Example M5: Testing and Code Review

1. Change History

Change Date	Modified Sections	Rationale
Nothing to show		

2. Back-end Test Specification: APIs

2.1. Locations of Back-end Tests and Instructions to Run Them

2.1.1. Tests (update line numbers)

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mo Cor
POST /api/journal	/Backend/tests/unmocked/journal.unmocked.test.ts#L266	/Backend/tests/mocked/journal.mocked.test.ts#L90 /Backend/tests/mocked/journal.llm.mocked.test.ts#L92	Mo Goo Tok Vali Ope Res
GET /api/journal	/Backend/tests/unmocked/journal.unmocked.test.ts#L171	/Backend/tests/mocked/journal.mocked.test.ts#L139	Mo Que Ope
PUT /api/journal	/Backend/tests/unmocked/journal.unmocked.test.ts#L362	/Backend/tests/mocked/journal.mocked.test.ts#L169	Dat Upc Ope
DELETE /api/journal	/Backend/tests/unmocked/journal.unmocked.test.ts#L405	/Backend/tests/mocked/journal.mocked.test.ts#L200	Mo Que Ope
GET /api/journal/file	/Backend/tests/unmocked/journal.unmocked.test.ts#L433	/Backend/tests/mocked/journal.mocked.test.ts#L230	Mo Que Ope
POST /api/payment-sheet	/Backend/tests/unmocked/payment.unmocked.test.ts#L57	/Backend/tests/mocked/payment.mocked.test.ts#L68	Stri Cus Eph Pay
GET /api/analytics	/Backend/tests/unmocked/analytics.unmocked.test.ts#L258	/Backend/tests/mocked/analytics.mocked.test.ts#L25	Mo Ana Que
POST /webhook	/Backend/tests/unmocked/webhook.unmocked.test.ts#L56	/Backend/tests/mocked/webhook.mocked.test.ts#L56	-
GET /api/profile	/Backend/tests/unmocked/user.unmocked.test.ts#L87	/Backend/tests/mocked/user.mocked.test.ts#L87	Fire Adr Mo Col
POST /api/profile	/Backend/tests/unmocked/user.unmocked.test.ts#L121	/Backend/tests/mocked/user.mocked.test.ts#L118	Goo API Inse
GET /api/profile/isPaid	/Backend/tests/unmocked/user.unmocked.test.ts#L199	/Backend/tests/mocked/user.mocked.test.ts#L235	-
POST /api/profile/reminder	/Backend/tests/unmocked/user.unmocked.test.ts#L231	/Backend/tests/mocked/user.mocked.test.ts#L197	Mo Upc Ope
POST /api/profile/fcmtoken	/Backend/tests/unmocked/user.unmocked.test.ts#L271	/Backend/tests/mocked/user.mocked.test.ts#L159	Fire Me (FC
POST /api/chat	/Backend/rasa_api/__tests__/server.test.js#L100	/Backend/rasa_api/__tests__/server.test.js#L30	Ras HTT me vali

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mo Cor
POST /api/chat	/Backend/rasa_api/__tests__/server.test.js#L200	/Backend/rasa_api/__tests__/server.test.js#L110	Ras Res

2.1.2. Commit Hash Where Tests Run

baf4e2bd2207d4b63d0b04665460d43b266f7613

2.1.3. Explanation on How to Run the Tests

0. To prevent some unexpected dependency issues preventing our peer group to evaluate our tests, We have recorded some videos of running the tests on our local machine. Click me to see recodings of running some tests

Backend Tests:

1. Clone the Repository: Open your terminal and run to clone:

```
git clone https://github.com/example/your-project.git
```

2. Set up environment variables by

```
export MONGODB_URI=`mongodb://mongo:27017`
export STRIPE_SECRET=`your_stripe_accuont's_secret_key`
export OPEN_API_KEY=`valid_open_ai_api_key`
export GOOGLE_USER_PREFIX= XXX
export GOOGLE_USER_ID=XXX@gmail.com
export PUBLISHABLE_STRIPE_KEY=`your_stripe_account's_publishable_key`
```

or create a new file called `.env` inside `/Backend` folder, put these variables in .env like this:

```
DB_URI=mongodb://localhost:27017
PORT=3001
OPEN_API_KEY=valid_open_ai_api_key
STRIPE_SECRET=your_stripe_accuont's_secret_key
GOOGLE_USER_PREFIX=XXX
GOOGLE_USER_ID=XXX@gmail.com
PUBLISHABLE_STRIPE_KEY=`your_stripe_account's_publishable_key
```

3. Set up variables for testing by creating a file called `unmocked_data.json` inside `/Backend/tests` folder. Put these information in it like this:

```
{
  "testGoogleToken": "this google token must be valid and corresponds to the google num id you put below",
  "googleNumID": "this field is a series of numbers of your google account numeric id",
  "OPEN_API_KEY": "valid_open_ai_api_key",
  "googleUserPrefix": "XXX",
}
```

4. In `/Backend/src/config`, put in the firebase admin json file, your stripe secret, and our server secret with names: `cpen321project-c324e-firebase-adminsdk.json`, `cpen321project-stripe-secret.txt`, and `severSecret.txt`. Contact our group to obtain the server secret.

5. Change you working directory to `/Backend`. Run the command:

```
npm test
```

Running Rasa tests

1. Ensure that your .env in file in the rasa_api directory has the following variables set:

```
{
RASA_SERVER_URL=https://ec2-54-234-28-190.compute-1.amazonaws.com:5005/webhooks/myio/webhook
ACTION_SERVER_URL=https://ec2-54-234-28-190.compute-1.amazonaws.com:5055/webhook
PORT=3001
}
```

2. Run the tests by navagating to `/Backend/rasa_api/__tests__` and Run with command:

```
PORT=4000 NODE_ENV=production npm test
```

Running Frontend tests

1. Set up the local.properties file to have:

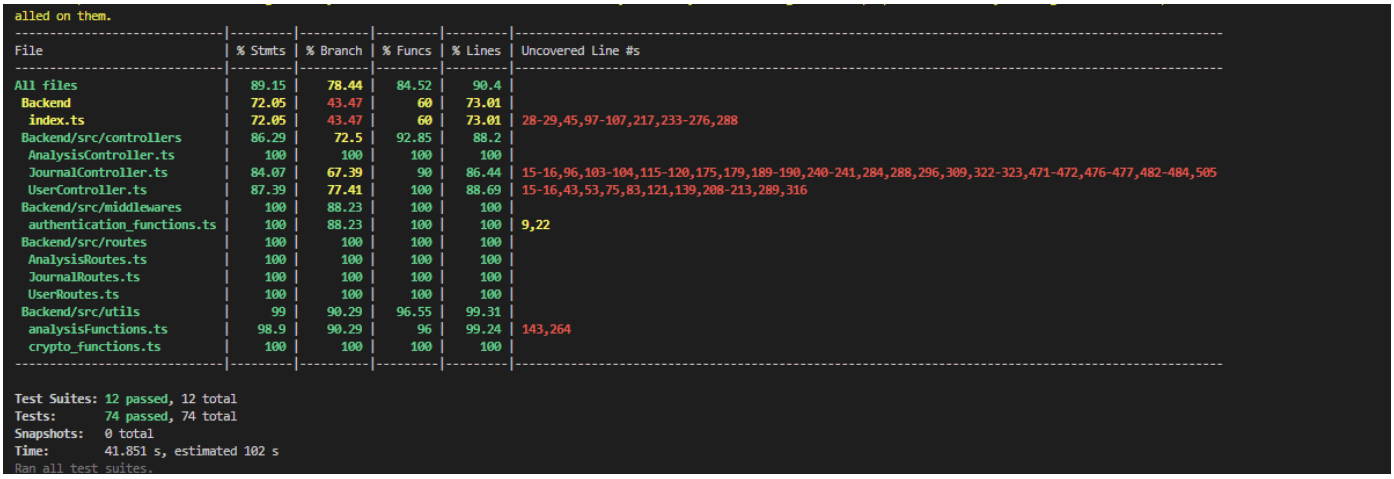
```
sdk.dir='your_path_to_android_sdk'
WEB_CLIENT_ID='google_web_client_id_for_this_project'
OPEN_API_KEY='a_valid_open_api_key'
GOOGLE_REAL_TOKEN='google_token_corresponds_to_your_google_email_this_needs_to_be_updated_1or2hours'
GOOGLE_USER_ID='your_google_email_address'
```

