Testing and Code Review

1. Change History

Change Date	Modified Sections	Rationale
29.11.2025	2.1.1	Updated backend test chart with correct line numbers after refactor, listed specific mocked components
27.11.2025	4.1	More detailed setup instructions
27.11.2025	3.2, 4.*	Moved the frontend nonfunctional from frontend to nonfunctional

---

2. Back-end Test Specification: APIs

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

2.1.1. Tests

Notes API

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
POST /api/notes	backend/src/__tests__/unmocked/notes.normal.test.ts#L46	backend/src/__tests__/mocked/notes.mocked.test.ts#L66	Notes, OpenAI
PUT /api/notes/:id	#L288	#L219	Notes, OpenAI
DELETE /api/notes/:id	#L406	#L315	Notes
GET /api/notes/:id	#L470	#L270	Notes
GET /api/notes	#L537	#L360	Notes, Workspaces, OpenAI
GET /api/notes/:id/workspaces	#L910	#L643	Notes
POST /api/notes/:id/share	#L671	#L688	Notes, Workspaces
POST /api/notes/:id/copy	#L793	#L787	Notes

Workspaces API

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
POST /api/workspace	backend/src/__tests__/unmocked/workspace.normal.test.ts#L44	backend/src/__tests__/mocked/workspace.mocked.test.ts#L51	Workspaces
GET /api/workspace/personal	#L166	#L94	Workspaces
GET /api/workspace/user	#L281	#L147	Workspaces
GET /api/workspace/:id	#L343	#L181	Workspaces

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/workspace/:id/\ members	#L401	#L215	Workspaces, Firebase Cloud Messaging
GET /api/workspace/:id/tags	#L445	#L265	Workspaces
GET /api/workspace/:id/\ membership/:userId	#L527	#L299	Workspaces
POST /api/workspace/:id/\ members	#L610	#L333	Workspaces, Firebase Cloud Messaging
POST /api/workspace/:id/leave	#L838	#L570	Workspaces
PUT /api/workspace/:id	#L950	#L588	Workspaces
PUT /api/workspace/:id/\ picture	#L1042	#L624	Workspaces,
DELETE /api/workspace/:id/\ members/:userId	#L1147	#L660	Workspaces
DELETE /api/workspace/:id	#L1314	#L694	Workspaces, Firebase Cloud Messaging
GET /api/workspace/:id/poll	#L1420	#L728	Workspaces

Authentication API

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
POST /api/auth/signup	backend/src/__tests__/\ unmocked/auth.normal.\ test.ts#L57	backend/src/__tests__/\ mocked/auth.mocked.\ test.ts#L57	Users, Workspaces, Google Auth
POST /api/auth/signin	#L105	#L453	Users, Google Auth
POST /api/auth/dev-login	#L122	#L655	Users, Database

User API

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/user/profile	backend/src/__tests__/\ unmocked/user.normal.\ test.ts#L44	backend/src/__tests__/\ mocked/user.mocked.\ test.ts	—
PUT /api/user/profile	#L76	#L52	Users, Workspaces
DELETE /api/user/profile	#L279	#L125	Workspaces
POST /api/user/fcm-token	#L494	#L176	Users
GET /api/user/:id	#L630	#L247	Users

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/user/email/\ :email	#L700	#L349	Users

Message API

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/messages/\ workspace/:workspaceId	backend/src/__tests__/\ unmocked/message.normal.\ test.ts#L40	backend/src/__tests__/\ mocked/message.mocked.\ test.ts#L48	Messages
POST /api/messages/\ workspace/:workspaceId	#L181	#L72	Messages, Workspaces
DELETE /api/messages/\ :messageId	#L268	#L115	Messages

2.1.2. Commit Hash Where Tests Run `c5f46d61177b82ff74c9c30dfd32a5e24de5d683`

2.1.3. How to Run the Tests

1. `cd backend`
2. `npm install`
3. Add an OpenAI api key to your .env file. You can find our api key in the M5 Report.
4. `npm test`

2.2. GitHub Actions Configuration Location

`~/.github/workflows/backend-tests.yml`

2.3. Jest Coverage Report Screenshots (Without Mocking)

2.4. Jest Coverage Report Screenshots (With Mocking)

2.5. Combined Jest Coverage Reports (With & Without Mocking)

---

3. Back-end Non-functional Requirements

3.1. Test Locations

Non-Functional Requirement	Location in Git
Backend – Search Speed	ThingSpace.ts/backend/src/__tests__/notes.latency.test.ts |
Frontend – Two-Click Navigation	'frontend/app/src/androidTest/java/com/cpen321/usermanagement/TestReachWithTwoClic

Backend – Search Speed (`notes.latency.test.ts`)

