

Testing and Code Review

1. Change History

Change Date	Modified Sections	Rationale
<i>Nothing to show</i>		

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#L40	backend/src/__tests__/\ mocked/notes.mocked.\ test.ts#L51	noteService.createNote, OpenAI embeddings client
PUT /api/notes/:id	#L282	#L202	noteService.updateNoteById
DELETE /api/notes/:id	#L398	#L298	noteService.deleteNote
GET /api/notes/:id	#L462	#L253	noteService.getNoteById
GET /api/notes	#L528	#L343	noteService.searchNotes, OpenAI embeddings
GET /api/notes/:id/\ workspaces	#L893	#L620	workspaceModel.findById
POST /api/notes/:id/\ share	#L662	#L665	noteService.\ shareNoteToWorkspace, workspaceModel
POST /api/notes/:id/\ copy	#L784	#L764	noteService.copyNote, workspaceModel

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#L45	backend/src/__tests__/\ mocked/workspace.mocked.\ test.ts#L237	workspaceService.\ createWorkspace
GET /api/workspace/personal	#L167	#L277	workspaceService.\ getPersonalWorkspaces
GET /api/workspace/user	#L290	#L330	workspaceService.\ getUserWorkspaces
GET /api/workspace/:id	#L352	#L364	workspaceService.\ getWorkspaceById

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/workspace/:id/members	#L410	#L398	workspaceService.getMembers, notificationService
GET /api/workspace/:id/tags	#L454	#L448	workspaceService.\getWorkspaceTags
GET /api/workspace/:id/\membership/:userId	#L536	#L482	workspaceService.getMembership
POST /api/workspace/:id/members	#L619	#L516	workspaceService.addMembers, notificationService
POST /api/workspace/:id/leave	#L847	#L694	workspaceService.\leaveWorkspace
PUT /api/workspace/:id	#L959	#L712	workspaceService.\updateWorkspace
PUT /api/workspace/:id/picture	#L1051	#L748	workspaceService.\updateWorkspacePicture, storage
DELETE /api/workspace/:id/\members/:userId	#L1156	#L784	workspaceService.removeMember
DELETE /api/workspace/:id	#L1323	#L818	workspaceService.\deleteWorkspace, notificationService
GET /api/workspace/:id/poll	#L1429	#L852	workspaceService.\getWorkspaceWithPolling

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#L123	backend/src/__tests__/\mocked/auth.mocked.\test.ts#L74	authService.signup, Google token verifier, workspaceService
POST /api/auth/signin	#L151	#L290	authService.signIn
POST /api/auth/dev-login	#L68	#L394	authService.devLogin

User API

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/users/profile	backend/src/__tests__/\unmocked/user.normal.\test.ts#L39	—	—
PUT /api/users/profile	#L57	backend/src/__tests__/\mocked/user.mocked.\test.ts#L48	userModel.updateOne, workspaceModel
DELETE /api/users/profile	#L166	#L104	workspaceModel, notification service
POST /api/users/fcm-token	#L230	#L155	userModel.updateOne

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/users/:id	#L273	#L209	userModel.findById
GET /api/users/email/:email	#L318	#L260	userModel.findByEmail

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#L61	backend/src/__tests__/\mocked/message.mocked.\test.ts#L67	messageModel.findByWorkspace
POST /api/messages/\workspace/:workspaceId	#L188	#L91	messageModel.create, workspaceModel
DELETE /api/messages/:messageId	#L261	#L134	messageModel.deleteOne

Media API

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
POST /api/media/upload	backend/src/__tests__/\unmocked/media.normal.\test.ts#L56	backend/src/__tests__/\mocked/media.mocked.\test.ts#L55	storage.uploadImage, file system stubs

2.1.2. Commit Hash Where Tests Run c5f46d61177b82ff74c9c30dfd32a5e24de5d683

2.1.3. How to Run the Tests

1. cd backend
2. npm install
3. 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