2. Please note that for GOOGLE_REAL_TOKEN, it needs to be updated every 1 or 2 hours to ensure the tests pass.

2.2. GitHub Actions Configuration Location

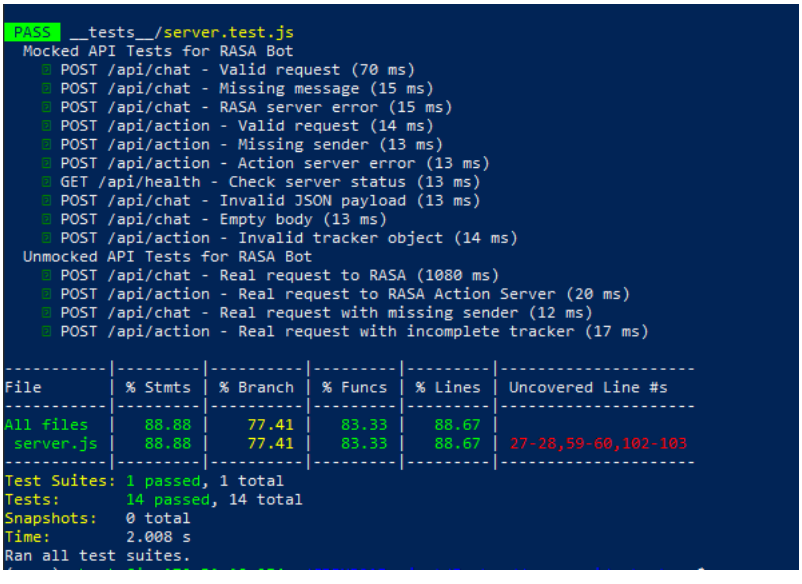
`/.github/workflows/deploy.yml`

2.3. Jest Coverage Report Screenshots With Mocks



```
alled on them.
-----------------------------|---------|----------|---------|---------|-------------------------------------------------------------------------------------------------------------
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|-------------------------------------------------------------------------------------------------------------
All files                    |   89.15 |    78.44 |   84.52 |    90.4 |
 Backend                     |   72.05 |    43.47 |      60 |   73.01 |
  index.ts                   |   72.05 |    43.47 |      60 |   73.01 | 28-29,45,97-107,217,233-276,288
 Backend/src/controllers     |   86.29 |     72.5 |   92.85 |    88.2 |
  AnalysisController.ts      |     100 |      100 |     100 |     100 |
  JournalController.ts       |   84.07 |    67.39 |      90 |   86.44 | 15-16,96,103-104,115-120,175,179,189-190,240-241,284,288,296,309,322-323,471-472,476-477,482-484,505
  UserController.ts          |   87.39 |    77.41 |     100 |   88.69 | 15-16,43,53,75,83,121,139,208-213,289,316
 Backend/src/middlewares     |     100 |    88.23 |     100 |     100 |
  authentication_functions.ts |     100 |    88.23 |     100 |     100 | 9,22
 Backend/src/routes          |     100 |      100 |     100 |     100 |
  AnalysisRoutes.ts          |     100 |      100 |     100 |     100 |
  JournalRoutes.ts           |     100 |      100 |     100 |     100 |
  UserRoutes.ts              |     100 |      100 |     100 |     100 |
 Backend/src/utils           |      99 |    90.29 |   96.55 |   99.31 |
  analysisFunctions.ts       |    98.9 |    90.29 |      96 |   99.24 | 143,264
  crypto_functions.ts        |     100 |      100 |     100 |     100 |
-----------------------------|---------|----------|---------|---------|-------------------------------------------------------------------------------------------------------------

Test Suites: 12 passed, 12 total
Tests:       74 passed, 74 total
Snapshots:   0 total
Time:        41.851 s, estimated 102 s
Ran all test suites.
```



```
PASS __tests__/server.test.js
  Mocked API Tests for RASA Bot
    ✓ POST /api/chat - Valid request (70 ms)
    ✓ POST /api/chat - Missing message (15 ms)
    ✓ POST /api/chat - RASA server error (15 ms)
    ✓ POST /api/action - Valid request (14 ms)
    ✓ POST /api/action - Missing sender (13 ms)
    ✓ POST /api/action - Action server error (13 ms)
    ✓ GET /api/health - Check server status (13 ms)
    ✓ POST /api/chat - Invalid JSON payload (13 ms)
    ✓ POST /api/chat - Empty body (13 ms)
    ✓ POST /api/action - Invalid tracker object (14 ms)
  Unmocked API Tests for RASA Bot
    ✓ POST /api/chat - Real request to RASA (1080 ms)
    ✓ POST /api/action - Real request to RASA Action Server (20 ms)
    ✓ POST /api/chat - Real request with missing sender (12 ms)
    ✓ POST /api/action - Real request with incomplete tracker (17 ms)

-----------|---------|----------|---------|---------|--------------------
File       | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------|---------|----------|---------|---------|--------------------
All files  |   88.88 |    77.41 |   83.33 |   88.67 |
 server.js |   88.88 |    77.41 |   83.33 |   88.67 | 27-28,59-60,102-103
-----------|---------|----------|---------|---------|--------------------
Test Suites: 1 passed, 1 total
Tests:       14 passed, 14 total
Snapshots:   0 total
Time:        2.008 s
Ran all test suites.
```

For coverage of journal controller, there are some lines that are implemented to support some extra actions of the application that couldn't be fully implemented end-to-end yet and this accounts for a considerable amount of the number of uncovered lines. A small portion of the uncovered lines is becasuse we don't call openapi with tests, we mock valid and invalid outputs for it. And the rest are checks for db to see if the db got values.

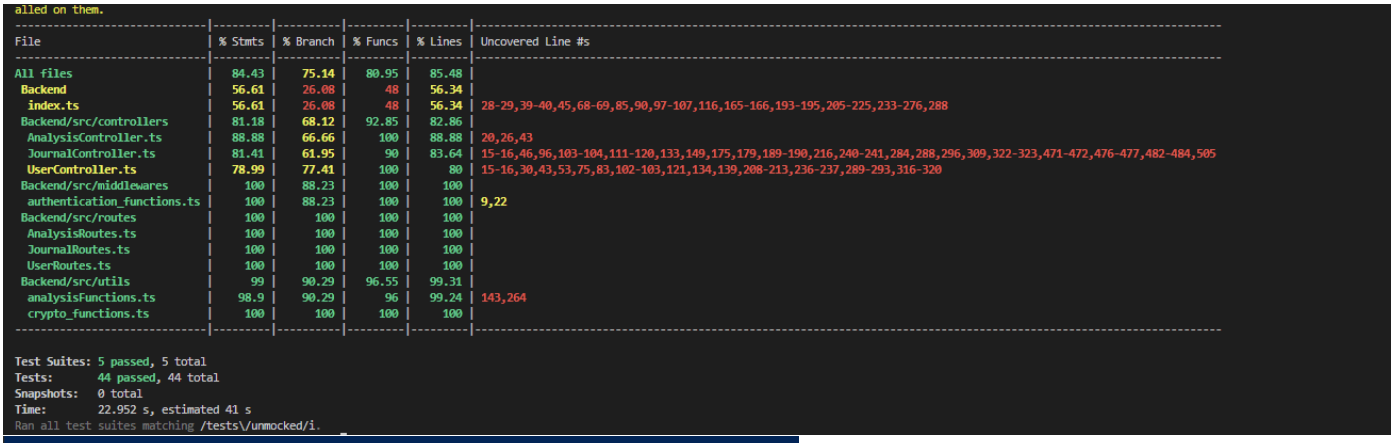
For analysisFunctions, the two uncovered lines are checks within the data structure functions made that are never supposed to be triggered by it's use, but are useful to checking for valid inputs to the functions.

For coverage of user controller, some lines (e.g. lines 15-16) it is checking Firebase initialization file. They depend on environment variables and file-based configurations. Testing this would require modifying environment variables dynamically, which is not standard for unit tests. Some lines accounts for specific time zone conversions that only acts as a safeguard against unexpected user inputs. Some lines like User not found error or checking input type are already checked by express-

validator and middleware before the controller logic runs. Since the middleware handles these errors, they do not need to be tested within the controller functions. And the rest are checks for db to see if the db got values.

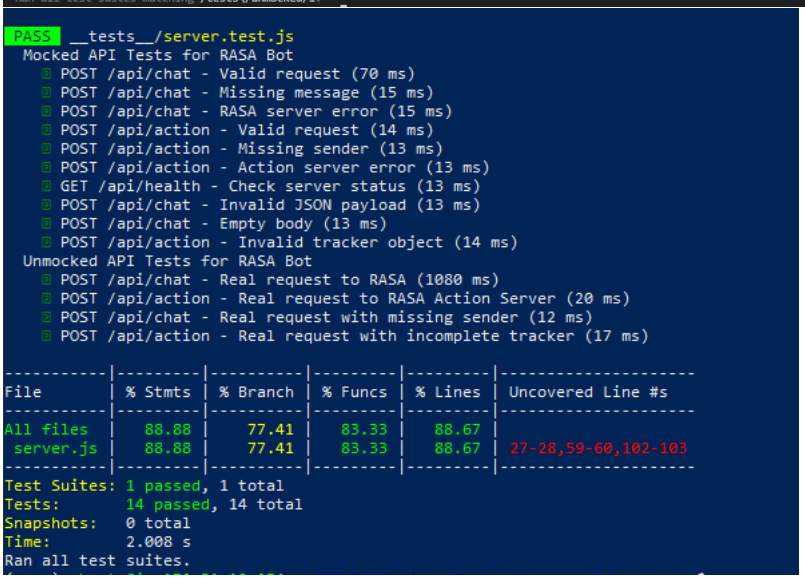
For Rasa Jest Testing, the test suite includes both mocked and unmocked tests to validate API behavior, the mocked tests include Valid requests to /api/chat and api/action, Missing parameters (e.g., message, sender), Server errors (500s), etc.. Only minor lines remain uncovered, mostly in server startup and TLS fallback logic.