- **How to run:** `cd backend && npm test -- __tests__/non-func-tests`
- **What it checks:** Seeds 400 notes, issues three representative search queries, and reports the mean latency. Latest runs average ~1.1s/query, comfortably under the 5s budget.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	81.35	59.36	88.59	81.46	
src	100	100	100	100	
routes.ts	100	100	100	100	
src/authentication	68.07	48.57	63.63	67.68	
auth.controller.ts	50	23.07	66.66	50	23-38,47,53,59,74-106,124-126
auth.middleware.ts	100	100	100	100	
auth.routes.ts	100	100	100	100	
auth.service.ts	43.63	11.11	50	43.63	17-40,47,53,59-96
auth.types.ts	100	100	100	100	
src/media	84.14	50	100	83.95	
media.controller.ts	88.23	60	100	88.23	24,48
media.routes.ts	100	100	100	100	
media.service.ts	80.35	47.82	100	80	22,34,45,72,80-83,96,104-105,128
src/messages	86.81	76.92	100	86.81	
message.controller.ts	83.78	76.92	100	83.78	14-15,55-56,65-66,103-104,113-114,138-139
message.model.ts	100	100	100	100	
message.routes.ts	100	100	100	100	
message.types.ts	100	100	100	100	
src/middleware	95.65	100	100	95	
errorHandler.middleware.ts	100	100	100	100	
validation.middleware.ts	91.66	100	100	90.9	23
src/notes	81.18	59.77	86.2	82.75	
note.model.ts	100	100	100	100	
notes.controller.ts	82.94	52.38	100	82.94	10-11,26-27,35-36,58-59,79-80,95-96,104-105,138,146-147,172-173,180,203-204
notes.routes.ts	100	100	100	100	
notes.service.ts	74.1	65.11	77.77	77.45	152,167,224,254-290
notes.types.ts	100	100	100	100	
src/notifications	65	33.33	50	65	
notification.service.ts	65	33.33	50	65	7,46-63
src/users	68.75	20	93.75	68.75	
user.controller.ts	65.43	12	100	65.43	15-16,33,39,49-57,65,91-99,107,117,126-134,161-169,178,196-204
user.model.ts	63.93	60	90	63.93	85-90,113-114,122-123,146-162,188-189,205-206
user.routes.ts	100	100	100	100	
user.types.ts	100	100	100	100	
src/utils	61.76	66.66	63.15	61.29	
asyncHandler.util.ts	100	100	100	100	
constants.ts	100	100	100	100	
database.ts	0	0	0	0	1-44
logger.util.ts	100	50	100	100	7-11
sanitizeInput.util.ts	100	100	100	100	
storage.ts	100	100	100	100	
src/workspaces	90.63	70	100	90.37	
workspace.controller.ts	82.95	54.44	100	82.95	...249,277,285-286,321,329-330,370,378-379,409,417-418,449,457-458,489,496-497,517
workspace.model.ts	100	100	100	100	
workspace.routes.ts	100	100	100	100	
workspace.service.ts	99.46	97.91	100	99.42	50
workspace.types.ts	100	100	100	100	

