looked into two different profilers that work with a Java backend which were the built-in profiler in IntelliJ and VisualVM. The problem was that the backend was running in a docker container and the only way for the previously mentioned profilers to track the performance of the backend is to run the backend directly from IntelliJ itself. We were not able to figure out a

way to run the backend on IntelliJ and were not able to record the statistics from the profilers. Other performance tests were done through the form of load/stress testing.

Notes on Code Coverage

We are using the same strategy from the previous deliverable: we generate two different code coverage reports for both frontend and backend. We are keeping up with our standard of 70% of code coverage for the code that we created on the frontend, and we continued increasing the code coverage for the backend code. With the new tests that we created we have reached approximately + 25% of coverage.

Notes on Updated Smoke Test

We are running a playwright test for our smoke test. This test is labeled with a tag of @smoke and we have added another workflow specifically to run it. The workflow is run on all pull requests and any commits to master. To increase the speed of the smoke test we have implemented a default account into our database that has 2 files in each part of the system so that our smoke does not have to wait for processing times while still checking if the backend is hooked up to the front end. By both performing an actual login and checking that all the files are shown we should be able to see any large problems. Finally, the smoke test also checks that some basic elements are shown.

The code for this test is located inside frontend/tests_end2end/SmokeTest.spec.ts

Notes on Stress and Load Testing

*Note: There is a separate document for the detailed overview of the Stress and Load Testing of the backend.

<https://docs.google.com/document/d/16cCqoXyRwKVEK9FgEpy4cGCTmbSgO3bU0IRzPVhz328/edit?usp=sharing>

We ran a load test that mocked 50 users that went through the main functionality of the Beap Engine twice through for a total of 100 samples. The main functionality includes the Login, File Upload, Processing, Prediction, and Download. This exceeds the expected users of around 5 users at a time, but the system still managed to handle all 50 users without much hindrance to speed or success. All tests ran with 0% errors and server response speeds within our expected ranges.

We did our best to test the APIs with the appropriate file sizes, but found it hard to test the processing, predicting and downloading because our example files were small in size. It is not possible to test the API endpoints with random files with the appropriate sizes since the APIs expect a very specific file and any file that does not fit the requirements causes error responses.

Notes on Acceptance Test Cases

Based on the updated software requirements of the system and our Use cases we have created the following acceptance test cases:

1. ID5.88 Test: User can log in

User Story: As a researcher I want to be able to log into the page

Preconditions: None

Acceptance Criteria:

- The login button is available
- The user can input their username and password
- The user can enter into the page.

Steps:

- Navigate to beapengine.com
- Click on login
- Enter username and password
- Click login

2. ID5. 89 Test: User can upload data, process it and download the process data

User story: As a researcher I want to be able upload data, process the data and download the processed file

Preconditions: The user is logged in

Acceptance Criteria:

- The zone to drop the file is visible
- The system provides feedback as it uploads/process the file
- The button to process a file is visible
- The button to download a file is available
- The file is saved into the user's computer

Steps:

- Navigate to the file upload page
- Select the type of file that you want to upload (Fitbit or AppleWatch)
- Click on the file dropzone to select the file / Drop the file in the dropzone
- Optionally: Click on the "Add file" button to upload more files
- Click on the "Upload" button
- Click on "Minimize" button of the pop up progress bar
- Check the box for the file that was just uploaded and click on the "Process" button
- Minimize the popup progress bar
- Navigate to the Processed Files page
- Select the file that you want to download
- Click on the "Download File" button
- Save the file in your computer

3. ID5. 90 Test: User can use the machine learning algorithms on the uploaded files

User story: As a researcher I want to use the machine learning algorithms on my uploaded files to predict

Preconditions: The user is logged in and a file was uploaded and processed correctly

Acceptance Criteria:

- The button to Predict a file is available
- Different predict models can be selected and they all work
- Predicted Files can be downloaded

Steps:

- Navigate to the Processed Files Page
- Select the Machine Learning Algorithm to be used
- Select the file to be predicted
- Click on the "Predict File" button
- Minimize the pop up progress up bar

- Navigate to the Predicted Files page
- Select the predicted File
- Click on the "Download File" button
- Save the file to your computer

4. ID5. 91 Test: User can sign up into the page

User story: As a new user I want to be able to sign up and use the page

Preconditions: None

Acceptance Criteria:

- The button to sign up is visible
- A new account is created
- The User can log into with the new created account

Steps:

- Navigate to beapengine.com
- Click on the "Sign up" Button
- Fill the forms with your information
- Check that the box to agree to "the terms and privacy policy"
- Click on the "Sign up" Button

6. ID5. 92 Test: User can delete their data and their account

User story: As a participant I want to be able to delete my data and my account

Preconditions: The user has an account, has uploaded data and is logged in the page

Acceptance Criteria:

- The Data is deleted
- The account is deleted

Steps:

- Click on the upper right corner, on the user icon
- Click on "My account"
- Click on "Delete My Data"
- Click "Confirm"
- Click "Delete My Account"
- Click "Confirm"

7. ID5. 93 Test: User can manage the information of their account

User story: As a participant I want to be able to change the password of my account

Preconditions: The user has an account and is logged in the page

Acceptance Criteria:

- The button to change the password is visible
- The password of the user is changed
- The user can log in with their new password

Steps:

- Click on the upper right corner, on the user icon
- Click on "My account"
- Click on "Update my Password"
- Fill the forms with the new password
- Click on the "Logout" button
- Sign in with the new password

8. ID5. 94 Test: User can get help on how to use the system

User story: As a new user I want to get instructions on how to use the system

Preconditions: The user has an account and is logged in the page

Acceptance Criteria:

- The "Help" button is visible
- The "Help" button displays the steps to use the system

Steps:

- Click on the upper right corner, on the "Help" button
- Click on the arrows to expand the text of each section

Logs of tests:

Front end(Ran on 2024/04/03):

PASS src/pages/LoginPage/LoginPage.test.tsx (10.304 s)

LoginPage component

- ✓ T4.11 renders login form with initial state (185 ms)
- ✓ T4.12 allows user to type in username and password fields (68 ms)
- ✓ T4.13 test that button submits form with username and password on click (48 ms)
- ✓ T4.13 test that the rotator buttons change the text on the screen (59 ms)

PASS src/components/ProgressBar/ProgressBar.test.tsx (10.39 s)

- ✓ TID 5.01. Renders ProgressBar component (445 ms)
- ✓ TID 5.02. Does not render ProgressBar component with isVisible=false (18 ms)
- ✓ TID 5.03. Does not render ProgressBar component with percentage < 0 (9 ms)
- ✓ TID 5.04. Does not render ProgressBar component with percentage > 100 (9 ms)
- ✓ TID 5.05. Renders ProgressBar component (108 ms)

PASS src/pages/FileUploadPage/components/UploadedFiles.test.tsx (10.522 s)

Uploaded Files Page

- ✓ T5.10 Should properly render uploaded files with fitbit and apple data with -1 processedId (337 ms)
- ✓ T5.11 On process click process all checked files and call onProgress fn (362 ms)