2.4. Jest Coverage Report Screenshots Without Mocks



```
alled on them.
-----------------------------|---------|----------|---------|---------|-------------------------------------------------------------------------------------------------------------------------------
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|-------------------------------------------------------------------------------------------------------------------------------
All files                    |   84.43 |    75.14 |   80.95 |   85.48 |
 Backend                     |   56.61 |    26.08 |      48 |   56.34 |
  index.ts                   |   56.61 |    26.08 |      48 |   56.34 | 28-29,39-40,45,68-69,85,90,97-107,116,165-166,193-195,205-225,233-276,288
 Backend/src/controllers     |   81.18 |    68.12 |   92.85 |   82.86 |
  AnalysisController.ts      |   88.88 |    66.66 |     100 |   88.88 | 20,26,43
  JournalController.ts       |   81.41 |    61.95 |      90 |   83.64 | 15-16,46,96,103-104,111-120,133,149,175,179,189-190,216,240-241,284,288,296,309,322-323,471-472,476-477,482-484,505
  UserController.ts          |   78.99 |    77.41 |     100 |      80 | 15-16,30,43,53,75,83,102-103,121,134,139,208-213,236-237,289-293,316-320
 Backend/src/middlewares     |     100 |    88.23 |     100 |     100 |
  authentication_functions.ts|     100 |    88.23 |     100 |     100 | 9,22
 Backend/src/routes          |     100 |      100 |     100 |     100 |
  AnalysisRoutes.ts          |     100 |      100 |     100 |     100 |
  JournalRoutes.ts           |     100 |      100 |     100 |     100 |
  UserRoutes.ts              |     100 |      100 |     100 |     100 |
 Backend/src/utils           |      99 |    90.29 |   96.55 |   99.31 |
  analysisFunctions.ts       |    98.9 |    90.29 |      96 |   99.24 | 143,264
  crypto_functions.ts        |     100 |      100 |     100 |     100 |
-----------------------------|---------|----------|---------|---------|-------------------------------------------------------------------------------------------------------------------------------

Test Suites: 5 passed, 5 total
Tests:       44 passed, 44 total
Snapshots:   0 total
Time:        22.952 s, estimated 41 s
Ran all test suites matching /tests\/unmocked/i.
```



```
PASS __tests__/server.test.js
  Mocked API Tests for RASA Bot
    √ POST /api/chat - Valid request (70 ms)
    √ POST /api/chat - Missing message (15 ms)
    √ POST /api/chat - RASA server error (15 ms)
    √ POST /api/action - Valid request (14 ms)
    √ POST /api/action - Missing sender (13 ms)
    √ POST /api/action - Action server error (13 ms)
    √ GET /api/health - Check server status (13 ms)
    √ POST /api/chat - Invalid JSON payload (13 ms)
    √ POST /api/chat - Empty body (13 ms)
    √ POST /api/action - Invalid tracker object (14 ms)
  Unmocked API Tests for RASA Bot
    √ POST /api/chat - Real request to RASA (1080 ms)
    √ POST /api/action - Real request to RASA Action Server (20 ms)
    √ POST /api/chat - Real request with missing sender (12 ms)
    √ POST /api/action - Real request with incomplete tracker (17 ms)

-----------|---------|----------|---------|---------|---------------------
File       | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------|---------|----------|---------|---------|---------------------
All files  |   88.88 |    77.41 |   83.33 |   88.67 |
 server.js |   88.88 |    77.41 |   83.33 |   88.67 | 27-28,59-60,102-103
-----------|---------|----------|---------|---------|---------------------
Test Suites: 1 passed, 1 total
Tests:       14 passed, 14 total
Snapshots:   0 total
Time:        2.008 s
Ran all test suites.
```

3. Back-end Test Specification: Tests of Non-Functional Requirements

3.1. Test Locations in Git

Non-Functional Requirement	Location in Git
Performance (Response Time)	`/Backend/tests/*` (All tests for api endpoints)
Encryption of Entries	`/Backend/tests/unmocked/journal.unmocked.test.ts#L266`
Usability (Frontend)	'Frontend/app/src/androidTest/java/com/example/cpen321project/Nonfunctional_clicks_test.kt'

3.2. Test Verification and Logs

- **Journal Data Security**
  - **Verification:** This test ensures that journal entries are properly encrypted before being stored in the database. The test creates a sample journal entry through the API and retrieves the stored data directly from MongoDB. The retrieved entry is then checked to confirm that it does not match the plaintext input, indicating successful encryption. Additionally, the system is tested for correct decryption by retrieving the entry to verify that it matches the original input. This guarantees that encryption and decryption mechanisms work as expected and that journal entries remain secure in storage.
  - **Log Output**

```
console.log
Starting test: Encrypt and Retrieve Journal Entry

  at tests/unmocked/journal.unmocked.test.ts:267:17

  console.log
    Sending POST /api/journal request to create journal entry...

      at tests/unmocked/journal.unmocked.test.ts:269:17
```

```
POST /api/journal 200 74 - 61.322 ms
  console.log
    Received response: 200 {
      activities: {},
      message: 'Existing journal entry updated successfully!'
    }

      at tests/unmocked/journal.unmocked.test.ts:275:17

  console.log
    Fetching stored entry directly from Mongo database...

      at tests/unmocked/journal.unmocked.test.ts:279:17

  console.log
    Checking encryption...

      at tests/unmocked/journal.unmocked.test.ts:283:17

  console.log
    Encryption verified: Stored content does not match original input.

      at tests/unmocked/journal.unmocked.test.ts:286:17

  console.log
    Retrieving journal entry via GET /api/journal ...

      at tests/unmocked/journal.unmocked.test.ts:291:17

GET /api/journal?date=2025-03-11&userID=llcce44%40gmail.com&googleNumID=102768322270580370699 200 4824 - 60.818 ms
  console.log
    Received response: 200 {
      journal: {
        text: 'Testing...',
        media: [
          '...'
        ]
      }
    }
```

- **Performance (Response Time)**
  - **Verification:** The response time for our project is evaluated using automated test logs from all written test cases that validate API endpoints. Each test suite is executed five times, and the response time for each endpoint is averaged to obtain a reliable measurement. These tests ensure that our system meets the expected performance criteria under typical usage scenarios. The logs provide insight into the execution time of key endpoints, including journal entry management, sentiment analysis, and chatbot interactions. By leveraging real-world usage patterns captured through our test automation, we can proactively identify and address performance bottlenecks, ensuring a seamless user experience. The comprehensive logs from multiple test runs help verify that response times remain within acceptable limits, preventing delays that could impact usability.
  - **Log Output**

API Endpoint	Time to Complete
GET /api/profile?userID=***123%40gmail.com	6.887 ms
POST /api/profile	85.247 ms
POST /api/profile	43.847 ms
GET /api/profile/isPaid?userID=***test%40gmail.com	2.523 ms
POST /api/profile/reminder	16.441 ms
POST /api/profile/fcmtoken	1.100 ms
GET /api/analytics?userID=***&date=2025-01-07	20.473 ms
GET /api/analytics?userID=***&date=2024-12-31	7.368 ms
GET /api/analytics?userID=***&date=2025-01-07	6.948 ms
POST /api/journal	108.077 ms
GET /api/journal?date=2025-03-11&userID=undefined%40gmail.com&googleNumID=***	90.280 ms
POST /api/payment-sheet	1.075 ms
POST /api/journal	73.551 ms
GET /name	0.393 ms
GET /	0.227 ms

API Endpoint	Time to Complete
POST /webhook	0.910 ms
POST /webhook	0.220 ms
POST /api/chat	1080 ms
POST /api/action	20 ms

- **Usability (Frontend)**
  - **Verification:** This test simulates user interaction with the journal feature by performing multiple clicks (less than the threshold), including selecting dates, typing journal entries, sending messages, and confirming deletions. The test verifies that each step, from creating, editing and deleting journal entries, functions correctly in the app within 3 clicks as discussed din the requirements. The test also ensures that UI elements such as the chat input field, send button, delete button, and confirmation dialogs are displayed and intractable.
  - **Log Output**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 12:10:59.548	7938-7967	EspressoTest	com.example.cpen321project	D	Starting test: Usability for managing journal
2025-03-17 12:10:59.548	7938-7967	EspressoTest	com.example.cpen321project	D	Test for Creating: Clicking on an unhighlighted date, Click 1
2025-03-17 12:11:02.040	7938-7967	EspressoTest	com.example.cpen321project	D	Checking if chat input is displayed
2025-03-17 12:11:02.083	7938-7967	EspressoTest	com.example.cpen321project	D	Typing start message: Start
2025-03-17 12:11:06.285	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking send chat button. Click 2
2025-03-17 12:11:08.049	7938-7967	EspressoTest	com.example.cpen321project	D	Typing journal entry: I had a good balance between work and fun today. Days like this remind me why balance is so important.
2025-03-17 12:11:18.738	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking send chat button again. Click 3
2025-03-17 12:11:24.313	7938-7967	EspressoTest	com.example.cpen321project	D	Done in 3 clicks. Clicking back button to entries. Now delete
2025-03-17 12:11:29.143	7938-7967	EspressoTest	com.example.cpen321project	D	Checking if calendar view is displayed
2025-03-17 12:11:29.168	7938-7967	EspressoTest	com.example.cpen321project	D	Testing: Deleting
2025-03-17 12:11:30.171	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking on a highlighted date. Click 1
2025-03-17 12:11:32.967	7938-7967	EspressoTest	com.example.cpen321project	D	Checking if delete button is displayed
2025-03-17 12:11:32.975	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking delete button, click 2
2025-03-17 12:11:37.264	7938-7967	EspressoTest	com.example.cpen321project	D	Checking if confirmation dialog is displayed
2025-03-17 12:11:38.285	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking Yes to confirm deletion, Click 3
2025-03-17 12:11:42.460	7938-7967	EspressoTest	com.example.cpen321project	D	Testing: Editing
2025-03-17 12:11:43.461	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking on a highlighted date, Click 1
2025-03-17 12:11:44.879	7938-7967	EspressoTest	com.example.cpen321project	D	Checking if Save Entry button is displayed
2025-03-17 12:11:45.504	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking edit button, Click 2

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 12:11:47.414	7938-7967	EspressoTest	com.example.cpen321project	D	Typing additional text: . I also played soccer to make my day feel even better
2025-03-17 12:11:54.350	7938-7967	EspressoTest	com.example.cpen321project	D	Clicking Save Entry button, click 3
2025-03-17 12:11:58.373	7938-7967	EspressoTest	com.example.cpen321project	D	Checking if calendar view is displayed

4. Front-end Test Specification

4.1. Location in Git of Front-end Test Suite:

`Frontend/app/src/androidTest/java/com/example/cpen321project`

4.2. Tests

- **Use Case: User Click on Unhighlighted Create Entry**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on an unhighlighted date.	Click on an unhighlighted date in`calenderrecycleView`.
2. The chat input field is displayed.	Check that`chatInput` is displayed.
3. User enters a start message.	Input "Start" in`chatInput`, close keyboard, and click `sendChatButton`.
4. User enters a journal entry.	Input "I had a good balance between work and fun today. Days like this remind me why balance is so important." in`chatInput`, close keyboard, and click `sendChatButton`.
5. The entry is saved, and the calendar refreshes.	Click`Backbuttonentries`, check that `calenderrecycleView` is displayed with the date highlighted.