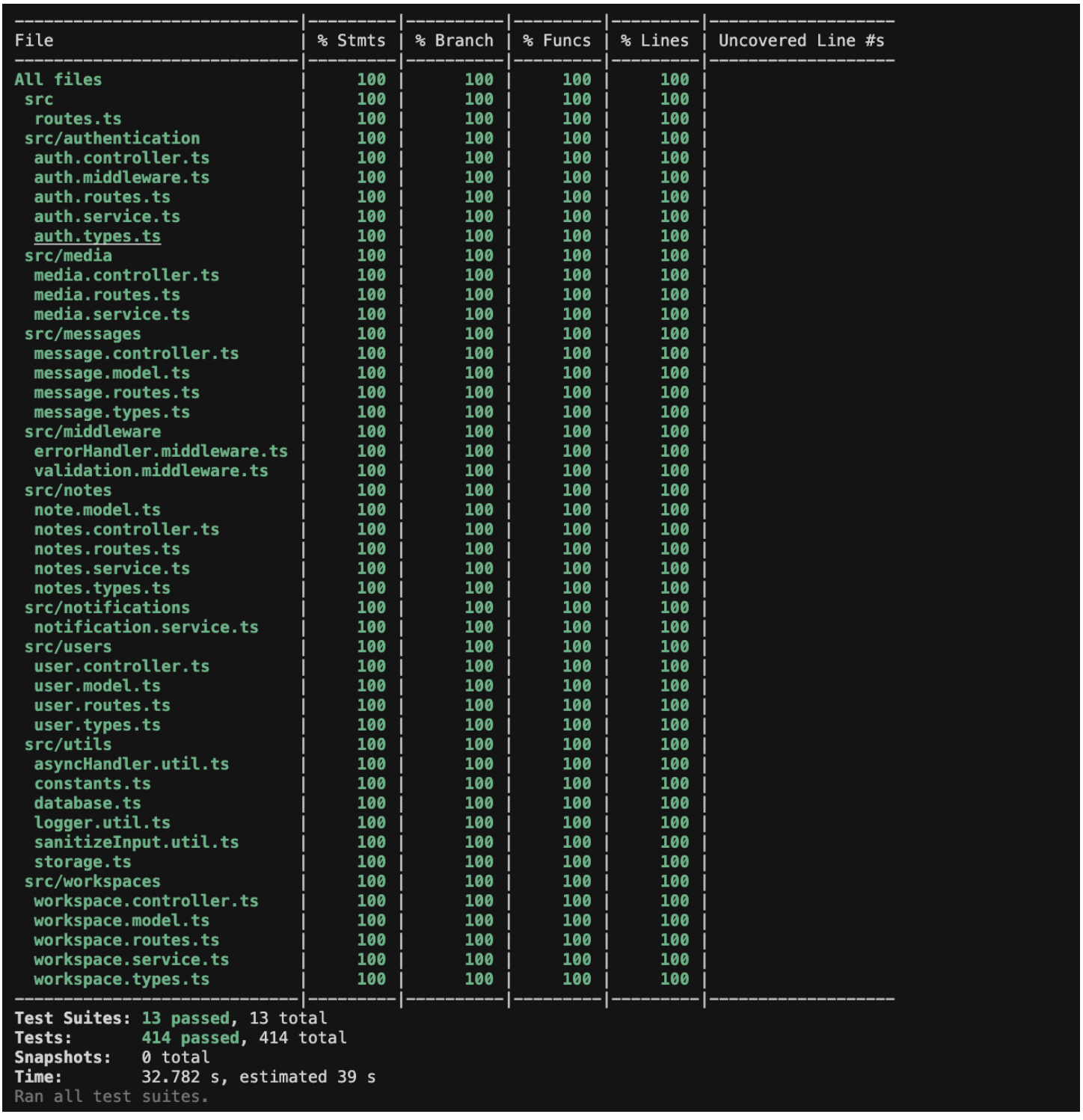
```
Test Suites: 6 passed, 6 total
Tests:       211 passed, 211 total
Snapshots:   0 total
Time:        25.728 s, estimated 41 s
Ran all test suites matching /unmocked/i.
```

Figure 1: image info

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	74.15	60.51	72.48	74.51	
src	100	100	100	100	
routes.ts	100	100	100	100	
src/authentication	77.71	57.14	90.9	77.43	
auth.controller.ts	100	92.3	100	100	112
auth.middleware.ts	38.88	0	50	36.53	18-22,27-31,41-45,53-57,61-65,79-120
auth.routes.ts	100	100	100	100	
auth.service.ts	92.72	88.88	100	92.72	108-109,126-127
auth.types.ts	100	100	100	100	
src/media	97.56	96.42	100	97.53	
media.controller.ts	94.11	80	100	94.11	16
media.routes.ts	100	100	100	100	
media.service.ts	98.21	100	100	98.18	102
src/messages	76.92	23.07	100	76.92	
message.controller.ts	71.62	23.07	100	71.62	21-22,29-30,34-35,43,72-73,80-81,85-86,101,119-120,126-127,131-132,136
message.model.ts	100	100	100	100	
message.routes.ts	100	100	100	100	
message.types.ts	100	100	100	100	
src/middleware	86.95	50	60	85	
errorHandler.middleware.ts	90.9	100	50	88.88	6
validation.middleware.ts	83.33	0	66.66	81.81	13-16
src/notes	75.64	62.06	75.86	75.47	
note.model.ts	100	100	100	100	
notes.controller.ts	77.51	69.04	100	77.51	...113,118,126-127,134-135,154-155,160,168-169,176-177,189,212-213,216-217,225-228
notes.routes.ts	100	100	100	100	
notes.service.ts	66.96	53.48	61.11	65.68	13,60,73-113,128,131,137,143,155,171-210,230,239,246
notes.types.ts	100	100	100	100	
src/notifications	100	100	100	100	
notification.service.ts	100	100	100	100	
src/users	92.5	86.66	100	92.5	
user.controller.ts	87.65	92	100	87.65	19,44,120-121,150-156,185-191
user.model.ts	96.72	60	100	96.72	111,132
user.routes.ts	100	100	100	100	
user.types.ts	100	100	100	100	
src/utils	97.05	88.88	94.73	98.38	
asyncHandler.util.ts	80	100	66.66	100	
constants.ts	100	100	100	100	
database.ts	100	100	100	100	
logger.util.ts	100	100	100	100	
sanitizeInput.util.ts	100	100	100	100	
storage.ts	94.73	66.66	100	94.73	39
src/workspaces	56	48.57	43.39	57.11	
workspace.controller.ts	71.59	65.55	100	71.59	...391,400-401,404-405,431,440-441,444-445,467,476-477,480-481,484-485,503,512-513
workspace.model.ts	100	100	100	100	
workspace.routes.ts	100	100	100	100	
workspace.service.ts	24.19	14.58	21.05	24.85	19,53-118,130,138-189,196,202,208,214,220,226,255,271-477
workspace.types.ts	100	100	100	100	

```
Test Suites: 6 passed, 6 total
Tests:       202 passed, 202 total
Snapshots:   0 total
Time:        12.864 s, estimated 39 s
Ran all test suites matching /__tests__\/mocked\//i.
```