PASS src/components/ConfirmModal/ConfirmModal.test.tsx (10.792 s)

Uploaded Files Page

- ✓ T5.95 should render confirm modal (244 ms)
- ✓ T5.96 should render null when confirm modal is not visible (11 ms)
- ✓ T5.97 should call onClose when cancel button clicked (105 ms)
- ✓ T5.98 should call onConfirm when confirm button clicked (49 ms)
- ✓ T5.99 should disable the confirm and cancel button when isLoading is true (23 ms)

PASS src/pages/ProcessDataPage/ProcessedDataPage.test.tsx (11.033 s)

Processed Data Page

- ✓ T3.16 Should properly display the processed data page (1138 ms)

PASS src/pages/FileUploadPage/components/FileDropzoneControls.test.tsx (11.09 s)

- ✓ Renders FileDropzoneControls components (273 ms)
- ✓ TID 1.7. Should be able to click Fitbit or Apple Watch radio (466 ms)

PASS src/pages/PredictedDataPage/PredictedDataPage.test.tsx (11.667 s)

Predicted Data Page

✓ T5.117 Should properly display the predicted data page and download file (888 ms)

PASS src/pages/ProfilePage/ProfilePage.test.tsx (12.148 s)

Profile Page

- ✓ T5.100 should render profile page (220 ms)
- ✓ T5.101 should render loading spinner when user details are loading (33 ms)
- ✓ T5.102 should delete data when delete data is clicked and confirmed (347 ms)
- ✓ T5.103 should delete data when delete account is clicked and confirmed (125 ms)
- ✓ T5.104 should change password successfully and show confirm password error state when necessary (420 ms)

PASS src/components/Navbar/Navbar.test.tsx

Navbar

- ✓ renders all links when user is authenticated (147 ms)
- ✓ does not render when user is not authenticated (5 ms)
- ✓ should show modal while logout is loading (124 ms)

PASS src/pages/HomePage/HomePage.test.tsx

- ✓ TID 1.2. Render Home Page components (182 ms)
- ✓ TID 1.2.2. Extraction protocol links are correct (42 ms)
- ✓ TID 1.2.3. Footer links are correct (33 ms)
- ✓ TID 1.2.4. BEAP Lab link is correct (35 ms)
- ✓ TID 1.2.5. Images are rendered correctly (42 ms)

PASS src/App.test.tsx

- ✓ TID 1.4. renders all links (137 ms)

PASS src/pages/FileUploadPage/FileUploadPage.test.tsx

- ✓ TID 1.1. Renders FileUploadPage components (107 ms)

PASS src/components/ErrorSnackBar/ErrorSnackBar.test.tsx

Error SnackBar

- ✓ T5.06 Should return null if error is null (42 ms)
- ✓ T5.07 Should render snackbar if error is defined (135 ms)

PASS src/components/Navbar/components/ProfileMenu.test.tsx

Profile Menu

- ✓ T5.112 should render profile while not clicked (133 ms)
- ✓ T5.113 should render profile and dropdown while clicked (238 ms)
- ✓ T5.114 should call logout and my account nav links when clicked (160 ms)

PASS src/components/LoadingModal/LoadingModal.test.tsx

Loading Modal

- ✓ T5.110 should render loading modal (175 ms)
- ✓ T5.111 should render null when loading modal is not visible (10 ms)

PASS src/pages/FileUploadPage/components/FileDropzone.test.tsx

Uploaded Files Page

- ✓ T5.12 Should drop fitbit files as initial default (520 ms)
- ✓ T5.13 Should drop only xml files as for apple watch (181 ms)
- ✓ T5.14 Should not be able to drop json when apple watch is selected (47 ms)
- ✓ T5.15 Should not be able to drop xml when fitbit is selected (47 ms)

PASS src/pages/AboutUs/AboutUs.test.tsx

AboutUs component

Logo

- ✓ renders the BEAP logo (338 ms)

Title

- ✓ renders the main title (86 ms)

Sections

- ✓ renders the Our Mission section (99 ms)
- ✓ renders the Contributors section (92 ms)

Contributors

- ✓ renders the contributors (84 ms)

PASS src/components/PrivacyPolicy/PrivacyPolicy.test.tsx

PrivacyPolicy component

Logo

- ✓ renders the BEAP logo (111 ms)

Titles

- ✓ renders the main title (36 ms)
- ✓ renders the subtitle (26 ms)

Researcher Info

- ✓ renders the researcher info (19 ms)

Section Titles

- ✓ renders title of Introduction (45 ms)
- ✓ renders title of Purpose of Study (32 ms)
- ✓ renders questions or concerns title (24 ms)
- ✓ renders title of Privacy and Confidentiality (33 ms)
- ✓ renders title of Anonymity (22 ms)
- ✓ renders title of Storage of Data (25 ms)

Section Contents

- ✓ renders the content of the Introduction section (28 ms)
- ✓ renders the content of the Purpose of Study section (25 ms)
- ✓ renders the content of the Questions or Concerns section (28 ms)
- ✓ renders the content of the Privacy and Confidentiality section (23 ms)
- ✓ renders the content of the Anonymity section (19 ms)
- ✓ renders the content of the Storage of Data section (24 ms)

PASS src/pages/LogoutPage/Logout.test.tsx

Logout

- ✓ T.14 should call handleLogout on render (41 ms)
- ✓ T4.15 should navigate to home page after logout (74 ms)
- ✓ T4.16 should display loading state (36 ms)
- ✓ T4.17 should display error state (11 ms)

PASS src/components/HelpPopup/HelpPopup.test.tsx
✓ TID 3.1. Renders HelpPopup component (642 ms)

PASS src/shared/hooks/useUpload.test.ts
useUpload
✓ should handle uploads successfully (107 ms)
✓ should handle errors (14 ms)

PASS src/shared/hooks/useGetUploadedFiles.test.ts
useGetUploadedFiles
✓ T3.19 should get uploaded files successfully (54 ms)
✓ T3.20 should handle getUploadedFiles when it errors (65 ms)

PASS src/shared/hooks/useProcessFile.test.ts
useProcessFile
✓ T5.08 should handle process successfully (103 ms)
✓ T5.09 should handle process when it errors (12 ms)

PASS src/shared/hooks/useGetPredictedDataList.test.ts
useGetPredictedDataList
✓ T4.25 should get predicted files successfully (50 ms)
✓ T4.26 should handle getPredictedFiles when it errors (9 ms)

PASS src/shared/hooks/useGetProcessedDataList.test.ts
useListGetProcessedFiles
✓ should get processed files successfully (50 ms)
✓ should handle getProcessedFiles when it errors (8 ms)

PASS src/shared/hooks/useIsUserLoggedIn.test.ts
useIsUserLoggedIn
✓ T3.8 returns false and clears storage and cookie when expiresAt is in the past (65 ms)
✓ T3.9 returns true when user is logged in and expiresAt is in the future (7 ms)