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 11:24:30.240	2767-2864	EspressoTest	com.example.cpen321project	D	Clicking on an unhighlighted date
2025-03-17 11:24:32.784	2767-2864	EspressoTest	com.example.cpen321project	D	Checking if chat input is displayed
2025-03-17 11:24:32.818	2767-2864	EspressoTest	com.example.cpen321project	D	Typing start message: Start
2025-03-17 11:24:36.887	2767-2864	EspressoTest	com.example.cpen321project	D	Clicking send chat button
2025-03-17 11:24:38.679	2767-2864	EspressoTest	com.example.cpen321project	D	Typing journal entry: I had a good balance between work and fun today. Days like this remind me why balance is so important.
2025-03-17 11:24:47.059	2767-2864	EspressoTest	com.example.cpen321project	D	Clicking send chat button again
2025-03-17 11:24:52.746	2767-2864	EspressoTest	com.example.cpen321project	D	Clicking back button to entries
2025-03-17 11:24:54.152	2767-2864	EspressoTest	com.example.cpen321project	D	Checking if calendar view is displayed
2025-03-17 11:24:54.937	2767-2864	EspressoTest	com.example.cpen321project	D	Checking if the selected date is highlighted
2025-03-17 11:24:55.037	2767-2864	EspressoTest	com.example.cpen321project	D	Test A_User_click_on_unhiglighted_create_entry completed successfully

- **Use Case: User Click on Highlighted Edit Entry**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps

Scenario Steps	Test Case Steps
1. User clicks on a highlighted date.	Click on a highlighted date in`calenderrecycleView`.
2. The edit button is displayed.	Check that`Saveentrybutton` is displayed.
3. User clicks edit and modifies the journal entry.	Click`editbutton`, input ". I also played soccer to make my day feel even better" in `journalEntryInput`, close keyboard, and click `Saveentrybutton`.
4. The entry is saved, and the calendar refreshes.	Check that`calenderrecycleView` is displayed with the date still highlighted.

- **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:41:30.810	31252-31279	EspressoTest	com.example.cpen321project	D	Starting test: B_User_click_on_higlighted_edit_entry
2025-03-17 10:41:31.813	31252-31279	EspressoTest	com.example.cpen321project	D	Clicking on a highlighted date
2025-03-17 10:41:33.215	31252-31279	EspressoTest	com.example.cpen321project	D	Checking if Save Entry button is displayed
2025-03-17 10:41:53.756	31252-31279	EspressoTest	com.example.cpen321project	D	Clicking edit button
2025-03-17 10:41:54.782	31252-31279	EspressoTest	com.example.cpen321project	D	Typing additional text: . I also played soccer to make my day feel even better
2025-03-17 10:41:59.951	31252-31279	EspressoTest	com.example.cpen321project	D	Clicking Save Entry button
2025-03-17 10:42:02.737	31252-31279	EspressoTest	com.example.cpen321project	D	Checking if calendar view is displayed
2025-03-17 10:42:02.752	31252-31279	EspressoTest	com.example.cpen321project	D	Verifying if the date is still highlighted
2025-03-17 10:42:02.768	31252-31279	EspressoTest	com.example.cpen321project	D	Test B_User_click_on_higlighted_edit_entry completed successfully

- **Use Case: User Click on Highlighted Delete Entry**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on a highlighted date.	Click on a highlighted date in`calenderrecycleView`.
2. The delete button is displayed.	Check that`deletebutton` is displayed.
3. User clicks delete and confirms.	Click`deletebutton`, verify "Delete Journal Entry" dialog, click "Yes".
4. The entry is deleted, and the calendar refreshes.	Check that`calenderrecycleView` is displayed, ensuring the date is no longer highlighted.

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:15:02.584	24435-24462	EspressoTest	com.example.cpen321project	D	Starting test: C_User_click_on_higlighted_delete_entry
2025-03-17 10:15:03.585	24435-24462	EspressoTest	com.example.cpen321project	D	Clicking on a highlighted date
2025-03-17 10:15:05.517	24435-24462	EspressoTest	com.example.cpen321project	D	Checking if delete button is displayed
2025-03-17 10:15:05.751	24435-24462	EspressoTest	com.example.cpen321project	D	Clicking delete button
2025-03-17 10:15:07.640	24435-24462	EspressoTest	com.example.cpen321project	D	Checking if confirmation dialog is displayed
2025-03-17 10:15:08.864	24435-24462	EspressoTest	com.example.cpen321project	D	Clicking Yes to confirm deletion

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:15:10.990	24435-24462	EspressoTest	com.example.cpen321project	D	Verifying if the date is no longer highlighted
2025-03-17 10:15:11.174	24435-24462	EspressoTest	com.example.cpen321project	D	Test C_User_click_on_higlighted_delete_entry completed successfully

- **Use Case: User Clicks Future Date**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User navigates to the next month.	Click`Next_month_button`.
2. User clicks on a future date.	Click on a date in`calenderrecycleView`.
3. The app displays a toast message.	Verify "Cannot add a journal for future dates!" toast message.

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:15:12.770	24435-24462	EspressoTest	com.example.cpen321project	D	Starting test: D_User_clicks_future_date
2025-03-17 10:15:14.186	24435-24462	EspressoTest	com.example.cpen321project	D	Clicking on a future date
2025-03-17 10:15:15.504	24435-24462	EspressoTest	com.example.cpen321project	D	Checking if toast message is displayed
2025-03-17 10:15:15.509	24435-24462	EspressoTest	com.example.cpen321project	D	Test D_User_clicks_future_date completed successfully

- **Use Case: User Cannot Upload Image**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on a date.	Click on a date in`calenderrecycleView`.
2. User tries to upload an image.	Click`addimageButton`.
3. The app displays a toast message.	Verify "Upgrade to upload media!" toast message.

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:15:17.433	24435-24462	EspressoTest	com.example.cpen321project	D	Starting test: E_User_cannot_upload_image
2025-03-17 10:15:19.197	24435-24462	EspressoTest	com.example.cpen321project	D	Clicking add image button
2025-03-17 10:15:20.769	24435-24462	EspressoTest	com.example.cpen321project	D	Checking if upgrade toast message is displayed
2025-03-17 10:15:20.770	24435-24462	EspressoTest	com.example.cpen321project	D	Test E_User_cannot_upload_image completed successfully

- **Use Case: User Exports Journal**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks export.	Click`export_button`.
2. The app displays a confirmation message.	Verify "File URL copied to clipboard!" toast message.

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:15:23.259	24435-24462	EspressoTest	com.example.cpen321project	D	Starting test: F_User_export_journal
2025-03-17 10:15:23.368	24435-24462	EspressoTest	com.example.cpen321project	D	Clicking export button
2025-03-17 10:15:25.153	24435-24462	EspressoTest	com.example.cpen321project	D	Checking if file copied toast message is displayed
2025-03-17 10:15:25.159	24435-24462	EspressoTest	com.example.cpen321project	D	Test F_User_export_journal completed successfully

- **Use Case: Paid User Uploads an Image (Camera)**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on a date.	Click on a date in`calenderrecycleView`.
2. User clicks the add image button.	Click`addimageButton`.
3. Upload media popup appears.	Check that "Upload Media" popup is displayed.
4. User selects "Take a Photo".	Click "Take a Photo".

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:34:50.735	29605-29633	EspressoTest	com.example.cpen321project	D	Starting test: A_User_Upload_Image_popup_check_camera
2025-03-17 10:34:50.735	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on date 1 in calendar
2025-03-17 10:34:52.978	29605-29633	EspressoTest	com.example.cpen321project	D	Checking if add image button is displayed
2025-03-17 10:34:54.249	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on add image button
2025-03-17 10:34:54.936	29605-29633	EspressoTest	com.example.cpen321project	D	Checking if Upload Media popup is displayed
2025-03-17 10:34:54.949	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on Take a Photo option
2025-03-17 10:34:55.274	29605-29633	EspressoTest	com.example.cpen321project	D	Test A_User_Upload_Image_popup_check_camera completed successfully

- **Use Case: Paid User Uploads an Image (Gallery)**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on a date.	Click on a date in`calenderrecycleView`.
2. User clicks the add image button.	Click`addimageButton`.
3. Upload media popup appears.	Check that "Upload Media" popup is displayed.
4. User selects "Select from Gallery".	Click "Select from Gallery".

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:34:56.815	29605-29633	EspressoTest	com.example.cpen321project	D	Starting test: B_User_Upload_Image_popup_check_device
2025-03-17 10:34:56.816	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on date 1 in calendar
2025-03-17 10:34:58.551	29605-29633	EspressoTest	com.example.cpen321project	D	Checking if add image button is displayed

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:34:59.567	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on add image button
2025-03-17 10:35:00.165	29605-29633	EspressoTest	com.example.cpen321project	D	Checking if Upload Media popup is displayed
2025-03-17 10:35:00.174	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on Select from Gallery option
2025-03-17 10:35:00.739	29605-29633	EspressoTest	com.example.cpen321project	D	Test B_User_Upload_Image_popup_check_device completed successfully

- **Use Case: Paid User Deletes an Image**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on a date with an existing image.	Click on a date in`calenderrecycleView`.
2. User clicks on the journal image.	Click`journalImageView`.
3. Delete image popup appears.	Check that "Delete Image" popup is displayed.
4. User confirms deletion.	Click "Delete".

