

Testing and Code Review

1. Change History

Change Date	Modified Sections	Rationale
November 28th, 2025	4 Front-end Test Specification	Added steps for how to run tests
November 28th, 2025	4.2 Tests	Updated test behavior based on new UI

2. Back-end Test Specification: APIs

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

Create a `.env.test` file under `/backend` and populate it with `JWT_SECRET={secret}`. The secret can be taken from the `/backend/.env` file

Test Organization:

- API endpoint tests are located in `backend/tests/api/mock` and `backend/tests/api/unmock` for mocked and unmocked tests respectively
- Unit tests and service tests are located in `backend/tests/non-api/mock` and `backend/tests/non-api/unmock`
- Non-functional requirement tests are located in `backend/tests/nonfunctional`

Note: All test import paths have been updated to work with the new directory structure and all test scripts are fully functional.

Tests can be run with `npm test` within the `/backend` directory

- `git clone`
- `cd ./backend`
- `npm test` Additional test commands can be found in section [2.1.3](#)

2.1.1. Tests

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks
GET /api/user/profile	<code>tests/api/unmocked/userProfile.test.ts#L13</code>	<code>tests/api/mock/userProfile.test.ts#L18</code>
POST /api/user/profile	<code>tests/api/unmocked/updateUserProfile.test.ts#L13</code>	<code>tests/api/mock/updateUserProfile.test.ts#L13</code>
DELETE /api/user/profile	<code>tests/api/unmocked/deleteUserProfile.test.ts#L13</code>	<code>tests/api/mock/deleteUserProfile.test.ts#L13</code>
GET /api/user/:userId	<code>tests/api/unmocked/getUserById.test.ts#L13</code>	<code>tests/api/mock/getUserById.test.ts#L13</code>
POST /api/auth/signin	<code>tests/api/unmocked/authSignin.test.ts#L11</code>	<code>tests/api/mock/authSignin.test.ts#L13</code>
POST /api/auth/signup	<code>tests/api/unmocked/authSignup.test.ts#L13</code>	<code>tests/api/mock/authSignup.test.ts#L13</code>
POST /api/projects	<code>tests/api/unmocked/createProject.test.ts#L13</code>	<code>tests/api/mock/createProject.test.ts#L14</code>

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks
POST /api/projects, GET /api/projects, PUT /api/projects/:projectId	tests/api/unmocked/projectManagement.test.ts#L17	tests/api/mockeds/projectManagement.test.ts#L17
PUT /api/projects/:projectId	tests/api/unmocked/updateProject.test.ts#L13	tests/api/mockeds/updateProject.test.ts#L13
DELETE /api/projects/:projectId	tests/api/unmocked/deleteProject.test.ts#L16	tests/api/mockeds/deleteProject.test.ts#L16
POST /api/projects/join	tests/api/unmocked/joinProject.test.ts#L13	tests/api/mockeds/joinProject.test.ts#L13
DELETE /api/projects/:projectId/members/:userId	tests/api/unmocked/removeMember.test.ts#L16	tests/api/mockeds/removeMember.test.ts#L16
POST /api/projects/:projectId/resources	tests/api/unmocked/addResource.test.ts#L13	tests/api/mockeds/addResource.test.ts#L13
POST /api/projects/:projectId/tasks	tests/api/unmocked/createTask.test.ts#L14	tests/api/mockeds/createTask.test.ts#L16
GET /api/projects/:projectId/tasks	tests/api/unmocked/taskManagement.test.ts#L13	tests/api/mockeds/taskManagement.test.ts#L14
GET /api/tasks/:taskId	tests/api/unmocked/taskManagement.test.ts#L13	tests/api/mockeds/taskManagement.test.ts#L14
PUT /api/tasks/:taskId	tests/api/unmocked/taskManagement.test.ts#L13	tests/api/mockeds/taskManagement.test.ts#L14
DELETE /api/tasks/:taskId	tests/api/unmocked/taskManagement.test.ts#L13	tests/api/mockeds/taskManagement.test.ts#L14