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	81.78	61.56	88.35	81.9	
asyncHandler.util.ts	100	100	100	100	
auth.controller.ts	43.18	23.07	66.66	43.18	23-38, 47, 53, 59, 74-106, 121-131
auth.middleware.ts	98.14	100	100	98.07	68
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	
constants.ts	100	100	100	100	
database.ts	0	0	0	0	1-44
errorHandler.middleware.ts	100	100	100	100	
logger.util.ts	100	50	100	100	7-11
media.controller.ts	94.11	80	100	94.11	48
media.routes.ts	100	100	100	100	
media.service.ts	89.28	78.26	100	89.09	72, 81, 104-105, 112, 128
message.model.ts	100	100	100	100	
message.routes.ts	86.95	76.92	100	86.95	18, 53-54, 64, 98-99, 109, 130-131
message.types.ts	100	100	100	100	
note.model.ts	100	100	100	100	
notes.controller.ts	82.11	50	100	82.11	... 6-27, 35-36, 60-61, 81-82, 98-99, 107-108, 141, 149-150, 175-176, 183, 206-207
notes.routes.ts	100	100	100	100	
notes.service.ts	73.45	65.9	76.47	76.69	40, 146, 161, 221, 251-287
notes.types.ts	100	100	100	100	
notification.service.ts	65	33.33	50	65	7, 46-63
routes.ts	100	100	100	100	
sanitizeInput.util.ts	100	100	100	100	
storage.ts	94.73	66.66	100	94.73	13
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	75.4	80	90	75.4	113-114, 122-123, 137-138, 146-162
user.routes.ts	100	100	100	100	
user.types.ts	100	100	100	100	
validation.middleware.ts	91.66	100	100	90.9	23
workspace.controller.ts	82.95	54.44	100	82.95	... -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: 218 passed, 218 total					
Snapshots: 0 total					
Time: 8.724 s, estimated 15 s					
Ran all test suites matching /__tests__\/unmocked/i.					

Figure 1: image info


File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s	Add to Chat 
All files	71.61	57.22	69.17	71.99		
asyncHandler.util.ts	75	100	66.66	100		
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		
constants.ts	100	100	100	100		
database.ts	100	100	100	100		
errorHandler.middleware.ts	90.9	100	50	88.88	6	
logger.util.ts	100	100	100	100		
media.controller.ts	88.23	60	100	88.23	16, 24	
media.routes.ts	100	100	100	100		
media.service.ts	80.35	65.21	80	81.81	22, 34, 45, 96, 102, 116-124	
message.model.ts	100	100	100	100		
message.routes.ts	81.15	23.07	100	81.15	24, 31, 35, 41, 51, 70, 77, 81, 96, 114, 120, 124, 128	
message.types.ts	100	100	100	100		
note.model.ts	100	100	100	100		
notes.controller.ts	79.67	75	100	79.67	... 121, 129-130, 137-138, 157-158, 163, 171-172, 179-180, 192, 215-216, 219-220	
notes.routes.ts	100	100	100	100		
notes.service.ts	67.25	56.81	64.7	66.01	58, 70-107, 122, 125, 131, 137, 149, 163-207, 227, 236, 243	
notes.types.ts	100	100	100	100		
notification.service.ts	100	100	100	100		
routes.ts	100	100	100	100		
sanitizeInput.util.ts	88.88	0	100	83.33	7	
storage.ts	94.73	66.66	100	94.73	39	
user.controller.ts	75.3	88	100	75.3	19, 44, 70-86, 120-121, 143, 150-156, 185-191	
user.model.ts	73.77	20	90	73.77	85-90, 102-111, 132, 188-206	
user.routes.ts	100	100	100	100		
user.types.ts	100	100	100	100		
validation.middleware.ts	83.33	0	66.66	81.81	13-16	
workspace.controller.ts	71.59	65.55	100	71.59	... 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	21.5	10.41	15.78	21.96	13-38, 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: 184 passed, 184 total						
Snapshots: 0 total						
Time: 17.137 s, estimated 27 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	
asyncHandler.util.ts	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	
constants.ts	100	100	100	100	
database.ts	100	100	100	100	
errorHandler.middleware.ts	100	100	100	100	
logger.util.ts	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	
message.model.ts	100	100	100	100	
message.routes.ts	100	100	100	100	
message.types.ts	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	
notification.service.ts	100	100	100	100	
routes.ts	100	100	100	100	
sanitizeInput.util.ts	100	100	100	100	
storage.ts	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	
validation.middleware.ts	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: 403 passed, 403 total					
Snapshots: 0 total					
Time: 35.87 s					
Ran all test suites.					

Figure 3: image info

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/\ TestReachWithTwoClicks.kt

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.

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:

```
console.log
  Search latency test passed: average 1120.00ms < 5000ms

    at Object.<anonymous> (src/__tests__/notes.latency.test.ts:224:15)

PASS   Backend Tests  src/__tests__/notes.latency.test.ts (8.436 s)
Notes API – Latency Test (Non-Functional Requirement)
  GET /api/notes – Search Latency Test
    ✓ Search query latency should be under 5 seconds with 400 notes (5898 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        8.562 s
Ran all test suites matching /notes.latency.test.ts/i.
(base) Andrews-MacBook-Air:~$ cd backend; Andrews-MacBook-Air:backend$
```

Figure 4: image info

4. Front-end Test Specification

This includes the non-functional requirement that every note/message containing screen must be reachable from the main screen within two clicks. This non-functional requirement can only be verified via frontend tests, hence it is included here as `TestReachWith2Clicks.kt`.