  - **Test Logs:**

Timestamp	Process ID	Test Suite	Package Name	Log Level	Description
2025-03-17 10:35:02.147	29605-29633	EspressoTest	com.example.cpen321project	D	Starting test: C_User_Deletes_existing_Image_popup_check
2025-03-17 10:35:02.147	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on date 12 in calendar
2025-03-17 10:35:03.508	29605-29633	EspressoTest	com.example.cpen321project	D	Checking if add image button is displayed
2025-03-17 10:35:04.521	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking on journal image view
2025-03-17 10:35:05.282	29605-29633	EspressoTest	com.example.cpen321project	D	Checking if Delete Image popup is displayed
2025-03-17 10:35:06.293	29605-29633	EspressoTest	com.example.cpen321project	D	Clicking Delete button
2025-03-17 10:35:06.643	29605-29633	EspressoTest	com.example.cpen321project	D	Test C_User_Deletes_existing_Image_popup_check completed successfully

- **Use Case: User Checks Analytics for Emotions**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on analytics button.	Click`analytics_button`.
2. Emotion filter button is displayed.	Check that`emotionFilterButton` is displayed.
3. User opens emotion filter.	Click`emotionFilterButton`.
4. User selects "Joy" and "Sadness".	Click "Joy", then click "Sadness".
5. User applies the filter.	Click "Apply".
6. The analytics chart updates.	Check that`analyticsChart` is displayed.

  - **Test Logs:**

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 10:03:05.619	22690	22779	EspressoTest	com.example.cpen321project	D	Starting test: User_Analytics_check_emotions
2025-03-17 10:03:05.619	22690	22779	EspressoTest	com.example.cpen321project	D	Clicking on Analytics button

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 10:03:07.069	22690	22779	EspressoTest	com.example.cpen321project	D	Waiting for Analytics screen to load
2025-03-17 10:03:17.070	22690	22779	EspressoTest	com.example.cpen321project	D	Checking if emotion filter button is displayed
2025-03-17 10:03:17.092	22690	22779	EspressoTest	com.example.cpen321project	D	Clicking on emotion filter button
2025-03-17 10:03:18.052	22690	22779	EspressoTest	com.example.cpen321project	D	Selecting emotions: Joy and Sadness
2025-03-17 10:03:18.761	22690	22779	EspressoTest	com.example.cpen321project	D	Clicking Apply button
2025-03-17 10:03:19.285	22690	22779	EspressoTest	com.example.cpen321project	D	Checking if analytics chart is displayed
2025-03-17 10:03:19.308	22690	22779	EspressoTest	com.example.cpen321project	D	Test User_Analytics_check_emotions completed successfully

- **Use Case: User Checks Analytics for Activities**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User clicks on analytics button.	Click`analytics_button`.
2. Activity filter button is displayed.	Check that`activityfilterButton` is displayed.
3. User opens activity filter.	Click`activityfilterButton`.
4. User selects "Sleep" and "Sadness".	Click "Sleep", then click "Sadness".
5. User applies the filter.	Click "Apply".
6. The activities chart updates.	Check that`activities_chart` is displayed.

  - **Test Logs:**

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 10:03:21.658	22690	22779	EspressoTest	com.example.cpen321project	D	Starting test: User_Analytics_check_activities
2025-03-17 10:03:21.658	22690	22779	EspressoTest	com.example.cpen321project	D	Clicking on Analytics button
2025-03-17 10:03:22.459	22690	22779	EspressoTest	com.example.cpen321project	D	Waiting for Analytics screen to load
2025-03-17 10:03:42.462	22690	22779	EspressoTest	com.example.cpen321project	D	Checking if activity filter button is displayed
2025-03-17 10:03:42.477	22690	22779	EspressoTest	com.example.cpen321project	D	Clicking on activity filter button
2025-03-17 10:03:44.314	22690	22779	EspressoTest	com.example.cpen321project	D	Selecting activities: Sleep and Walk
2025-03-17 10:03:45.486	22690	22779	EspressoTest	com.example.cpen321project	D	Clicking Apply button
2025-03-17 10:03:46.931	22690	22779	EspressoTest	com.example.cpen321project	D	Checking if activities chart is displayed
2025-03-17 10:03:46.945	22690	22779	EspressoTest	com.example.cpen321project	D	Test User_Analytics_check_activities completed successfully

- **Use Case: Update Reminder Settings**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. The user clicks on the profile icon.	Click on the profile button.

Scenario Steps	Test Case Steps
2. The system displays the user profile and the current reminder settings. (For new users, all weekdays are set to none, and the displayed time matches the device's current time.)	Check that the profile screen is displayed. Verify that the displayed time matches the device's current time if the user has not set a reminder before.
3. The user selects a weekday and sets a reminder time.	Click on a weekday button to select it. Use the time picker to set a reminder time.
4. The user clicks "Save Settings".	Click the "Save Settings" button.
5. The system displays a success message confirming the update.	Check that a toast message with text "Reminder updated successfully!" is displayed.
6. The system ensures that the selected weekday remains highlighted after saving.	Check that the selected weekday button remains highlighted after saving.
7. The system sends a notification at the selected time.	Wait for the selected time.Check that a notification with text "Journal Reminder" appears.

- **Test Logs:**

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 12:49:41.897	18933	18965	EspressoTest	com.example.cpen321project	D	Launching main activity
2025-03-17 12:49:42.364	18933	18965	EspressoTest	com.example.cpen321project	D	Starting test: Reminder Settings
2025-03-17 12:49:44.365	18933	18965	EspressoTest	com.example.cpen321project	D	Clicking on profile button
2025-03-17 12:49:47.033	18933	18965	EspressoTest	com.example.cpen321project	D	Verifying current reminder settings
2025-03-17 12:49:52.076	18933	18965	EspressoTest	com.example.cpen321project	D	Selecting weekday for reminder
2025-03-17 12:49:52.351	18933	18965	EspressoTest	com.example.cpen321project	D	Setting reminder time to 12:50
2025-03-17 12:49:52.369	18933	18965	EspressoTest	com.example.cpen321project	D	Saving reminder settings
2025-03-17 12:49:53.660	18933	18965	EspressoTest	com.example.cpen321project	D	Verifying success toast message
2025-03-17 12:49:53.663	18933	18965	EspressoTest	com.example.cpen321project	D	Verifying selected weekday remains highlighted
2025-03-17 12:49:53.669	18933	18965	EspressoTest	com.example.cpen321project	D	Waiting for notification...
2025-03-17 12:50:56.272	18933	18965	EspressoTest	com.example.cpen321project	D	Notification verified successfully
2025-03-17 12:50:58.434	18933	18965	EspressoTest	com.example.cpen321project	D	Releasing intents

- **Use Case: Update Activities Tracking**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. The user clicks on the profile icon.	Click on the profile button.
2. The system displays the user profile and current activity tracking settings.	Check that the activity list is displayed.
3. The user adds a new activity.	Click the "Add Activity" button. Enter "Running" in the activity name field. Enter "30" in the value field. Select "Minutes" from the dropdown. Click "Add".
4. The system updates the activity list with the new entry.	Verify that the activity list contains "Running".
5. The user deletes an activity from the list.	Long press the "Running" activity.Click "Delete".
6. The system updates the activity list and removes the entry.	Verify that the activity "Running" is no longer in the list.

Scenario Steps	Test Case Steps
7. The user clicks "Save Settings".	Click the "Save Settings" button.
8. The system displays a success message confirming the update.	Check that a toast message with text "Profile updated successfully!" is displayed.

- **Test Logs:**

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 12:51:09.765	18933	18965	EspressoTest	com.example.cpen321project	D	Launching main activity
2025-03-17 12:51:09.877	18933	18965	EspressoTest	com.example.cpen321project	D	Starting test: Activity List Management
2025-03-17 12:51:11.881	18933	18965	EspressoTest	com.example.cpen321project	D	Clicking on profile button
2025-03-17 12:51:14.368	18933	18965	EspressoTest	com.example.cpen321project	D	Original activity count: 1
2025-03-17 12:51:14.368	18933	18965	EspressoTest	com.example.cpen321project	D	Adding new activity: Running
2025-03-17 12:51:18.064	18933	18965	EspressoTest	com.example.cpen321project	D	Selecting 'Minutes' from dropdown
2025-03-17 12:51:19.738	18933	18965	EspressoTest	com.example.cpen321project	D	Confirming addition
2025-03-17 12:51:22.061	18933	18965	EspressoTest	com.example.cpen321project	D	Verifying new activity exists in the list
2025-03-17 12:51:22.079	18933	18965	EspressoTest	com.example.cpen321project	D	Deleting activity: Running
2025-03-17 12:51:23.663	18933	18965	EspressoTest	com.example.cpen321project	D	Saving settings
2025-03-17 12:51:24.959	18933	18965	EspressoTest	com.example.cpen321project	D	Verifying success toast message
2025-03-17 12:51:24.961	18933	18965	EspressoTest	com.example.cpen321project	D	Test completed successfully
2025-03-17 12:51:24.962	18933	18965	EspressoTest	com.example.cpen321project	D	Releasing intents

- **Use Case: Update Preferred Name**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. The user clicks on the profile icon.	Click on the profile button.
2. The system displays the user profile and the preferred name field.	Check that the preferred name field is visible.
3. The user enters a new preferred name.	Clear the existing text in the preferred name field. Type "John Doe" into the field.
4. The user clicks "Save Settings".	Click the "Save Settings" button.
5. The system displays a success message confirming the update.	Check that a toast message with text "Profile updated successfully!" is displayed.
6. The user navigates back and reopens the profile.	Click the back button. Click the profile button again.
7. The system retains the updated preferred name.	Verify that the preferred name field contains "John Doe".