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks
POST /api/expenses	tests/api/unmocked/createExpense.test.ts#L13	tests/api/mocked/createExpense.test.ts#L14
GET /api/expenses/project/:projectId	tests/api/unmocked/expenseOperations.test.ts#L17	tests/api/mocked/expenseOperations.test.ts#L18
DELETE /api/expenses/:expenseId	tests/api/unmocked/expenseOperations.test.ts#L17	tests/api/mocked/expenseOperations.test.ts#L18
POST /api/chat/:projectId/messages	tests/api/unmocked/sendChatMessage.test.ts#L11	tests/api/mocked/sendChatMessage.test.ts#L11
GET /api/chat/:projectId/messages	tests/api/unmocked/getChatMessages.test.ts#L13	tests/api/mocked/getChatMessages.test.ts#L13
DELETE /api/chat/:projectId/messages/:messageId	tests/api/unmocked/deleteChatMessage.test.ts#L13	tests/api/mocked/deleteChatMessage.test.ts#L13
WebSocket /chat	tests/api/unmocked/chatWebSocketService.test.ts#L12	tests/non-api/mocked/chatWebSocketService.test.ts#L14
POST /api/media/upload	tests/api/unmocked/uploadImage.test.ts#L11	tests/api/mocked/uploadImage.test.ts#L16
GET /api/calendar/status	tests/api/unmocked/calendar.test.ts#L16	tests/api/mocked/calendar.test.ts#L77
GET /api/calendar/oauth/authorize	tests/api/unmocked/calendarAuthorize.test.ts#L16	tests/api/mocked/calendarAuthorize.test.ts#L77

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks
POST /api/calendar/enable	tests/api/unmocked/calendarEnable.test.ts#L13	tests/api/mockeds/calendarEnable.test.ts#L13
POST /api/calendar/disable	tests/api/unmocked/calendarDisable.test.ts#L13	tests/api/mockeds/calendarDisable.test.ts#L13
POST /api/calendar/disconnect	tests/api/unmocked/calendarDisconnect.test.ts#L13	tests/api/mockeds/calendarDisconnect.test.ts#L13

2.1.2. Commit Hash Where Tests Run

30fc6c4738ccfd3c793189c42fdadc8c9d951840

2.1.3. How to Run the Tests

1. Clone the Repository:

```
git clone https://github.com/justinishiguro/cpen-321-project.git
cd cpen-321-project/backend
```

2. Install Dependencies:

```
npm install
```

3. Set Up Environment Variables:

- Create a `.env` file in the backend directory
- Add required environment variables (JWT_SECRET, MongoDB connection string, etc.)

4. Run All Tests:

```
npm run test
```

5. Run Tests Without Mocks Only (API Tests):

```
npm run test:api:unmocked
```

6. Run Tests With Mocks Only (API Tests):

```
npm run test:api:mocked
```

7. Run All API Tests:

```
npm run test:api
```

8. Run Unit/Service Tests (Non-API):

```
npm run test:non-api
```

9. Run Non-Functional Requirement Tests Only:

```
npm run test:nonfunctional
```

10. Run Tests with Coverage Report:

```
npm run test:coverage
```

2.2. GitHub Actions Configuration Location

[~/.github/workflows/main.yml](#)