4.1. Location in Git

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

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. 2. The app shows input fields and a disabled button. 3a. User enters workspace title already taken. 3a1. App shows error. 3. User enters valid title. 4. User clicks “Create”.	Open the “Create Workspace” screen. Verify “Pick a workspace name” and disabled “Create Workspace” button. Pre-create workspace “Test”. Enter “Test” and click Create. Check for dialog: “ Failed to create workspace. ” Enter “Studies”. Verify “Create” button enabled. Click “Create”. Verify workspace setup screen shown and “Studies” appears in list.
Update Workspace 10. Manager navigates to “Edit Workspace”. 11. Edit title and bio.	<After workspace creation, one is already at the edit workspace screen> Change title to “Studies v2” and bio “Study group”; click Save; expect “Profile updated successfully.”
Invite to Workspace 5. Member selects “Invite User”. 6. App shows input and button. 7a. Enter invalid email. 7a1. Error message shown. 7. Enter valid email. 7b. Invite already-member user. 7b1. Error message.	Open Studies v2 → Manage Workspace → Invite icon. Verify email field + “Invite to Workspace” button visible. Enter “invalidemail”, click Invite. Check: “ Could not retrieve profile matching the given email! ” Input teammate email → click Invite → expect “ The user got added to the workspace. ” Enter existing member email and click Invite. “ The user is already a member! ”
Send Chat Message 8. User opens workspace chat. 9a. Empty message. 9. Valid message.	Open chat icon; verify chat screen shown. Send blank message; verify no change. Send “Hello team!”; verify appears with picture & timestamp.
Update Workspace as Non-Manager 10a. 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) 12. Non-manager clicks Leave. 13. App removes user.	Open Studies v2 → Leave Workspace. Studies v2 no longer appears in workspace list.
Ban Users 14. Manager opens Members screen. 15. Manager bans user.	Log in as manager → Workspaces Screen → <select “Studies v2”> → Manage Workspace → Members icon. Click trash next to user.

Scenario Steps	Test Case Steps
16. User banned permanently.	User removed, cannot be re-invited → “That user is banned” message should show upon an invite attempt
Delete Workspace	
17. Manager deletes workspace.	Click Delete Workspace (trash icon).
18. Workspace deleted.	Studies v2 disappears and an appropriate success message is shown.

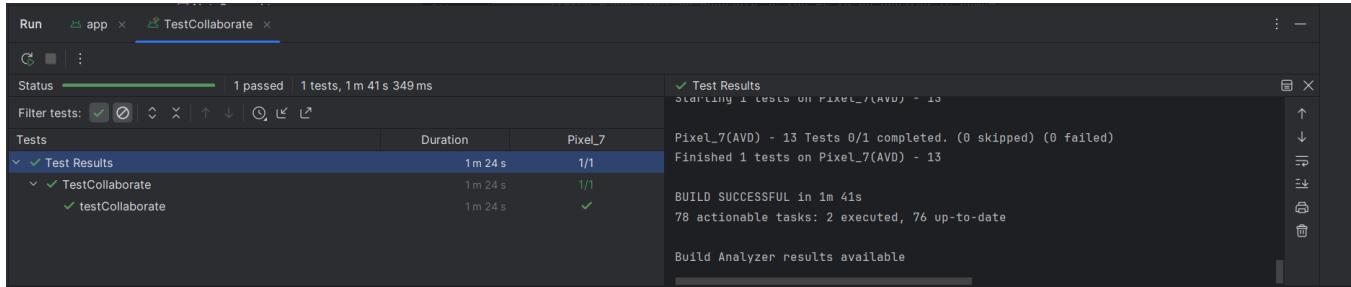


Figure 5: image info

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.
2. App shows metadata fields and create button.	Verify “Note Type”, “Tags”, “Fields”.
3a. Create note with no fields.	Click Create with no fields.
3a1. Error.	“Please add at least one field”
3b. Create note with empty field label.	Add field but leave label blank; click Create.
3b1. Error.	“All fields must have a label”
3. User fills fields.	Add tag(s), add field “Notes”, enter content.
4. Click Create.	
5. Note created.	Verify note appears.
Update Note	
6. Open note to edit.	Click pencil icon on note.
7. App shows editable fields.	Verify tag add/remove.
8. Modify content & tags.	
9. Click Save.	
10. Note updated.	Verify new values.
Share Note	
11. Select “Share Note”.	Click Share icon.
12. Workspace selection dialog.	Verify “Share Note” + workspace list.
13. Select workspace & confirm.	Click Share.
14. Note shared.	“Note shared to workspace successfully” , note moves workspaces.
Copy Note	
15. Select “Copy Note”.	Click Copy icon.
16. Workspace selection.	Verify UI.
17. Select workspace.	Click Copy.
18. Note copied.	Appears in both workspaces.