PASS src/shared/hooks/useDownload.test.ts
useDownload
✓ T4.3 should download a file from the database (106 ms)
✓ T4.4 should handle download when it errors (16 ms)

PASS src/shared/hooks/useGetUser.test.ts
useGetUser
✓ T5.105 should get users successfully (55 ms)
✓ T5.106 should handle get user when it errors (63 ms)

PASS src/components/Icons/index.test.tsx
Icons
✓ T5.109 should render file icon (43 ms)

PASS src/components/Authentication/ProtectedRoute.test.tsx

ProtectedRoute Component

- ✓ T4.34: renders children when user is authenticated (63 ms)
- ✓ T4.35: redirects to Landing Page when user is not authenticated (18 ms)

PASS src/pages/SignUpPage/SignUpPage.test.tsx

SignUpPage component

- ✓ T4.18 renders sign up form with initial state (155 ms)
- ✓ T4.19 allows user to type in first name, last name, username and password fields (83 ms)
- ✓ T4.20 test that button submits form with first name, last name, username and password on click (84 ms)
- ✓ T4.21 test that form is not submitted if privacy policy is not agreed to (62 ms)
- ✓ T4.22 test that form info is not submitted if passwords don't match (88 ms)
- ✓ T4.23 test that form does not work if an element is missing (65 ms)
- ✓ T4.24 test that the rotator buttons change the text on the screen (53 ms)

PASS src/shared/hooks/useLogin.test.ts

useLogin

- ✓ T3.10 should handle login successfully (114 ms)
- ✓ T3.10 should handle login when it errors (13 ms)

PASS src/shared/hooks/useDeleteFile.test.ts

useDeleteFile

- ✓ T3.17 should handle delete successfully (97 ms)
- ✓ T3.18 should handle delete when it errors (11 ms)

PASS src/components/LoadingSpinner/LoadingSpinner.test.tsx

LoadingSpinner

- ✓ T4.8 renders the HashLoader with correct props when loading is true (35 ms)
- ✓ T4.9 does not render the HashLoader when loading is false (7 ms)

PASS src/shared/hooks/useChangePassword.test.ts

useChangePassword

- ✓ T5.107 should handle password change successfully (40 ms)
- ✓ T5.108 should handle password change when it errors (8 ms)

PASS src/shared/hooks/usePredictFile.test.ts

usePredictFile

- ✓ should handle predict successfully (91 ms)
- ✓ should handle predict when it errors (9 ms)

PASS src/shared/hooks/useDeleteAccount.test.ts

useDeleteAccount

- ✓ T5.115 should handle delete successfully (45 ms)
- ✓ T5.116 should handle delete when it errors (8 ms)

PASS src/shared/hooks/useLogout.test.ts

useLogout

- ✓ T3.12 should handle logout successfully (48 ms)
- ✓ T3.13 should handle logout when it errors (20 ms)

PASS src/components/Authentication/useAuth.test.tsx

useAuth

- ✓ T4.36: returns isAuthenticated as true when user is logged in (32 ms)
- ✓ T4.37: returns isAuthenticated as false when user is not logged in (6 ms)

PASS src/shared/api/Api.test.tsx

API Tests

- ✓ T3.2 Login: should return mapped data upon successful login (7 ms)
- ✓ T3.3 Login: should throw an error upon failed login (20 ms)
- ✓ T3.4 Logout: should logout successfully (1 ms)
- ✓ T3.5 Logout: should throw an error if fails (2 ms)
- ✓ T3.6 should sign up successfully (2 ms)
- ✓ T3.7 should throw an error if sign up fails (3 ms)
- ✓ T5.120 should get user successfully (1 ms)
- ✓ T5.121 should throw an error if get user fails (2 ms)
- ✓ T5.122 should delete account successfully (1 ms)
- ✓ T5.123 should throw an error if deleting account fails (1 ms)
- ✓ T5.124 should delete data successfully
- ✓ T5.125 should throw an error if deleting data fails (1 ms)
- ✓ T5.126 should change password successfully (1 ms)
- ✓ T5.127 should throw an error if change password fails (1 ms)

PASS src/shared/Data/index.test.ts

Data API Tests

- ✓ T4.38 Upload: should upload a file (8 ms)
- ✓ T4.39 Process: should process a file (2 ms)
- ✓ T4.40 Predict: should predict a file (1 ms)
- ✓ T4.41 Download: should download a file (2 ms)
- ✓ T4.42 Download: should download a undefined file (1 ms)
- ✓ T4.43 Delete: should delete a file (9 ms)
- ✓ T4.44 GetUploadedFiles: get if file exist (2 ms)
- ✓ T4.45 GetUploadedFiles: get if file does not exist (1 ms)
- ✓ T4.46 GetPredictedDataList: get if file exist (1 ms)
- ✓ T4.47 GetPredictedDataList: get if file does not exist (1 ms)
- ✓ T4.48 GetProcessedDataList: get if file exist (1 ms)
- ✓ T4.49 GetProcessedDataList: get if file does not exist

PASS src/shared/hooks/useDeleteData.test.ts

useDeleteData

- ✓ T5.118 should handle delete successfully (34 ms)
- ✓ T5.119 should handle delete when it errors (13 ms)

PASS src/shared/hooks/useSignup.test.ts

useSignup

✓ T3.14 should handle signup successfully (71 ms)

✓ T3.15 should handle signup when it errors (7 ms)

Test Suites: 43 passed, 43 total

Tests: 144 passed, 144 total

Snapshots: 0 total

Time: 23.379 s

Ran all test suites.

End 2 End Playwright tests (Ran of 2024/04/02)