2.3. Jest Coverage Report Screenshots for Tests Without Mocking

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered	Line #s
All files	44.29	33.24	50	44.22		
src	0	0	0	0		
index.ts	0	0	0	0	1-38	
task.routes.ts	0	100	0	0	1-39	
src/database	7.18	5.26	9.09	6.28		
database-init.ts	0	0	0	0	1-209	
database.ts	56.52	50	40	52.38	12,20-26,31-35	
show-database.ts	0	0	0	0	1-110	
src/database/scripts	0	0	0	0		
database-reset.ts	0	0	0	0	1-357	
database-seed.ts	0	0	0	0	1-270	
database-status.ts	0	0	0	0	1-43	
src/features/auth	54.28	28.57	85.71	54.28		
auth.controller.ts	54.83	50	100	54.83	21,35-48,62-65,79-92	
auth.service.ts	51.35	0	80	51.35	23-32,45,55-66,78-85	
auth.types.ts	100	100	100	100		
src/features/calendar	42.35	21.27	58.82	42.35		
calendar.controller.ts	56.38	41.66	83.33	56.38	19-20,28-90,125-126,132-133,156-157,169-170,176-177,194-195,207-208,218-219,231-232,238-239,261-262	
calendar.service.ts	25	0	45.45	25	55-69,89-230,244,258	
src/features/chat	51.4	46.87	53.33	50.94		
chat.websocket.service.ts	33.33	0	50	33.33	20-24,47-61,68-105,117,128	
chat.controller.ts	73.58	56.52	100	73.07	23-25,34-35,57-58,86,94-95,111-113,128-129,152-153,168-170,187-201	
chatMessage.model.ts	25.92	33.33	33.33	25.92	70-71,80-101,119-210	
src/features/expenses	49.24	64.1	54.54	50		
expense.controller.ts	75.8	78.12	100	75.4	12-13,60-61,69-70,113-114,122-123,146-155	
expense.model.ts	25.71	0	33.33	26.86	99-100,110-111,122-169,177-241	
src/features/invitations	0	0	0	0		
invitation.model.ts	0	0	0	0	1-281	
src/features/media	60.46	12.5	50	63.41		
media.controller.ts	73.33	25	100	73.33	35-43	
media.service.ts	53.57	0	40	57.69	17-33,40,48	
src/features/notifications	0	0	0	0		
notification.model.ts	0	0	0	0	1-284	
src/features/projects	58.21	43.18	76.31	58.09		
project-status.ts	0	0	0	0	1-61	
project.controller.ts	66.26	47.43	78.94	66.24	...14,326-327,331-332,363-364,385-386,405-407,411-412,416-417,423-424,444-445,466-467,479-480,486-487,508-509,516-517,546-547	
project.model.ts	59.74	25	93.33	60.27	123-124,132-133,141-142,150-151,162-163,175-176,188-189,201-216,224-225,235-236,257-258	
src/features/tasks	56	30.64	69.23	56.11		
task.controller.ts	64.22	38.77	78.57	64.6	...60-75,110-112,124-128,142-143,166-167,181-182,212-213,232,253-254,283-284,295-296,308-309,320-321,336-337,345-343,444-465	
task.model.ts	35.48	0	58.33	35.48	88-93,101-105,118-174,182-186,194-230,243-247	
src/features/users	48.78	16.66	70.58	49.18		
user.controller.ts	60	15.38	75	60	11-13,24,49-57,72,82-90,106-114	
user.model.ts	41.25	20	69.23	41.77	82-87,107-108,116-117,131-165,177-191,203-204,216-232,243-244	
user.types.ts	100	100	100	100		
src/middleware	74.46	62.5	66.66	72.09		
auth.middleware.ts	70.37	57.14	100	69.23	28-32,38-42,57-65	
errorHandler.middleware.ts	62.5	100	0	50	6,16-18	
validation.middleware.ts	91.66	100	100	90.9	23	
src/routes	94.59	0	100	94.59		
auth.routes.ts	100	100	100	100		
calendar.routes.ts	100	100	100	100		
chat.routes.ts	100	100	100	100		
dev-calendar.routes.ts	71.42	0	100	71.42	16-17,37-38,49-50,62-63	
expense.routes.ts	100	100	100	100		
media.routes.ts	100	100	100	100		
project.routes.ts	100	100	100	100		
routes.ts	100	100	100	100		
task.routes.ts	100	100	100	100		
user.routes.ts	100	100	100	100		
src/utils	87.87	45.45	90	87.09		
logger.utils.ts	85.71	100	75	85.71	14	
multer.config.ts	86.66	33.33	100	86.66	9,30	
sanitizeInput.utils.ts	90.9	50	100	88.88	14	