Figure 2: image info



File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
src	100	100	100	100	
routes.ts	100	100	100	100	
src/authentication	100	100	100	100	
auth.controller.ts	100	100	100	100	
auth.middleware.ts	100	100	100	100	
auth.routes.ts	100	100	100	100	
auth.service.ts	100	100	100	100	
auth.types.ts	100	100	100	100	
src/media	100	100	100	100	
media.controller.ts	100	100	100	100	
media.routes.ts	100	100	100	100	
media.service.ts	100	100	100	100	
src/messages	100	100	100	100	
message.controller.ts	100	100	100	100	
message.model.ts	100	100	100	100	
message.routes.ts	100	100	100	100	
message.types.ts	100	100	100	100	
src/middleware	100	100	100	100	
errorHandler.middleware.ts	100	100	100	100	
validation.middleware.ts	100	100	100	100	
src/notes	100	100	100	100	
note.model.ts	100	100	100	100	
notes.controller.ts	100	100	100	100	
notes.routes.ts	100	100	100	100	
notes.service.ts	100	100	100	100	
notes.types.ts	100	100	100	100	
src/notifications	100	100	100	100	
notification.service.ts	100	100	100	100	
src/users	100	100	100	100	
user.controller.ts	100	100	100	100	
user.model.ts	100	100	100	100	
user.routes.ts	100	100	100	100	
user.types.ts	100	100	100	100	
src/utils	100	100	100	100	
asyncHandler.util.ts	100	100	100	100	
constants.ts	100	100	100	100	
database.ts	100	100	100	100	
logger.util.ts	100	100	100	100	
sanitizeInput.util.ts	100	100	100	100	
storage.ts	100	100	100	100	
src/workspaces	100	100	100	100	
workspace.controller.ts	100	100	100	100	
workspace.model.ts	100	100	100	100	
workspace.routes.ts	100	100	100	100	
workspace.service.ts	100	100	100	100	
workspace.types.ts	100	100	100	100	

```
Test Suites: 13 passed, 13 total
Tests:       414 passed, 414 total
Snapshots:   0 total
Time:        32.782 s, estimated 39 s
Ran all test suites.
```

Figure 3: image info

Frontend – Two-Click Navigation (`TestReachWithTwoClicks.kt`)

- **How to run:** `cd frontend && ./gradlew connectedAndroidTest -Pandroid.testInstrumentationRunnerArguments.class=com.cpen321.usermanagement.TestReachWithTwoClicks`
- **What it checks:** Starting from the main workspace screen, the test traverses to note, template, and chat views—both within the current workspace and across other workspaces — counting taps to confirm every note-bearing screen is reachable in 2 clicks.

3.2. Test Verification and Logs

Performance: Search Speed

- Creates 400 notes, runs 3 queries, average ~1.1s (<5s target)
- Logs:

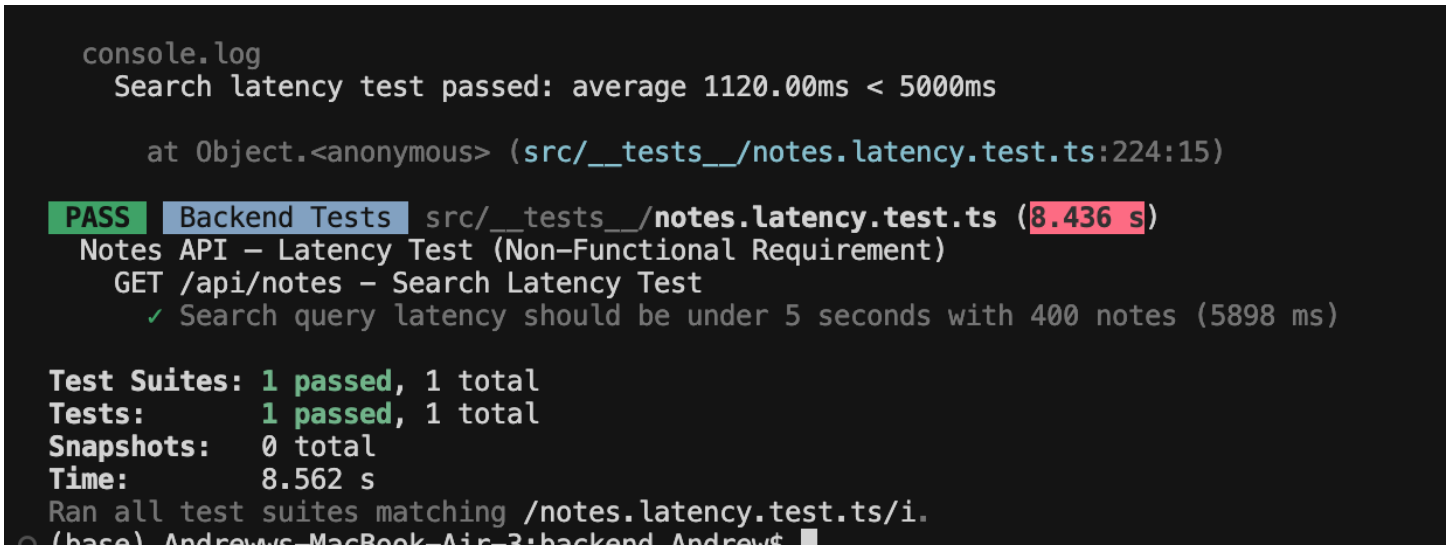


Figure 4: image info

Accessibility: Reach Note Screens in 2 Clicks (`TestReachWith2Clicks.kt`) This test verifies that from a workspace main screen, users can reach:

- templates screen
- chat screen
- content/templates/chat of another workspace

All screens must be reachable in **2 clicks or fewer**.

The test counts the number of clicks for each navigation path.

- Logs:

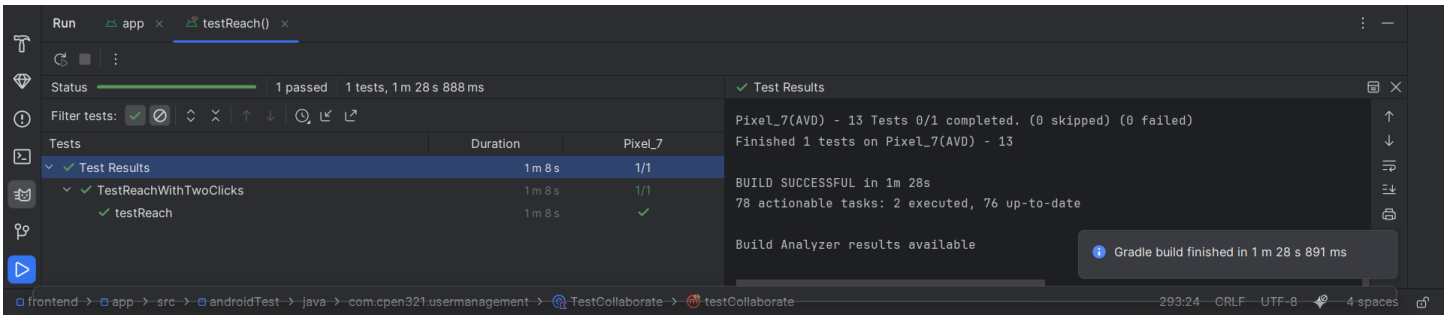


Figure 5: image info

4. Front-end Test Specification

4.1. Location in Git

`./frontend/app/src/androidTest/java/com.cpen321.usermanagement`

To run the frontend tests, one needs to have a working backend (deployed in the cloud or locally - .env setup in the backend test instructions).

On top of that it is necessary to update the `frontend/local.properties` file: sdk.dir= …sdk folder location on your computer API_BASE_URL= "…url_to_backend/api" IMAGE_BASE_URL="…path to a port on the emulator localhost" GOOGLE_CLIENT_ID="…google client id"

Google client Id can be obtained by setting up a project in the Aoogle OAuth console and creating a web application. The client ID of the web application has to be posted into local.properties. On top of that, one needs to register the frontend app inside the Google OAuth Console. To do that one needs to obtain the development SHA-1 key of the application, generated by the gradle signingReport task. With the key obtained, one can enter the package name: com.cpen321.usermanagement, and the key into the Android client creation menu. While it is vital to create the android client, its Google client id should not be pasted into local.properties, only the backend Google client id.