Running 39 tests using 8 workers

- ✓ 1 [chromium] › Navbar.spec.ts:54:9 › T4.90 individual Navbar tests from homepage › Homepage to Processed Files (9.5s)
- ✓ 2 [chromium] › FileUploadPage.spec.ts:35:5 › T4.8 Test Uploading Fitbit Files (19.6s)
- ✓ 3 [chromium] › Navbar.spec.ts:48:9 › T4.90 individual Navbar tests from homepage › Homepage to File Upload (9.6s)
- ✓ 4 [chromium] › HomePage.spec.ts:3:5 › T2.7 Home Page Test (2.1s)
- ✓ 5 [chromium] › FileUploadPage.spec.ts:7:5 › T2.8 File Upload Page Test (9.6s)
- ✓ 6 [chromium] › Navbar.spec.ts:8:5 › T2.6 Navbar Test (11.5s)
- ✓ 7 [chromium] › FileUploadPage.spec.ts:109:5 › T4.9 Test Uploading Applewatch Files (19.6s)
- ✓ 8 [chromium] › Navbar.spec.ts:60:9 › T4.90 individual Navbar tests from homepage › Homepage to Predicted Files (9.4s)
- ✓ 9 [chromium] › Navbar.spec.ts:66:9 › T4.90 individual Navbar tests from homepage › Homepage to Profile (8.6s)
- ✓ 10 [chromium] › Navbar.spec.ts:73:9 › T4.90 individual Navbar tests from homepage › Homepage to Logout (9.6s)
- ✓ 11 [chromium] › PredictedDataPage.spec.ts:4:5 › T2.10 Predicted Data Page Test (8.4s)
- ✓ 12 [chromium] › ProcessDataPage.spec.ts:4:5 › T2.9 Processed Files Page Test (8.5s)
- ✓ 13 [chromium] › SmokeTest.spec.ts:4:5 › smoke test @smoke (11.6s)
- ✓ 14 [firefox] › FileUploadPage.spec.ts:7:5 › T2.8 File Upload Page Test (11.2s)
- ✓ 15 [firefox] › FileUploadPage.spec.ts:35:5 › T4.8 Test Uploading Fitbit Files (21.0s)
- ✓ 16 [firefox] › FileUploadPage.spec.ts:109:5 › T4.9 Test Uploading Applewatch Files (21.6s)
- ✓ 17 [firefox] › HomePage.spec.ts:3:5 › T2.7 Home Page Test (4.6s)
- ✓ 18 [firefox] › Navbar.spec.ts:8:5 › T2.6 Navbar Test (12.8s)
- ✓ 19 [firefox] › Navbar.spec.ts:48:9 › T4.90 individual Navbar tests from homepage › Homepage to File Upload (11.1s)
- ✓ 20 [firefox] › Navbar.spec.ts:54:9 › T4.90 individual Navbar tests from homepage › Homepage to Processed Files (11.6s)
- ✓ 21 [firefox] › Navbar.spec.ts:60:9 › T4.90 individual Navbar tests from homepage › Homepage to Predicted Files (10.9s)
- ✓ 22 [firefox] › Navbar.spec.ts:66:9 › T4.90 individual Navbar tests from homepage › Homepage to Profile (9.8s)
- ✓ 23 [firefox] › Navbar.spec.ts:73:9 › T4.90 individual Navbar tests from homepage › Homepage to Logout (9.9s)
- ✓ 24 [firefox] › PredictedDataPage.spec.ts:4:5 › T2.10 Predicted Data Page Test (9.6s)

- ✓ 25 [firefox] › ProcessDataPage.spec.ts:4:5 › T2.9 Processed Files Page Test (9.9s)
- ✓ 26 [firefox] › SmokeTest.spec.ts:4:5 › smoke test @smoke (15.7s)
- ✓ 27 [webkit] › FileUploadPage.spec.ts:7:5 › T2.8 File Upload Page Test (10.7s)
- ✓ 28 [webkit] › FileUploadPage.spec.ts:35:5 › T4.8 Test Uploading Fitbit Files (19.8s)
- ✓ 29 [webkit] › FileUploadPage.spec.ts:109:5 › T4.9 Test Uploading Applewatch Files (20.3s)
- ✓ 30 [webkit] › HomePage.spec.ts:3:5 › T2.7 Home Page Test (2.3s)
- ✓ 31 [webkit] › Navbar.spec.ts:8:5 › T2.6 Navbar Test (11.4s)
- ✓ 32 [webkit] › Navbar.spec.ts:48:9 › T4.90 individual Navbar tests from homepage › Homepage to File Upload (9.4s)
- ✓ 33 [webkit] › Navbar.spec.ts:54:9 › T4.90 individual Navbar tests from homepage › Homepage to Processed Files (9.6s)
- ✓ 34 [webkit] › Navbar.spec.ts:60:9 › T4.90 individual Navbar tests from homepage › Homepage to Predicted Files (9.6s)
- ✓ 35 [webkit] › Navbar.spec.ts:66:9 › T4.90 individual Navbar tests from homepage › Homepage to Profile (9.6s)
- ✓ 36 [webkit] › PredictedDataPage.spec.ts:4:5 › T2.10 Predicted Data Page Test (9.4s)
- ✓ 37 [webkit] › Navbar.spec.ts:73:9 › T4.90 individual Navbar tests from homepage › Homepage to Logout (9.5s)
- ✓ 38 [webkit] › ProcessDataPage.spec.ts:4:5 › T2.9 Processed Files Page Test (9.1s)
- ✓ 39 [webkit] › SmokeTest.spec.ts:4:5 › smoke test @smoke (12.7s)

39 passed (1.1m)

Backend(Ran on 2024/04/02):

Console Output:

```
2024-04-02 23:16:57 INFO AccessGroupController:119 - in
AccessGroupController/accessgroup/{1} GET method
2024-04-02 23:16:57 INFO AccessGroupController:75 - in
AccessGroupController/accessgroup POST method
2024-04-02 23:16:57 INFO AccessGroupController:75 - in
AccessGroupController/accessgroup POST method
2024-04-02 23:16:57 INFO AccessGroupController:98 - in
AccessGroupController/accessgroup PUT method
2024-04-02 23:16:57 INFO AccessGroupController:54 - in
AccessGroupController/accessgroup GET method
2024-04-02 23:16:57 INFO AccessGroupController:119 - in
AccessGroupController/accessgroup/{1} GET method
2024-04-02 23:16:57 INFO AccessGroupController:140 - in
AccessGroupController/accessgroup/{1} DELETE method
2024-04-02 23:16:57 INFO AccessGroupController:140 - in
AccessGroupController/accessgroup/{1} DELETE method
2024-04-02 23:16:57 INFO AccessGroupController:54 - in
AccessGroupController/accessgroup GET method
2024-04-02 23:16:57 INFO AccessGroupController:98 - in
AccessGroupController/accessgroup PUT method
*****post initializaiton
```