Test Suites: 27 passed, 27 total
 Tests: 123 passed, 123 total
 Snapshots: 0 total
 Coverage: 70.5% (113 files)

2.4. Jest Coverage Report Screenshots for Tests With Mocking

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	48.46	40.89	51.15	48.47	
src/index.ts	0	0	0	0	1-38
src/task.routes.ts	0	100	0	0	1-39
src/database/database-init.ts	7.18	5.26	9.09	6.28	
src/database/database.ts	56.52	50	40	52.38	12,20-26,31-35
src/database/scripts/database-reset.ts	0	0	0	0	1-110
src/database/scripts/database-seed.ts	0	0	0	0	1-357
src/database/scripts/database-status.ts	0	0	0	0	1-270
src/features/auth/auth.controller.ts	70	50	85.71	70	
src/features/auth/auth.service.ts	90.32	87.5	100	90.32	41-48
src/features/auth/auth.types.ts	51.35	0	80	51.35	23-32,45,55-66,78-85
src/features/calendar/calendar.controller.ts	49.41	29.78	58.82	49.41	38-90,125-126,132-133,156-157,176-177,194-195,238-239,261-262
src/features/calendar.service.ts	69.14	58.33	83.33	69.14	55-69,89-230,244,258
src/features/chat/chat.websocket.service.ts	66.82	65.62	60	66.5	
src/features/chat/chat.controller.ts	57.4	33.33	66.66	57.4	51-52,68-105,117,128
src/features/chat/chatMessage.model.ts	92.45	78.26	100	92.3	86,128-129,152-153,188-189,198
src/features/expenses/expense.controller.ts	52.27	66.66	54.54	53.12	
src/features/expenses/expense.model.ts	82.25	81.25	100	81.96	69-70,113-114,122-123,146-155
src/features/invitations/invitation.model.ts	25.71	0	33.33	26.86	99-100,110-111,122-169,177-241
src/features/media/media.controller.ts	69.76	50	50	73.17	
src/features/media/media.service.ts	100	100	100	100	
src/features/notifications/notification.model.ts	53.57	0	40	57.69	17-33,40,48
src/features/projects/project.model.ts	59.88	48.86	76.31	59.82	
src/features/projects/project.controller.ts	0	0	0	0	1-61
src/features/projects/project.middleware.ts	68.69	53.84	78.94	68.77	...86,313-314,326-327,331-332,363-364,385-386,405-407,416-417,423-424,444-445,466-467,479-480,486-487,508-509,516-517,546-547
src/features/tasks/task.controller.ts	59.74	25	93.33	60.27	123-124,132-133,141-142,150-151,162-163,175-176,188-189,201-216,224-225,235-236,257-258
src/routes/auth.routes.ts	60	33.87	69.23	60.18	
src/routes/calender.routes.ts	69.82	42.85	78.57	70.35	...32,44-46,69-70,110-112,142-143,166-167,181-182,212-213,232,253-254,283-284,295-296,308-309,320-321,336-337,345-432,444-465
src/routes/chat.routes.ts	35.48	0	58.33	35.48	88-93,101-105,118-174,182-186,194-230,243-247
src/routes/dev-calendar.routes.ts	56.09	44.44	76.47	56.55	
src/routes/users/user.controller.ts	82.5	53.84	100	82.5	24,49-57,90,114
src/routes/users/user.model.ts	41.25	20	69.23	41.77	82-87,107-108,116-117,131-165,177-191,203-204,216-232,243-244
src/middleware/auth.middleware.ts	100	100	100	100	
src/middleware/errorHandler.middleware.ts	74.46	62.5	66.66	72.09	
src/middleware/validation.middleware.ts	70.37	57.14	100	69.23	28-32,38-42,57-65
src/routes/project.routes.ts	62.5	100	0	50	6,16-18
src/routes/routes.ts	91.66	100	100	90.9	23
src/routes/task.routes.ts	94.59	0	100	94.59	
src/routes/user.routes.ts	100	100	100	100	
src/routes/calender.routes.ts	100	100	100	100	
src/routes/chat.routes.ts	100	100	100	100	
src/routes/dev-calendar.routes.ts	100	100	100	100	
src/routes/expense.routes.ts	100	100	100	100	
src/routes/media.routes.ts	100	100	100	100	
src/routes/project.routes.ts	100	100	100	100	
src/routes/routes.ts	100	100	100	100	
src/routes/task.routes.ts	100	100	100	100	
src/routes/user.routes.ts	100	100	100	100	
src/utils/logger.utils.ts	87.87	45.45	90	87.09	
src/utils/logger.utils.ts	85.71	100	75	85.71	14
src/utils/multer.config.ts	86.66	33.33	100	86.66	9,30
src/utils/sanitizeInput.util.ts	90.9	50	100	88.88	14