To set up the content inside the app one can either sign in on their Android Emulator with Thing4G and Friedrich van Aukstin dummy users, which have the right notes and workspaces pre-created, or create two accounts and fill in their bios. In the latter case, one will need to pre-create notes and workspaces as per requirements of a specific test (the per test requirements are listed in comments at the beginning of each test file). One would also need to modify the test file's constants to reflect account names and gmails of the accounts used.

4.2. Tests Included

Test of the "Collaborate" Feature (`TestCollaborate.kt`)

Prerequisite:

1. Need 2 user accounts:
   - Workspace Manager
   - Workspace Member
2. Workspace Manager must have a workspace named **"Test"**
3. There must be **no existing workspaces** named *"Study"* or *"Study v2"*

Scenario Steps and Test Case Steps

Scenario Steps	Test Case Steps
Create Workspace	
1. The user opens the "Create Workspace" screen.	Open the "Create Workspace" screen.
2. The app shows input fields and a disabled button.	Verify "Pick a workspace name" and disabled "Create Workspace" button.
3a. User enters workspace title already taken.	Pre-create workspace "Test". Enter "Test" and click Create.
3a1. App shows error.	Check for dialog: **"Failed to create workspace."**
3. User enters valid title.	Enter "Studies". Verify "Create" button enabled.
4. User clicks "Create".	Click "Create". Verify workspace setup screen shown and "Studies" appears in list.
Update Workspace	
10. Manager navigates to "Edit Workspace".	<After workspace creation, one is already at the edit workspace screen>
11. Edit title and bio.	Change title to "Studies v2" and bio "Study group"; click Save; expect "Profile updated successfully."
Invite to Workspace	
12. Member selects "Invite User".	Open Studies v2 → Manage Workspace → Invite icon.
13. App shows input and button.	Verify email field + "Invite to Workspace" button visible.

Scenario Steps	Test Case Steps
14a. Enter invalid email. 14a1. Error message shown.	Enter “invalidemail”, click Invite. Check: **“Could not retrieve profile matching the given email!”**
15. Enter valid email.	Input teammate email → click Invite → expect **“The user got added to the workspace.”**
15b. Invite already-member user. 15b1. Error message.	Enter existing member email and click Invite. **“The user is already a member!”**
Send Chat Message	
16. User opens workspace chat.	Open chat icon; verify chat screen shown.
17a. Empty message.	Send blank message; verify no change.
17. Valid message.	Send “Hello team!”; verify appears with picture & timestamp.
Update Workspace as Non-Manager	
18a. Non-manager tries editing workspace.	Log out as the manager. Open edit screen as non-manager → fields should be greyed out.
Leave Workspace (Non-Manager)	
19. Non-manager clicks Leave.	Open Studies v2 → Leave Workspace.
19. App removes user.	Studies v2 no longer appears in workspace list.
Ban Users	
20. Manager opens Members screen.	Log in as manager → Workspaces Screen → <select “Studies v2”> → Manage Workspace → Members icon.
21. Manager bans user.	Click trash next to user.
22. User banned permanently.	User removed, cannot be re-invited → **“That user is banned”** message should show upon an invite attempt
Delete Workspace	
23. Manager deletes workspace.	Click Delete Workspace (trash icon).
24. Workspace deleted.	Studies v2 disappears and an appropriate success meassage is shown.

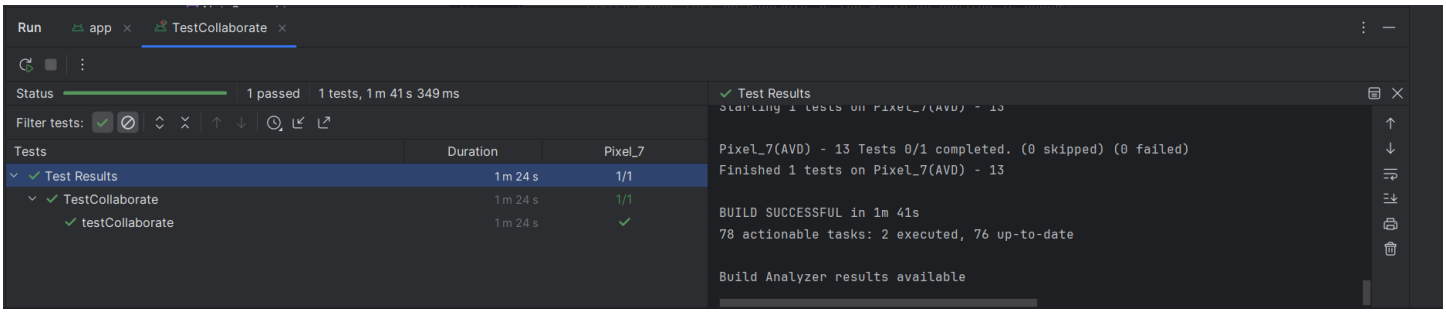


Figure 6: image info

Test of the “Manage Notes” Feature (`TestNotes.kt`)

Prerequisite:
Two pre-existing workspaces.