2024-04-02 23:16:57 INFO UserController:163 - in UserController/user/{1} GET method
2024-04-02 23:16:57 INFO UserController:59 - in UserController/user POST method
2024-04-02 23:16:57 INFO UserController:123 - in UserController/user GET method
2024-04-02 23:16:57 INFO UserController:143 - in UserController/user PUT method
2024-04-02 23:16:57 INFO UserController:163 - in UserController/user/{1} GET method
2024-04-02 23:16:57 INFO UserController:184 - in /user GET method
2024-04-02 23:16:57 INFO UserController:296 - in UserController/user/deleteUserData method
2024-04-02 23:16:57 INFO UserController:86 - in /password{ 123} for user id POST method
2024-04-02 23:16:57 INFO UserController:101 - in UserController: user found and attempting to update password to 123
2024-04-02 23:16:57 INFO UserController:230 - in UserController/user/{1} DELETE method
2024-04-02 23:16:57 INFO UserController:230 - in UserController/user/{1} DELETE method
2024-04-02 23:16:57 INFO UserController:253 - in UserController/user/deleteProfile method
2024-04-02 23:16:57 INFO UserController:143 - in UserController/user PUT method
2024-04-02 23:16:57 INFO UserController:59 - in UserController/user POST method
2024-04-02 23:16:57 INFO UserController:123 - in UserController/user GET method
2024-04-02 23:16:57 INFO PredictedDataDao:172 - In PredictedDataDao: delete
2024-04-02 23:16:57 WARN PredictedDataDao:182 - No linkage to processed data found for Predicted data with id: 3 to delete
2024-04-02 23:16:57 INFO PredictedDataDao:172 - In PredictedDataDao: delete
2024-04-02 23:16:57 INFO PredictedDataDao:179 - Deleted relation between PredictedData with id: 3 and its corresponding processed data
2024-04-02 23:16:57 INFO PredictedDataDao:145 - In PredictedDataDao: delete
2024-04-02 23:16:57 INFO PredictedDataDao:172 - In PredictedDataDao: delete
2024-04-02 23:16:57 WARN PredictedDataDao:182 - No linkage to processed data found for Predicted data with id: 3 to delete
2024-04-02 23:16:57 WARN PredictedDataDao:160 - No PredictedData found with id: 3 to delete
2024-04-02 23:16:57 INFO PredictedDataDao:145 - In PredictedDataDao: delete
2024-04-02 23:16:57 INFO PredictedDataDao:172 - In PredictedDataDao: delete
2024-04-02 23:16:57 INFO PredictedDataDao:179 - Deleted relation between PredictedData with id: 3 and its corresponding processed data
2024-04-02 23:16:57 INFO PredictedDataDao:157 - Deleted PredictedData with id: 3
2024-04-02 23:16:57 INFO ProcessedDataDao:204 - In ProcessedDataDao: delete
[]
2024-04-02 23:16:57 INFO ProcessedDataDao:223 - No predicted Data found linked to processed data of id: 3
2024-04-02 23:16:57 WARN ProcessedDataDao:235 - No ProcessedData found with id: 3 to delete
2024-04-02 23:16:57 INFO ProcessedDataDao:204 - In ProcessedDataDao: delete
[]
2024-04-02 23:16:57 INFO ProcessedDataDao:223 - No predicted Data found linked to processed data of id: 3
2024-04-02 23:16:57 INFO ProcessedDataDao:232 - Deleted ProcessedData with id: 3
2024-04-02 23:16:57 INFO ProcessedDataDao:204 - In ProcessedDataDao: delete
[1]

2024-04-02 23:16:57 INFO ProcessedDataDao:215 - found predicted Data linked to processed data of id: 3
2024-04-02 23:16:57 INFO ProcessedDataDao:232 - Deleted ProcessedData with id: 3
2024-04-02 23:16:57 INFO RawDataDao:202 - In RawDataDao: deleteRelationToUser
2024-04-02 23:16:57 WARN RawDataDao:212 - No linkage to user found for raw data with id: 3 to delete
2024-04-02 23:16:57 INFO RawDataDao:174 - In RawDataDao: delete
2024-04-02 23:16:57 INFO RawDataDao:202 - In RawDataDao: deleteRelationToUser
2024-04-02 23:16:57 WARN RawDataDao:212 - No linkage to user found for raw data with id: 3 to delete
2024-04-02 23:16:57 WARN RawDataDao:189 - No raw data found with id: 3 to delete
2024-04-02 23:16:57 INFO RawDataDao:202 - In RawDataDao: deleteRelationToUser
2024-04-02 23:16:57 INFO RawDataDao:209 - Deleted relation between raw data with id: 3 and its corresponding user
2024-04-02 23:16:57 INFO RawDataDao:174 - In RawDataDao: delete
2024-04-02 23:16:57 INFO RawDataDao:202 - In RawDataDao: deleteRelationToUser
2024-04-02 23:16:57 INFO RawDataDao:209 - Deleted relation between raw data with id: 3 and its corresponding user
2024-04-02 23:16:57 INFO RawDataDao:186 - Deleted raw data with id: 3
2024-04-02 23:16:57 INFO UserDao:289 - In UserDao: deleteRelationToAccessGroup
2024-04-02 23:16:57 WARN UserDao:300 - No linkage to access group found for user with id: 3 to delete
2024-04-02 23:16:57 INFO UserDao:397 - In UserDao: deleteUserData
2024-04-02 23:16:57 INFO UserDao:413 - found raw Data linked to user of id: 1
2024-04-02 23:16:57 INFO UserDao:429 - found processed Data linked to raw data of id: 1
2024-04-02 23:16:57 INFO UserDao:339 - In UserDao: deleteUserIncorrectLoginHistory
2024-04-02 23:16:57 INFO UserDao:347 - Deleted incorrect login history for user with id: 3
2024-04-02 23:16:57 INFO UserDao:397 - In UserDao: deleteUserData
2024-04-02 23:16:57 INFO UserDao:443 - No raw Data found linked to user of id: 1
2024-04-02 23:16:57 INFO UserDao:314 - In UserDao: deleteUserLoginHistory
2024-04-02 23:16:57 WARN UserDao:325 - No login history found for user with id: 3 to delete
2024-04-02 23:16:57 INFO UserDao:264 - In UserDao: deleteRelationToRole
2024-04-02 23:16:57 INFO UserDao:272 - Deleted relation between user with id: 3 and its corresponding role
2024-04-02 23:16:57 INFO UserDao:113 - in UserDao: newUpdate
2024-04-02 23:16:57 INFO UserDao:133 - in newUpdate: found user to update with id 1
2024-04-02 23:16:57 WARN UserDao:156 - User with id: 1 was found but could not be updated
2024-04-02 23:16:57 INFO UserDao:314 - In UserDao: deleteUserLoginHistory
2024-04-02 23:16:57 INFO UserDao:322 - Deleted login history for user with id: 3
2024-04-02 23:16:57 INFO UserDao:113 - in UserDao: newUpdate
2024-04-02 23:16:57 INFO UserDao:133 - in newUpdate: found user to update with id 1
2024-04-02 23:16:57 INFO UserDao:153 - User with id: 1 was found and updated
2024-04-02 23:16:57 INFO UserDao:289 - In UserDao: deleteRelationToAccessGroup
2024-04-02 23:16:57 INFO UserDao:297 - Deleted relation between user with id: 3 and its corresponding access group
2024-04-02 23:16:57 INFO UserDao:364 - In UserDao: deleteUserAccount