2.5. Jest Coverage Report Screenshots for Both Tests With and Without Mocking

File	%Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	56.37	43.85	55.83	56.46	
src	0	100	0	0	
task.routes.ts	0	100	0	0	1-39
src/database	17.1	7.14	14.28	15.06	
database.ts	56.52	59	48	52.38	12,20-26,31-35
show-database.ts	0	0	0	0	1-110
src/features/auth	70	59	85.71	70	
auth.controller.ts	90.32	87.5	100	90.32	41-48
auth.service.ts	51.35	0	80	51.35	23-32,45,55-66,78-85
auth.types.ts	100	100	100	100	
src/features/calendar	49.41	29.78	58.82	49.41	
calendar.controller.ts	69.14	58.33	83.33	69.14	38-90,125-126,132-133,156-157,176-177,194-195,238-239,261-262
calendar.service.ts	25	0	45.45	25	55-69,89-230,244,258
src/features/chat	68.69	71.87	63.33	68.39	
chat-websocket.service.ts	57.4	33.33	66.66	57.4	51-52,68-105,117,128
chat.controller.ts	94.33	86.95	100	94.23	86,152-153,188-189,198
chatMessage.model.ts	29.62	33.33	41.66	29.62	70-71,80-101,119-163,172-210
src/features/expenses	52.27	66.66	54.54	53.12	
expense.controller.ts	82.25	81.25	100	81.96	69-70,113-114,122-123,146-155
expense.model.ts	25.71	0	33.33	26.86	99-100,110-111,122-169,177-241
src/features/invitations	0	0	0	0	
invitation.model.ts	0	0	0	0	1-281
src/features/media	69.76	50	56	73.17	
media.controller.ts	100	100	100	100	
media.service.ts	53.57	0	40	57.69	17-33,40,48
src/features/notifications	0	0	0	0	
notification.model.ts	0	0	0	0	1-284
src/features/projects	59.88	48.86	76.31	59.82	
project.status.ts	0	0	0	0	1-61
project.controller.ts	68.69	53.84	78.94	68.77	...86,313-314,326-327,331-332,363-364,385-386,405-407,416-417,423-424,444-445,466-467,479-480,486-487,508-509,516-517,546-547
project.model.ts	59.74	25	93.33	60.27	123-124,132-133,141-142,150-151,162-163,175-176,188-189,201-216,224-225,235-236,257-258
src/features/tasks	60	33.87	69.23	60.18	
task.controller.ts	69.82	42.85	78.57	70.35	...32,44-46,69-70,110-112,142-143,166-167,181-182,212-213,232,253-254,283-284,295-296,308-309,320-321,336-337,345-432,444-465
task.model.ts	35.48	0	58.33	35.48	88-93,101-105,118-174,182-186,194-230,243-247
src/features/users	56.09	44.44	76.47	56.55	
user.controller.ts	82.5	53.84	100	82.5	24,49-57,98,114
user.model.ts	41.25	20	69.23	41.77	82-87,107-108,116-117,131-165,177-191,203-204,216-232,243-244
user.types.ts	100	100	100	100	
src/middleware	74.46	62.5	66.66	72.09	
auth.middleware.ts	70.37	57.14	100	69.23	28-32,38-42,57-65
errorhandler.middleware.ts	62.5	100	0	50	6,16-18
validation.middleware.ts	91.66	100	100	90.9	23
src/routes	94.59	0	100	94.59	
auth.routes.ts	100	100	100	100	
calendar.routes.ts	100	100	100	100	
chat.routes.ts	100	100	100	100	
dev-calendar.routes.ts	71.42	0	100	71.42	16-17,37-38,49-50,62-63
expense.routes.ts	100	100	100	100	
media.routes.ts	100	100	100	100	
project.routes.ts	100	100	100	100	
routes.ts	100	100	100	100	
task.routes.ts	100	100	100	100	
user.routes.ts	100	100	100	100	
src/utils	87.87	45.45	90	87.09	
logger.util.ts	85.71	100	75	85.71	14
multer.config.ts	86.66	33.33	100	86.66	9,30
sanitizeInput.util.ts	90.9	50	100	88.88	14