Scenario Steps	Test Case Steps
Create Note	
1. Open “Create Note” screen.	Tap pencil icon in workspace screen.
2. App shows metadata fields and create button.	Verify “Create Note” button present.
3a. Create note with no title. 3a1. Error.	Click “Create Note” button. Don’t input title. Click “Create”. **“Please enter a title”**

Scenario Steps	Test Case Steps
3. Add title.	Enter “Test Note”.
4a. Create note with no fields.	Click “Create” with no fields added.
4a1. Error.	**“Please add at least one field”**
4b. Create note with empty field label.	Add field (select TEXT type). Add tag “important”. Clear field label; click “Create”.
4b1. Error.	**“All fields must have a label”**
4. Add field label and content.	Input field label “Notes”. Input content in field.
5. Click Create.	Click “Create”.
6. Note created.	Verify note “Test Note” appears in workspace.
Update Note	
7. Open note to edit.	Navigate to note → click pencil icon.
8. App shows editable fields.	Verify fields editable. Verify tags can be added/removed (click existing tag).
9. Modify content & tags.	Change field content. Add new tag. Remove existing tag.
10. Click Save.	Click “Save”.
11. Note updated.	Verify changes reflected in Note Details screen.
Share Note	
12. Select “Share Note”.	Open note edit screen → click Share icon.
13. Workspace selection dialog.	Verify “Share Note” + workspace list visible.
14. Select workspace & confirm.	Select workspace → click “Share”.
15. Note shared.	**“Note shared to workspace successfully”**. Verify note in target workspace, removed from original.
Copy Note	
16. Select “Copy Note”.	Open note edit screen → click Copy icon.
17. Workspace selection.	Verify “Copy Note” + workspace list visible.
18. Select workspace.	Select workspace → click “Copy”.
19. Note copied.	Verify note appears in both target and source workspaces.
Delete Note	
20. Select “Delete Note”.	Navigate to note → click trash icon.
21. Confirmation dialog.	Verify “Delete Note” + **“Are you sure... cannot be undone”**
22. Confirm.	Click “Delete”.
23. Note deleted.	**“Note successfully deleted”**. Verify note removed from workspace.

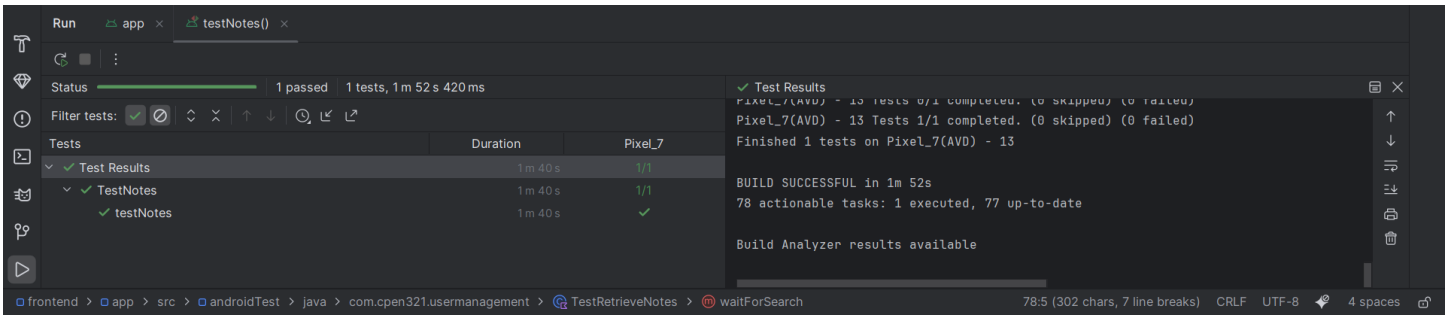


Figure 7: image info

Test of the "Retrieve Notes" Feature (`TestRetrieveNotes.kt`)

Prerequisite:
Existing workspace with notes.