2024-04-02 23:16:57 INFO UserDao:397 - In UserDao: deleteUserData
2024-04-02 23:16:57 INFO UserDao:443 - No raw Data found linked to user of id: 3
2024-04-02 23:16:57 INFO UserDao:264 - In UserDao: deleteRelationToRole
2024-04-02 23:16:57 INFO UserDao:272 - Deleted relation between user with id: 3 and its corresponding role
2024-04-02 23:16:57 INFO UserDao:289 - In UserDao: deleteRelationToAccessGroup
2024-04-02 23:16:57 INFO UserDao:297 - Deleted relation between user with id: 3 and its corresponding access group
2024-04-02 23:16:57 INFO UserDao:314 - In UserDao: deleteUserLoginHistory
2024-04-02 23:16:57 INFO UserDao:322 - Deleted login history for user with id: 3
2024-04-02 23:16:57 INFO UserDao:339 - In UserDao: deleteUserIncorrectLoginHistory
2024-04-02 23:16:57 INFO UserDao:347 - Deleted incorrect login history for user with id: 3
2024-04-02 23:16:57 INFO UserDao:382 - Deleted user with id: 3
2024-04-02 23:16:57 INFO UserDao:339 - In UserDao: deleteUserIncorrectLoginHistory
2024-04-02 23:16:57 WARN UserDao:350 - No incorrect login history found for user with id: 3 to delete
2024-04-02 23:16:57 INFO UserDao:264 - In UserDao: deleteRelationToRole
2024-04-02 23:16:57 WARN UserDao:275 - No linkage to role found for user with id: 3 to delete
2024-04-02 23:16:57 INFO UserDao:364 - In UserDao: deleteUserAccount
2024-04-02 23:16:57 INFO UserDao:397 - In UserDao: deleteUserData
2024-04-02 23:16:57 INFO UserDao:443 - No raw Data found linked to user of id: 3
2024-04-02 23:16:57 INFO UserDao:264 - In UserDao: deleteRelationToRole
2024-04-02 23:16:57 WARN UserDao:275 - No linkage to role found for user with id: 3 to delete
2024-04-02 23:16:57 INFO UserDao:289 - In UserDao: deleteRelationToAccessGroup
2024-04-02 23:16:57 WARN UserDao:300 - No linkage to access group found for user with id: 3 to delete
2024-04-02 23:16:57 INFO UserDao:314 - In UserDao: deleteUserLoginHistory
2024-04-02 23:16:57 WARN UserDao:325 - No login history found for user with id: 3 to delete
2024-04-02 23:16:57 INFO UserDao:339 - In UserDao: deleteUserIncorrectLoginHistory
2024-04-02 23:16:57 WARN UserDao:350 - No incorrect login history found for user with id: 3 to delete
2024-04-02 23:16:57 WARN UserDao:385 - No user with id: 3 to delete
2024-04-02 23:16:57 INFO AccessGroupService:90 - in AccessGroupService: get
2024-04-02 23:16:57 INFO AccessGroupService:50 - in AccessGroupService: list
2024-04-02 23:16:57 INFO AccessGroupService:63 - in AccessGroupService: save
2024-04-02 23:16:57 INFO AccessGroupService:50 - in AccessGroupService: list
2024-04-02 23:16:57 INFO AccessGroupService:90 - in AccessGroupService: get
2024-04-02 23:16:57 INFO AccessGroupService:106 - in AccessGroupService: delete
2024-04-02 23:16:57 INFO AccessGroupService:76 - in AccessGroupService: Update
2024-04-02 23:16:57 INFO LoginUserService:81 - in IncorrectLoginsService: get
2024-04-02 23:16:57 INFO LoginUserService:110 - in IncorrectLoginsService: getByUserId
2024-04-02 23:16:57 INFO LoginUserService:42 - in IncorrectLoginsService: list
2024-04-02 23:16:57 INFO LoginUserService:54 - in IncorrectLoginsService: save
2024-04-02 23:16:57 INFO LoginUserService:42 - in IncorrectLoginsService: list
2024-04-02 23:16:57 INFO LoginUserService:81 - in IncorrectLoginsService: get

2024-04-02 23:16:57 INFO LoginUserService:97 - in IncorrectLoginsService: delete
2024-04-02 23:16:57 INFO LoginUserService:67 - in IncorrectLoginsService: Update
2024-04-02 23:16:57 INFO LoginUserService:110 - in IncorrectLoginsService: getByUserId
2024-04-02 23:16:57 INFO LoginUserService:92 - in LoginUserService: get
2024-04-02 23:16:57 INFO LoginUserService:52 - in LoginUserService: list
2024-04-02 23:16:57 INFO LoginUserService:65 - in LoginUserService: save
2024-04-02 23:16:57 INFO LoginUserService:52 - in LoginUserService: list
2024-04-02 23:16:57 INFO LoginUserService:92 - in LoginUserService: get
2024-04-02 23:16:57 INFO LoginUserService:108 - in LoginUserService: delete
2024-04-02 23:16:57 INFO LoginUserService:78 - in LoginUserService: Update
2024-04-02 23:16:57 INFO PredictedDataService:79 - in PredictedDataService: delete
2024-04-02 23:16:57 INFO PredictedDataService:55 - in PredictedDataService: get
2024-04-02 23:16:57 INFO PredictedDataService:66 - in RawDataService: list
2024-04-02 23:16:57 INFO PredictedDataService:42 - in PredictedDataService: save
2024-04-02 23:16:57 INFO PredictedDataService:79 - in PredictedDataService: delete
2024-04-02 23:16:57 INFO PredictedDataService:55 - in PredictedDataService: get
2024-04-02 23:16:57 INFO ProcessedDataService:55 - in ProcessedDataService: get
2024-04-02 23:16:57 INFO ProcessedDataService:71 - in ProcessedDataService: list
2024-04-02 23:16:57 INFO ProcessedDataService:71 - in ProcessedDataService: list
2024-04-02 23:16:57 INFO ProcessedDataService:44 - in ProcessedDataService: save
2024-04-02 23:16:57 INFO ProcessedDataService:88 - in ProcessedDataService: delete
2024-04-02 23:16:57 INFO ProcessedDataService:55 - in ProcessedDataService: get
2024-04-02 23:16:57 INFO ProcessedDataService:88 - in ProcessedDataService: delete
2024-04-02 23:16:57 INFO RawDataService:72 - in RawDataService: list
2024-04-02 23:16:57 INFO RawDataService:56 - in RawDataService: get
2024-04-02 23:16:57 INFO RawDataService:66 - in RawDataService: getProcessDataId
2024-04-02 23:16:57 INFO RawDataService:72 - in RawDataService: list
2024-04-02 23:16:57 INFO RawDataService:40 - in RawDataService: save
2024-04-02 23:16:57 INFO RawDataService:88 - in PredictedDataService: delete
2024-04-02 23:16:57 INFO RawDataService:66 - in RawDataService: getProcessDataId
2024-04-02 23:16:57 INFO RawDataService:88 - in PredictedDataService: delete
2024-04-02 23:16:57 INFO RawDataService:56 - in RawDataService: get
2024-04-02 23:16:57 INFO UserService:340 - in UserService: deleteUserAccount
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login
2024-04-02 23:16:57 INFO UserService:288 - in UserService: loadUserByUsername
2024-04-02 23:16:57 INFO UserService:353 - in UserService: deleteUserData
2024-04-02 23:16:57 INFO UserService:128 - in UserService: get
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login
2024-04-02 23:16:57 INFO UserService:270 - in UserService: getUserByUsername
2024-04-02 23:16:57 INFO UserService:66 - in UserService: save
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login
2024-04-02 23:16:57 INFO UserService:270 - in UserService: getUserByUsername
2024-04-02 23:16:57 INFO UserService:312 - in UserService: loadUserDetails
2024-04-02 23:16:57 INFO UserService:56 - in UserService: list
2024-04-02 23:16:57 INFO UserService:66 - in UserService: save
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login
2024-04-02 23:16:57 INFO UserService:270 - in UserService: getUserByUsername
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login