Test Suites: 57 passed, 57 total
 Tests: 288 passed, 288 total
 Snapshots: 0 total
 Time: 15.763 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)	tests/nonfunctional/response_time.test.ts
Data Security & Authorization	tests/nonfunctional/data_security.test.ts
Real-time Communication (WebSocket)	tests/non-api/mocked/chatWebSocketService.test.ts & tests/api/unmocked/chatWebSocketService.test.ts

3.2. Test Verification and Logs

- Performance (Response Time)

- **Verification:** This test suite measures API response times to ensure acceptable performance under normal operating conditions. It includes three main test scenarios: (1) Health check endpoint should respond within 200ms, (2) Multiple concurrent requests should complete within 1000ms total, and (3) Sequential requests should maintain consistent performance with each request under 500ms. The tests use Jest with supertest to measure actual response times and validate that the system meets performance requirements for good user experience.

- **Test Cases:**

- **should respond to health check within 200ms:** Tests basic endpoint responsiveness
- **should handle multiple concurrent requests within acceptable time:** Tests system performance under concurrent load (5 simultaneous requests)
- **should maintain performance under sequential load:** Tests consistent performance over 10 sequential requests

- **Performance Thresholds:**

- Individual request: < 500ms
- Concurrent batch (5 requests): < 1000ms total
- Health check: < 200ms
- Average sequential performance: < 200ms

- **Log Output Example**

```

⌚ Testing health check endpoint response time...
⌚ Response time: 45ms
⌚ Testing concurrent request performance...
⌚ Total time for 5 requests: 234ms
⌚ Average time per request: 46.8ms
⌚ Testing sequential request performance...
⌚ Average response time across 10 requests: 52ms
⌚ Maximum response time: 78ms

```

• Data Security & Authorization

- **Verification:** This test suite validates that users can only access data they are authorized to view, ensuring proper data isolation and privacy protection. It tests three main security areas: (1) Chat message access control - users cannot read messages from projects they're not members of, (2) Project access security - users cannot see projects they don't have access to, and (3) Token security validation - invalid tokens are properly rejected and token validation performs efficiently under load.
- **Test Cases:**
 - `should deny access to chat messages from unauthorized project`: Tests data isolation between user groups
 - `should allow access to chat messages for authorized project member`: Validates legitimate access works correctly
 - `should prevent unauthorized users from sending messages to private projects`: Tests write access control
 - `should deny unauthorized access to project information`: Tests project-level data privacy
 - `should allow authorized access to own project information`: Validates authorized project access
 - `should reject requests with invalid JWT tokens`: Tests token validation security
 - `should reject requests without authentication tokens`: Tests missing token handling
 - `should validate token performance under load`: Tests authentication performance under repeated requests
- **Security Thresholds:**
 - Access denial response time: < 500ms
 - Token validation time: < 100ms per request
 - Average token validation performance: < 50ms
 - All unauthorized access attempts: Must return 401/403 status codes