  - **Test Logs:**

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 12:50:58.616	18933	18965	EspressoTest	com.example.cpen321project	D	Launching main activity
2025-03-17 12:50:58.764	18933	18965	EspressoTest	com.example.cpen321project	D	Starting test: Update Preferred Name

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 12:51:00.768	18933	18965	EspressoTest	com.example.cpen321project	D	Clicking on profile button
2025-03-17 12:51:03.299	18933	18965	EspressoTest	com.example.cpen321project	D	Entering new preferred name: John Doe
2025-03-17 12:51:04.958	18933	18965	EspressoTest	com.example.cpen321project	D	Saving name update
2025-03-17 12:51:06.338	18933	18965	EspressoTest	com.example.cpen321project	D	Verifying success toast message
2025-03-17 12:51:06.340	18933	18965	EspressoTest	com.example.cpen321project	D	Navigating back and reopening profile to verify change
2025-03-17 12:51:09.644	18933	18965	EspressoTest	com.example.cpen321project	D	Verifying preferred name is updated
2025-03-17 12:51:09.648	18933	18965	EspressoTest	com.example.cpen321project	D	Test completed successfully
2025-03-17 12:51:09.648	18933	18965	EspressoTest	com.example.cpen321project	D	Releasing intents

- **Use Case: Paid vs. Non-Paid User Profile Access**
  - **Expected Behaviors:**

    For non-paid users:

Scenario Steps	Test Case Steps
Non-paid users:	
1. The user logs in as a non-paid user.	Set authentication state for a non-paid user. Launch the app.
2. The user clicks on the profile button.	Click the profile button.
3. The system displays an upgrade button.	Verify that the "Upgrade" button is visible.
Paid User:	Click the "Save Settings" button.
1. The user logs in as a paid user.	Set authentication state for a paid user. Launch the app.
2. The user clicks on the profile button.	Click the profile button.
3. The system does not show an upgrade button.	Verify that the "Upgrade" button is not visible.

  - **Test Logs:**

Non-Paid user Test:

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 13:05:31.626	21077	21108	EspressoTest	com.example.cpen321project	D	Launching main activity
2025-03-17 13:05:32.160	21077	21108	EspressoTest	com.example.cpen321project	D	Starting test: View Non-Paid User Profile
2025-03-17 13:05:34.946	21077	21108	EspressoTest	com.example.cpen321project	D	Waiting for user profile to load...
2025-03-17 13:05:36.946	21077	21108	EspressoTest	com.example.cpen321project	D	Checking if there is an upgrade button...
2025-03-17 13:05:36.951	21077	21108	EspressoTest	com.example.cpen321project	D	Test completed successfully

Paid user Test

Timestamp	Process ID	Thread ID	Tag	Package	Log Level	Message
2025-03-17 13:06:17.986	21389	21422	EspressoTest	com.example.cpen321project	D	Launching main activity
2025-03-17 13:06:18.443	21389	21422	EspressoTest	com.example.cpen321project	D	Starting test: View Paid User Profile
2025-03-17 13:06:21.193	21389	21422	EspressoTest	com.example.cpen321project	D	Waiting for user profile to load...
2025-03-17 13:06:23.197	21389	21422	EspressoTest	com.example.cpen321project	D	Checking if there is no upgrade button...
2025-03-17 13:06:23.210	21389	21422	EspressoTest	com.example.cpen321project	D	Test completed successfully

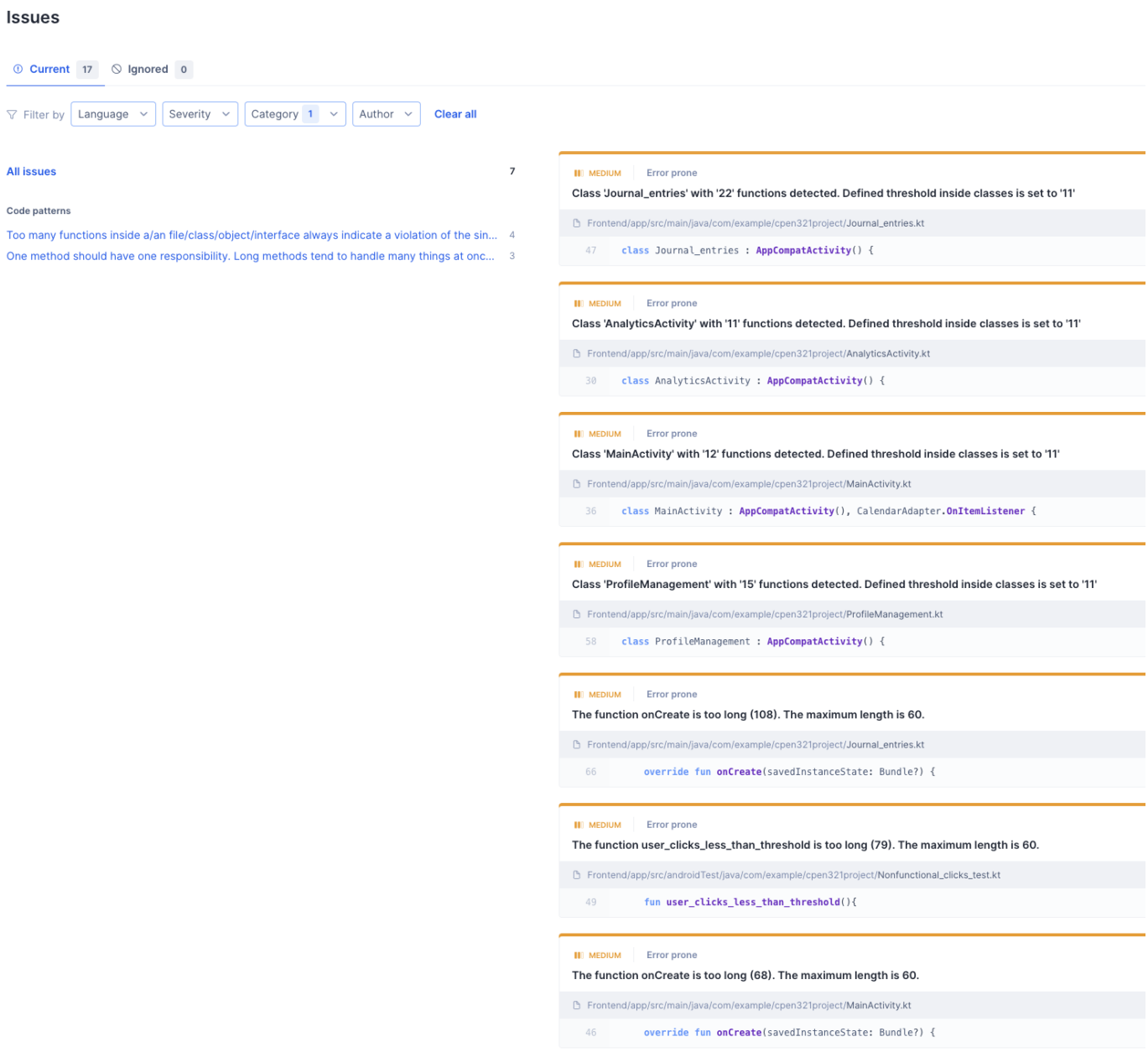
5. Automated Code Review Results

5.1. Commit Hash Where Codacy Ran

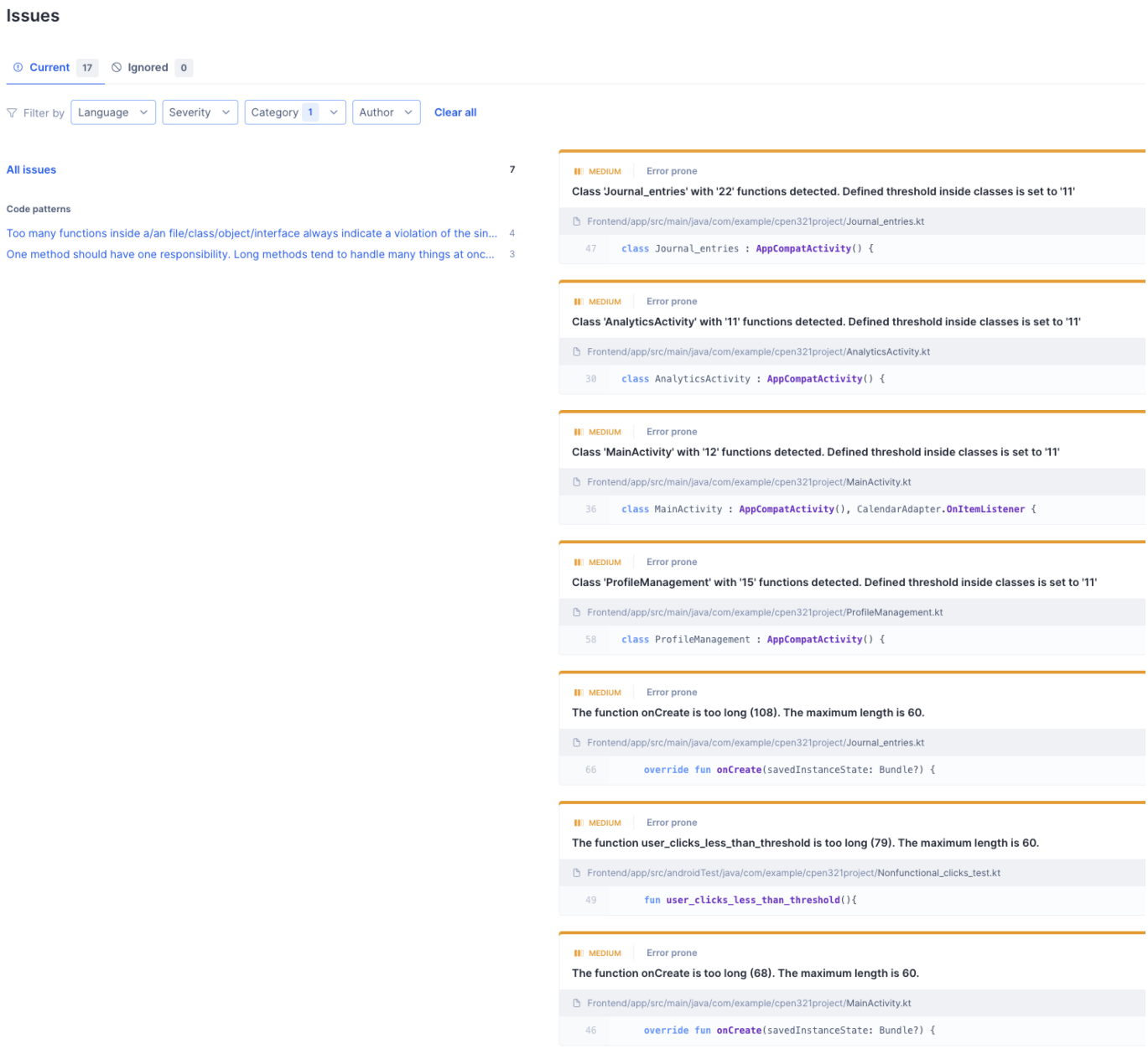
f579e43beef4743617ee236638461d7b239306e0