Scenario Steps	Test Case Steps
Delete Note	
19. Select “Delete Note”.	Tap trash.
20. Confirmation dialog.	“Are you sure... cannot be undone”
21. Confirm.	Click Delete.
22. Note deleted.	Verify removal.

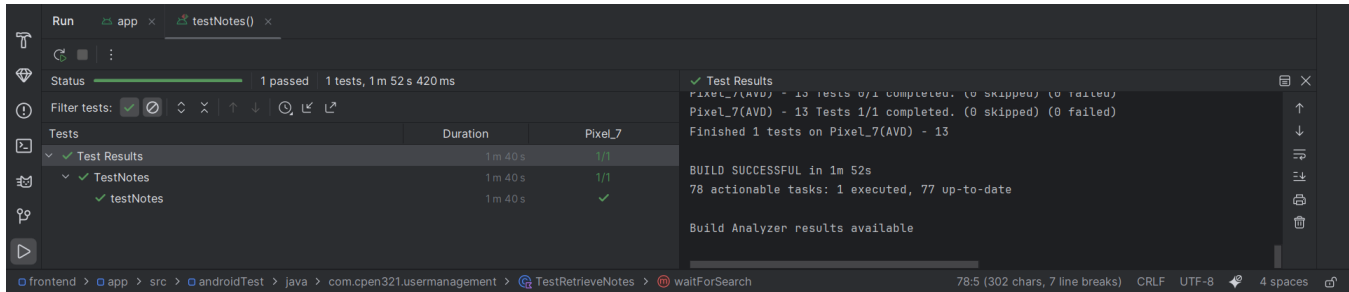


Figure 6: image info

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

Test: 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

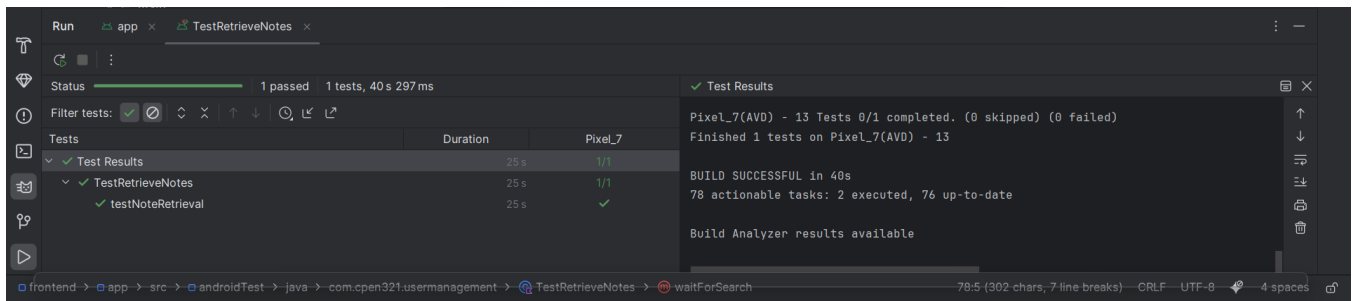


Figure 7: image info

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

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

Test Logs

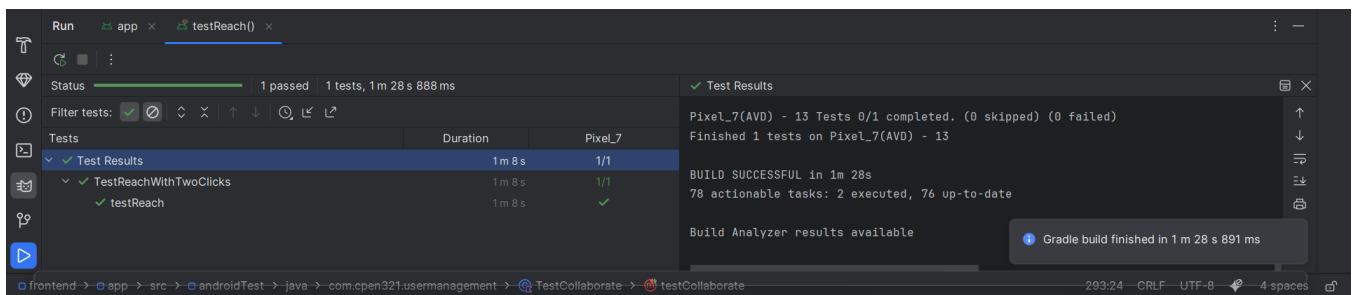


Figure 8: image info

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

Issues

ⓘ **Current** 0 ⓘ Ignored 0

⌵ Filter by Languages Severities Categori

All issues 0

Figure 10: image info