2024-04-02 23:16:57 INFO UserService:312 - in UserService: loadUserDetails
2024-04-02 23:16:57 INFO UserService:312 - in UserService: loadUserDetails
2024-04-02 23:16:57 INFO UserService:56 - in UserService: list
2024-04-02 23:16:57 INFO UserService:128 - in UserService: get
2024-04-02 23:16:57 INFO UserService:288 - in UserService: loadUserByUsername
2024-04-02 23:16:57 INFO UserService:270 - in UserService: getUserByUsername
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login
2024-04-02 23:16:57 INFO UserService:270 - in UserService: getUserByUsername
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login
2024-04-02 23:16:57 INFO UserService:312 - in UserService: loadUserDetails
2024-04-02 23:16:57 INFO UserService:142 - in UserService: delete
2024-04-02 23:16:57 INFO UserService:340 - in UserService: deleteUserAccount
2024-04-02 23:16:57 INFO UserService:156 - in UserService: login
2024-04-02 23:16:57 INFO UserService:270 - in UserService: getUserByUsername
2024-04-02 23:16:57 INFO UserService:116 - in UserService: newUpdateUser
2024-04-02 23:16:57 INFO UserService:103 - in UserService: Update
2024-04-02 23:16:57 INFO UserService:270 - in UserService: getUserByUsername

ScreenShots with tests:

▼ ✓ AccessGroupControllerTest

- ✓ testGet
- ✓ testSaveNull
- ✓ testSave
- ✓ testUpdateNull
- ✓ testListControllerNull
- ✓ testGetNull
- ✓ testDeleteNull
- ✓ testDelete
- ✓ testListController
- ✓ testUpdate

▼ ✓ RoleControllerTest

- ✓ testGet
- ✓ testSaveNull
- ✓ testList
- ✓ testSave
- ✓ testUpdateNull
- ✓ testGetNull
- ✓ testDeleteNull
- ✓ testDelete
- ✓ testUpdate
- ✓ testListNull

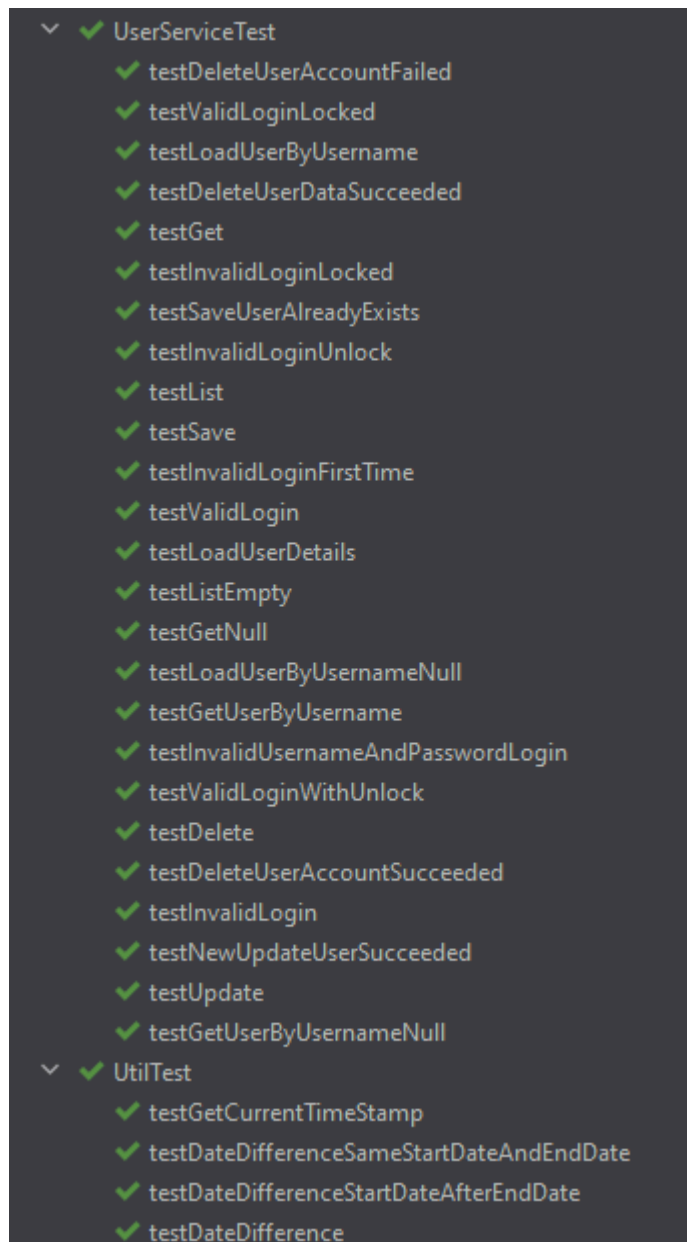
▼ ✓ UserControllerTest

- ✓ testGet
- ✓ testSaveNull
- ✓ testList
- ✓ testUpdateNull
- ✓ testGetNull
- ✓ testGetUser
- ✓ testDeleteUserData
- ✓ testUpdatePassword
- ✓ testDeleteNull
- ✓ testDelete
- ✓ testDeleteUserProfile
- ✓ testUpdate
- ✓ testSaveController
- ✓ testListNull

- ✓ WatchControllerTest
 - ✓ testDownloadFileApple
 - ✓ testListProcessedApple
 - ✓ testPredictWatchApple
 - ✓ testListProcessedFitbit
 - ✓ testDownloadFileAppleFileNotFound
 - ✓ testUploadFitbit
 - ✓ testUploadInvalidWatch
 - ✓ testListInvalidSession
 - ✓ testGetUploadViewInvalidWatch
 - ✓ testPredictWatchInvalidWatch
 - ✓ testUploadAppleNoData
 - ✓ testProcessFilesApple
 - ✓ testUploadApple
 - ✓ testUploadInvalidSession
 - ✓ testGetUploadViewAppleWatch
 - ✓ testDownloadFileFitbitFileNotFound
 - ✓ testUploadFitbitNoData
 - ✓ testGetUploadViewFitbit
 - ✓ testDownloadFileFitbit
 - ✓ testProcessFilesInvalidWatch
 - ✓ testListProcessedInvalidWatch
 - ✓ testPredictWatchFitbit
 - ✓ testProcessFilesFitbit
 - ✓ testDownloadFileInvalidWatch
- ✓ PredictedDataDaoTest
 - ✓ testDeleteRelationToProcessedDoesNotExists
 - ✓ testDeleteRelationToProcessedExists
 - ✓ testDeleteFailed
 - ✓ testDeleteSucceeded
- ✓ ProcessedDataDaoTest
 - ✓ testDeleteFailed1
 - ✓ testDeleteFailed2
 - ✓ testDeleteSucceeded
- ✓ RawDataDaoTest
 - ✓ testDeleteRelationToUserDoesNotExist
 - ✓ testDeleteFailed
 - ✓ testDeleteRelationToUserExists
 - ✓ testDeleteSucceeded