5.2. Unfixed Issues per Codacy Category

Issue category: Errorprone

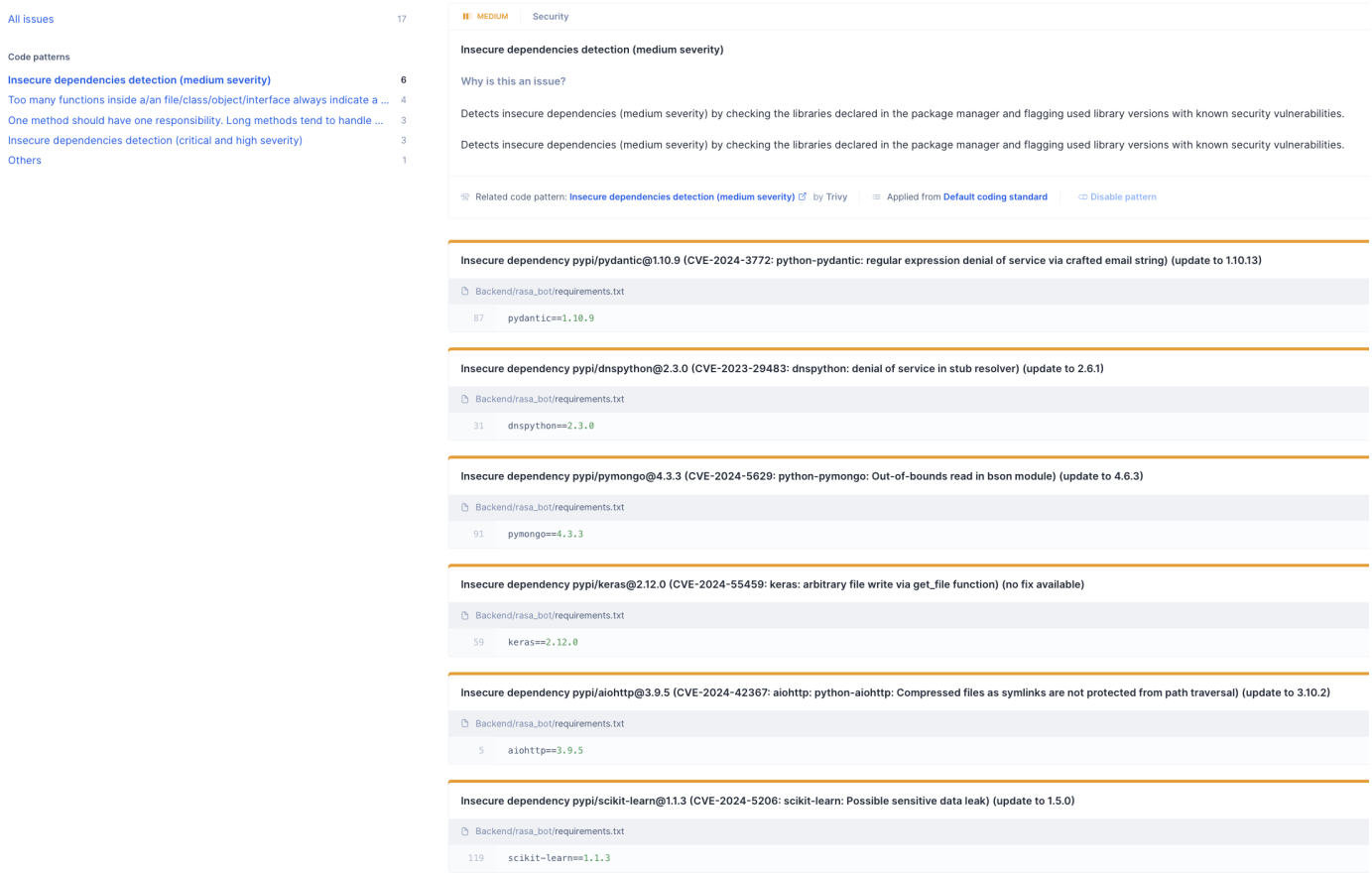


Issue category: Security

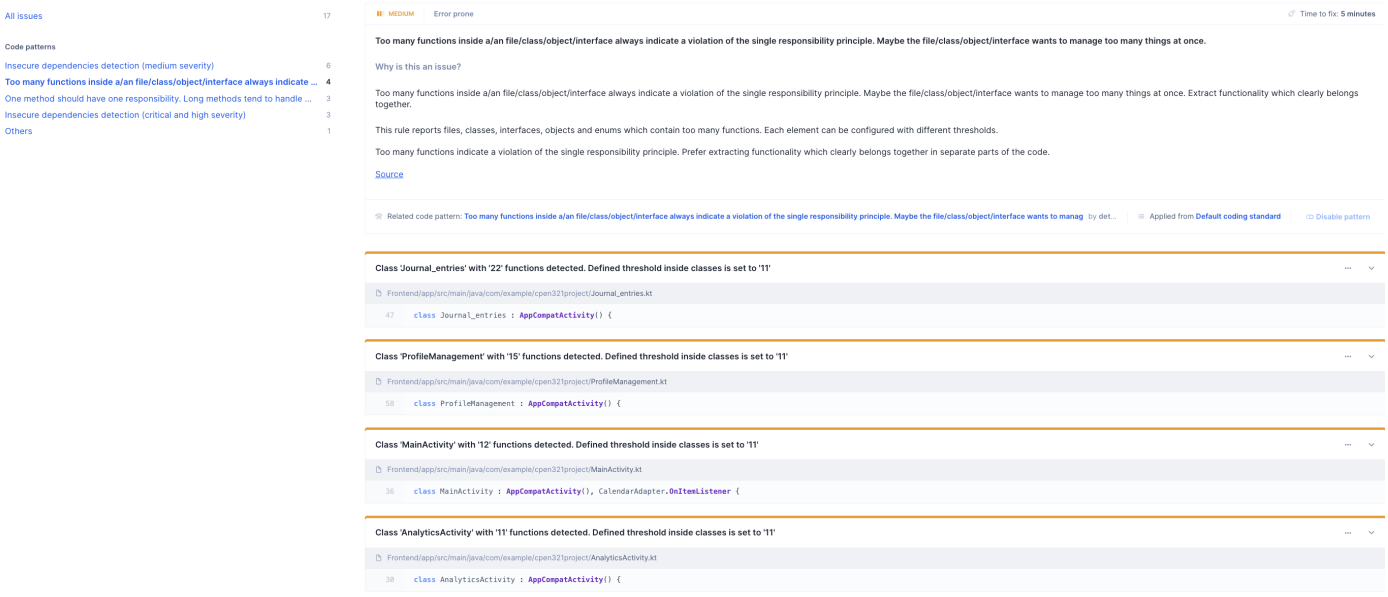


5.3. Unfixed Issues per Codacy Code Pattern

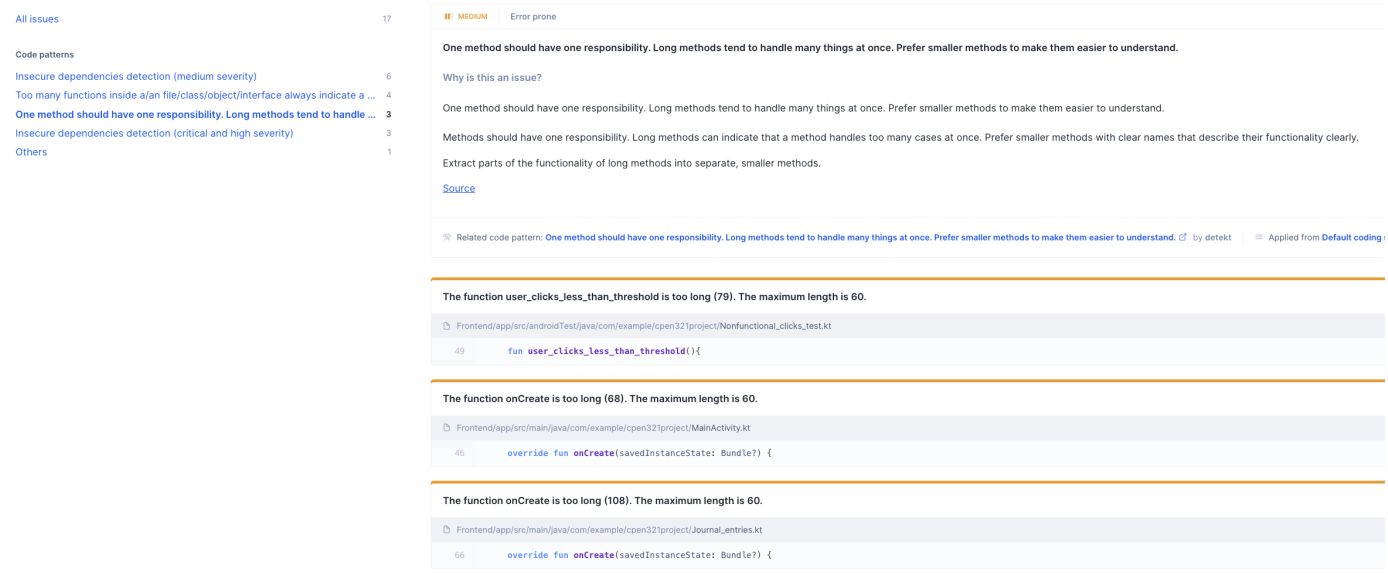
Code Pattern: Insecure dependencies detection (medium severity)