Scenario Steps	Test Case Steps
Search Notes	
1. Open workspace.	Navigate to workspace.
2. Search bar visible.	Verify presence.
3a. Empty query.	Search blank.
3a1. All notes shown.	Verify list.
3. Enter query.	Input string → Search.
4. Matching notes shown.	Relevant notes at top.
Filter Notes by Tags	
5. Click filter icon.	
6. Tag selection screen.	Verify "All" + checkboxes.
7. Select "All".	All tags selected.
9. Deselect "All".	All tags cleared.
11. Select specific tags.	
12. Go back.	
13. Filter applied.	Only matching notes. Ordered according to the search query

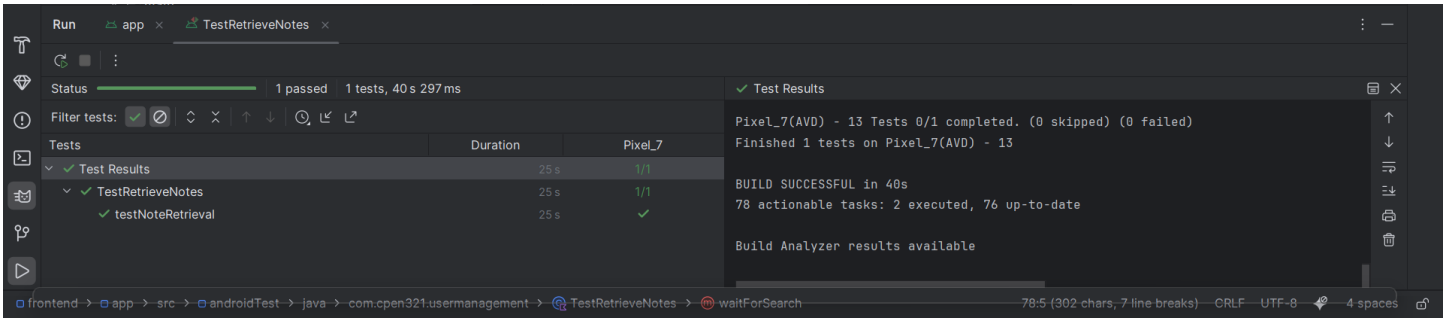


Figure 8: image info

---

5. Automated Code Review Results

5.1. Commit Hash Where Codacy Ran

`96e4c5520f47503662f56029212714c229f3617f`

5.2. Unfixed Issues per Codacy Category

5.3. Unfixed Issues per Codacy Code Pattern

5.4. Justifications for Unfixed Issues

No unfixed codacy issues

Issues breakdown

All clear!

When issues come up, you'll be able to see here the issues breakdown per category.

Figure 9: image info

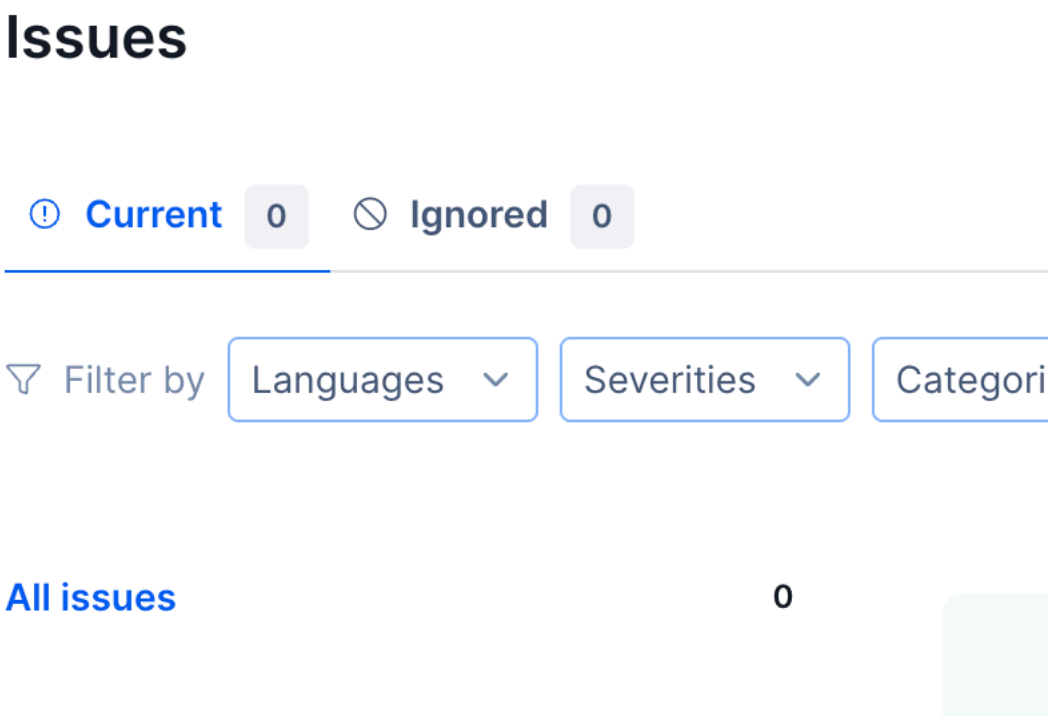


Figure 10: image info