- ▼ ✓ UseDaoTest
 - ✓ testDeleteRelationToAccessGroupNotExists
 - ✓ testDeleteUserDataSucceeded
 - ✓ testDeleteUserIncorrectLoginHistoryExist
 - ✓ testDeleteUserDataFailed
 - ✓ testDeleteUserLoginHistoryNotExists
 - ✓ testDeleteRelationToRoleExists
 - ✓ testNewUpdateWhenUserDoesNotExists
 - ✓ testDeleteUserLoginHistoryExists
 - ✓ testNewUpdateWhenUserExists
 - ✓ testDeleteRelationToAccessGroupExists
 - ✓ testDeleteUserAccountExist
 - ✓ testDeleteUserIncorrectLoginHistoryNotExist
 - ✓ testDeleteRelationToRoleNotExists
 - ✓ testDeleteUserAccountNotExist
- ▼ ✓ AccessGroupServiceTest
 - ✓ testGet
 - ✓ testList
 - ✓ testSave
 - ✓ testListEmpty
 - ✓ testGetNull
 - ✓ testDelete
 - ✓ testUpdate
- ▼ ✓ IncorrectLoginsServiceTest
 - ✓ testGet
 - ✓ testGetByUserId
 - ✓ testList
 - ✓ testSave
 - ✓ testListEmpty
 - ✓ testGetNull
 - ✓ testDelete
 - ✓ testUpdate
 - ✓ testGetByUserIdNull
- ▼ ✓ LoginUserTest
 - ✓ testGet
 - ✓ testList
 - ✓ testSave
 - ✓ testListEmpty
 - ✓ testGetNull
 - ✓ testDelete
 - ✓ testUpdate

- ▼ ✓ PredictedDataServiceTest
 - ✓ testDeleteFails
 - ✓ testGet
 - ✓ testList
 - ✓ testSave
 - ✓ testDelete
 - ✓ testGetWhenPredictedDataDoesNotExist
- ▼ ✓ ProcessedDataServiceTest
 - ✓ testGet
 - ✓ testListWhenProcessedDataDoesNotExist
 - ✓ testList
 - ✓ testSave
 - ✓ testDeleteFailed
 - ✓ testGetWhenProcessedDataDoesNotExist
 - ✓ testDeleteSucceeded
- ▼ ✓ RawDataServiceTest
 - ✓ testListWhenRawDataDoesNotExist
 - ✓ testGet
 - ✓ testGetProcessDataId
 - ✓ testList
 - ✓ testSave
 - ✓ testDeleteFailed
 - ✓ testGetProcessDataIdWhenRawDataDoesNotExist
 - ✓ testDeleteSucceeded
 - ✓ testGetWhenRawDataDoesNotExist



Code Coverage Reports

Frontend:

Frontend Code Coverage Report

St. ?	Category	Percentage	Covered / Total
●	Statements	80.73% (+80.73% ▲)	733/908
●	Branches	58.1% (+58.1% ▲)	183/315
●	Functions	80.62% (+80.62% ▲)	183/227
●	Lines	81.2% (+81.2% ▲)	730/899

St. ?	File	Statements	Branches	Functions	Lines
●	... / baseapi.ts	100%	75%	100%	100%
●	shared/api/Api.ts	100%	93.75%	100%	100%
●	... / useLogin.ts	100%	100%	100%	100%
●	... / ErrorSnackbar.tsx	73.33%	66.67%	50%	73.33%
●	... / assert.ts	20%	25%	100%	20%
●	... / LoginPage.tsx	78.05%	70%	61.54%	77.5%
●	... / useSignup.ts	100%	100%	100%	100%
●	... / SignUpPage.tsx	82.81%	69.23%	70.59%	82.54%
●	... / index.ts	60%	17.65%	100%	60%
●	... / index.tsx	100%	100%	100%	100%
●	... / FileDropzone.tsx	68.97%	40%	52.94%	74.07%
●	... / PrivacyPolicy.tsx	100%	100%	100%	100%
●	... / useIsUserLoggedIn.ts	100%	100%	100%	100%
●	... / useAuth.tsx	90%	50%	100%	88.89%
●	... / HelpPopup.tsx	85.71%	56.25%	100%	85.71%
●	... / useGetUser.ts	91.67%	100%	83.33%	91.67%

●	... / useDeleteAccount.ts	100%	100%	100%	100%
●	... / useDeleteData.ts	100%	100%	100%	100%
●	... / useChangePassword.ts	100%	100%	100%	100%
●	... / confirm-modal.css	100%	100%	100%	100%
●	... / ConfirmModal.tsx	100%	75%	100%	100%
●	... / LoadingSpinner.css	100%	100%	100%	100%
●	... / LoadingSpinner.tsx	100%	100%	100%	100%
●	... / profile.css	100%	100%	100%	100%
●	... / ProfilePage.tsx	92.86%	83.33%	78.57%	92.86%
●	... / ProgressBar.tsx	93.55%	89.47%	100%	93.55%
●	... / useGetProcessedDataList.ts	100%	100%	100%	100%
●	... / usePredictFile.ts	100%	100%	100%	100%
●	... / useDownload.ts	46.88%	0%	75%	50%
●	... / ProcessedDataPage.tsx	49.32%	29.73%	66.67%	49.32%
●	... / loading-modal.css	100%	100%	100%	100%
●	... / LoadingModal.tsx	100%	100%	100%	100%
●	... / ProfileMenu.tsx	91.67%	100%	80%	91.67%
●	... / Navbar.tsx	90.91%	75%	66.67%	90.91%

●	... / useGetUploadedFiles.ts	91.67%	100%	83.33%	91.67%
●	... / useProcessFile.ts	100%	100%	100%	100%
●	... / UploadedFiles.tsx	91.11%	66.67%	100%	91.11%
●	... / HomePage.tsx	62.5%	100%	50%	62.5%
●	... / ProtectedRoute.tsx	100%	100%	100%	100%
●	... / useGetPredictedDataList.ts	100%	100%	100%	100%
●	... / Logout.tsx	100%	100%	100%	100%
●	... / AboutUs.tsx	85%	66.67%	100%	85%
●	... / useUpload.ts	100%	100%	100%	100%
●	... / useDeleteFile.ts	100%	100%	100%	100%
●	... / PredictedDataPage.tsx	87.76%	54.17%	90%	87.76%
●	... / FileUploadPage.tsx	45.45%	25%	40%	45.45%
●	... / FileDropzoneControls.tsx	100%	100%	100%	100%
●	... / GoogleLogin.tsx	0%	0%	0%	0%

Backend:

github-actions
bot
commented 36 minutes ago

Backend Code Coverage Report

Overall Project	29.85%	✖
-----------------	--------	---

There is no coverage information present for the Files changed