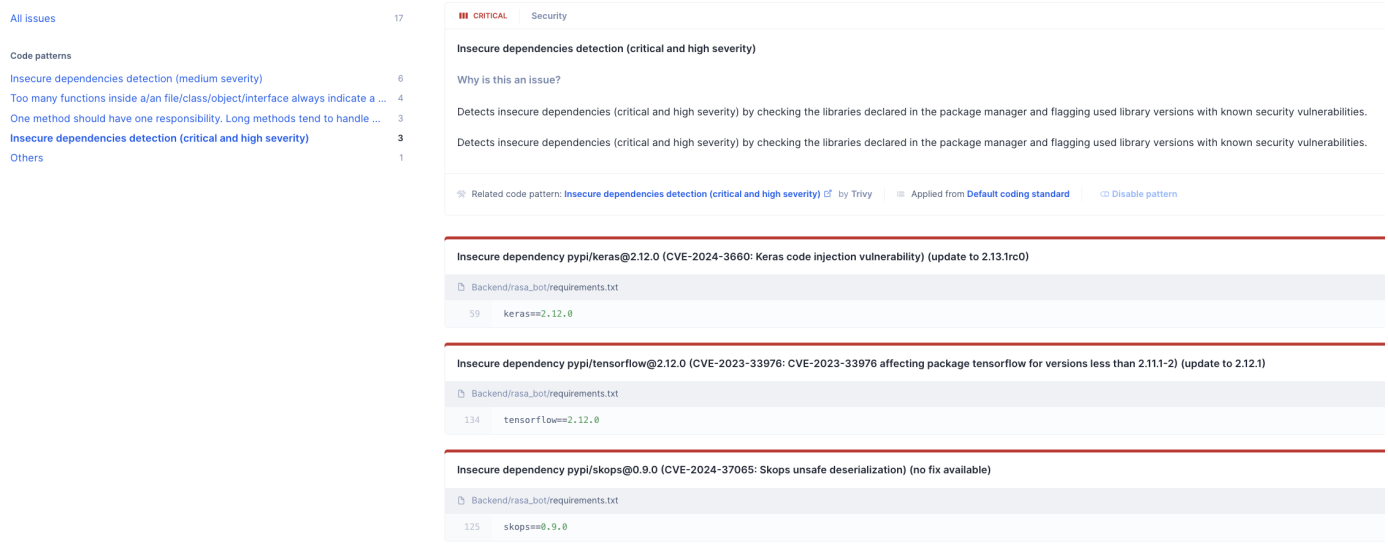
Code Pattern: Too many functions inside a class (medium severity)



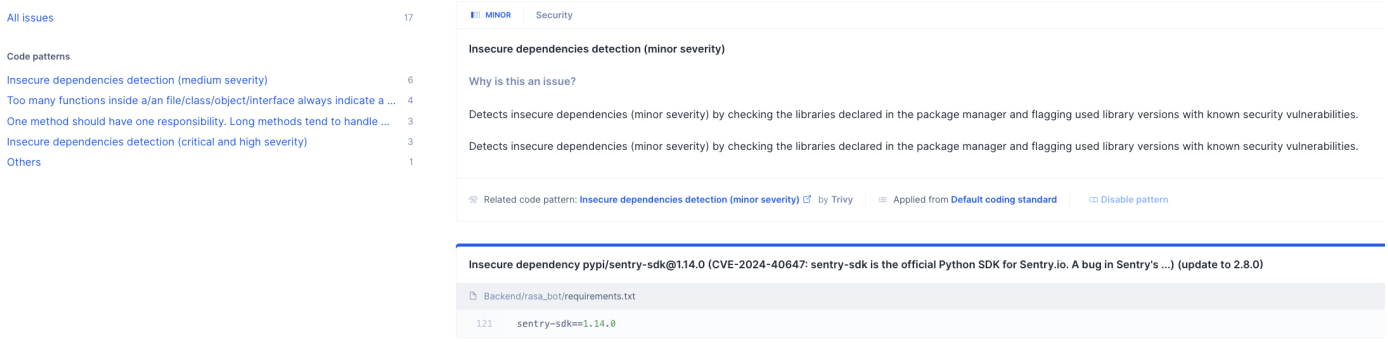
Code Pattern: One method should have one responsibility (medium severity)



Code Pattern: Insecure Dependencies Detection (high severity)



Code Pattern: Insecure dependencies detection (minor severity)



5.4. Justifications for Unfixed Issues

- **Code Pattern: Insecure dependencies detection (medium severity)**
  1. **Issue**
     - **Location in Git:** `Backend/rasa_bot/requirements.txt#L91`
     - **Justification:** The pydantic version 1.10.9 is flagged for a regular expression denial of service vulnerability. This issue can be mitigated by upgrading to a more recent, stable version once it is available and tested for compatibility with RASA's required functionality.
  2. **Issue**
     - **Location in Git:** `Backend/rasa_bot/requirements.txt#L59`

    - **Justification:** The dnspython version 2.3.0 has a denial of service vulnerability, upgrading to version 2.6.1 will be considered during the next dependency for a more stable and secure version.
  3. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L87`
    - **Justification:** The pymongo version 4.3.3 has an out-of-bounds read in the BSON module. Updating to 4.6.3 will be performed during the next dependency once it has been tested for backward compatibility with RASA's database handling functions.
  4. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L31`
    - **Justification:** The keras dependency at version 2.12.0 has an arbitrary file write vulnerability via its get_file function. Unfortunately, no fix is currently available, and a fix will be implemented as soon as it is feasible.
  5. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L119`
    - **Justification:** The aiohttp version 3.9.5 contains a vulnerability in handling compressed files as symlinks, updating to a more stable version will be considered during the next developnment cycle.
  6. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L5`
    - **Justification:** The scikit-learn dependency at version 1.1.3 is flagged for a possible sensitive data leak. The dependency will be updated to version 1.5.0 once it has been tested for compatibility with RASA's ML operations during the next developnment cycle.
- **Code Pattern: Too many functions inside a class (medium severity)**
  1. **Issue**
    - **Location in Git:** `Frontend/app/src/main/java/com/example/cpen321project/AnalyticsActivity.kt#L30`
    - **Justification:** The AnalyticsActivity class contains 11 functions, reaching the defined threshold. Refactoring is planned to distribute responsibilities across multiple classes to adhere to the single responsibility principle.
  2. **Issue**
    - **Location in Git:** `Frontend/app/src/main/java/com/example/cpen321project/MainActivity.kt#L36`
    - **Justification:** The MainActivity class has 12 functions, exceeding the recommended limit. A redesign is scheduled to modularize functionalities into separate components.
  3. **Issue**
    - **Location in Git:** `Frontend/app/src/main/java/com/example/cpen321project/Journal_entries.kt#L47`
    - **Justification:** The Journal_entries class comprises 22 functions, significantly surpassing the threshold. A comprehensive refactor is planned to improve maintainability by delegating responsibilities to smaller classes.
  4. **Issue**
    - **Location in Git:** `Frontend/app/src/main/java/com/example/cpen321project/ProfileManagement.kt#L58
    - **Justification:** The ProfileManagement class contains 15 functions. Future development cycles will address this by implementing a more modular architecture.
- **Code Pattern: One method should have one responsibility (medium severity)**
  1. **Issue**
    - **Location in Git:** `Frontend/app/src/main/java/com/example/cpen321project/Journal_entries.kt#L72`
    - **Justification:** The `onCreate` function in this file is lengthy due to multiple necessary initializations, including UI setup and event listeners. Breaking it into smaller methods could add unnecessary complexity and reduce readability. Additionally, restructuring would require extensive refactoring and testing, which is currently not feasible within the project's timeline.
  2. **Issue**
    - **Location in Git:** `Frontend/app/src/androidTest/java/com/example/cpen321project/Nonfunctional_clicks_test.kt#L49`
    - **Justification:** The function `user_clicks_less_than_threshold` is part of a testing script where all actions are meant to be executed sequentially in a controlled environment. Splitting it into multiple methods would not improve maintainability and may negatively impact the readability of test execution flow.
  3. **Issue**
    - **Location in Git:** `Frontend/app/src/main/java/com/example/cpen321project/MainActivity.kt#L46`
    - **Justification:** Similar to the previous `onCreate` function, this method handles multiple setup operations necessary for the activity lifecycle. While some refactoring is possible in the future, the current implementation follows Android development best practices for initializing UI components and handling necessary dependencies.
- **Code Pattern: Insecure Dependencies Detection (high severity)**
  1. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L59`
    - **Justification:** The keras version 2.12.0 is flagged for a code injection vulnerability. This is a critical issue that needs to be addressed by updating to 2.13.1rc0. We plan to prioritize this update as part of the next major dependency update, after verifying compatibility with other parts of the application.
  2. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L134`
    - **Justification:** The tensorflow version 2.12.0 is affected by a known security vulnerability (CVE-2023-33976), upgrading to 2.13.0 will be carried out once the new version is tested for compatibility.
  3. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L125`
    - **Justification:** The skops version 0.9.0 has a vulnerability due to unsafe deserialization, which could lead to remote code execution. Unfortunately, no fix is available at the moment. And a fix will be implemented as soon as it is feasible.
- **Code Pattern: Insecure dependencies detection (minor severity)**
  1. **Issue**
    - **Location in Git:** `Backend/rasa_bot/requirements.txt#L121`
    - **Justification:** The sentry-sdk version 1.14.0 is flagged for a known vulnerability (CVE-2024-40647). We plan to upgrade to version 2.8.0, which addresses this vulnerability, after verifying compatibility with other parts of the application.