4. Front-end Test Specification

These steps assume that `local.properties` and `.env` have been created and the backend is running locally with `npm run dev`. **Front-end Test Prerequisites:**

1. Create a new device (Pixel 7 - Android 13.0 "Tiramisu")
2. Ensure that a Google account is available on the emulator (the tests select the first account shown after clicking "Sign In With Google").
3. Ensure that a project named "Test1" was previously created or that the test device can create one during the run (the tests assume a project named "Test1" exists).
4. Ensure that your "Name" under "Manage Profile" is "Test User" (this name is utilized in tests).
5. Note: The app uses generated invitation codes for project invites (email-invite UI is not implemented). Front-end tests check that a generated invitation code is displayed after project creation.

How To Run Front-end Tests:

1. Open the repository in Android Studio
2. Run the (Pixel 7 - Android 13.0 "Tiramisu") device you created
3. Open the `MyComposeTest.kt` file in Android Studio and click "Run" on each test individually

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

`cpen-321-project\frontend\app\src\androidTest\java\com\cpen321\usermanagement\MyComposeTest.kt`

4.2. Tests

• Use Case: Creating a New Project (Use Case 2)

- **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. The app displays a "Create New Project" button which the user clicks	Check button labelled "Create New Project" is present on screen. Click button labelled "Create New Project".
2. The app displays a "Create New Project" form	Check "Create New Project" form is present on screen.
5a. The user inputs an empty project name	Check button labelled "Create" is disabled.
3. The user inputs a non-empty project name and optional description	Input "ProjectRandomNumber" under "Project Name".
4. The user clicks the "Create" button	Check button labelled "Create" is enabled. Click "Create" button.
5. The app displays a generated invitation code for the new project	Check that an invitation code or "Share code" UI element is visible on screen and that the code can be copied or copied via a button.

Scenario Steps	Test Case Steps
6. The user sees the new project created on the home page	Check project "ProjectRandomNumber" is visible on the screen
7. The user clicks on the created project	Click button labelled "ProjectRandomNumber".
10. The user clicks on the Settings button	Check the button labelled "Settings" is visible Click button labelled "Settings".
11. The user clicks on the Delete button	Check the button labelled "Delete" is visible Click button labelled "Delete".
12. The user does not see the project on the home page	Check project "ProjectRandomNumber" is not visible on the screen

- **Test Logs:**

```

Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_7(AVD) - 13

Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Pixel_7(AVD) - 13 Tests 1/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13

BUILD SUCCESSFUL in 2m 13s
72 actionable tasks: 1 executed, 71 up-to-date

```

Tests	Duration	Pixel_7
✓ Test Results	40s	1/1
✓ MyComposeTest	40s	1/1
✓ create_project	40s	✓
✓ com.cpen321.usermanagement.MyComposeTest.create_project		
Pixel_7	Passed	

- **Use Case: Adding Project Expenses (Use Case 4)**

- **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. The user opens an existing project	Click on "Test1" project to open project screen.
2. The app displays an "Expense" button which the user clicks	Check button labelled "Expense" is present on screen. Click button labelled "Expense".
3. The app displays an "Add Expense" button which the user clicks	Check button labelled "Add Expense" is present on screen. Click button labelled "Add Expense".
4. The app displays an "Add New Expense" form	Check "Add New Expense" form is present on screen.
5a. The user inputs an empty description	Click "Add Expense" button on form.
5a1. The app displays an error message prompting the user to input valid values	Check dialog is opened with text: "Please fill all fields correctly".
5b. The user inputs a non-numeric amount	Input "Test randomAmount" in "Description" text field. Input "NON_INTEGER" in "Amount" text field. Select "Arnav Prasad" in "Paid By" dropdown. Select "Arnav Prasad" in "Split Between" section. Click "Add Expense" button on form.
5b1. The app displays an error message prompting the user to input valid values	Check dialog is opened with text: "Please fill all fields correctly".
5c. The user does not select who paid	Input "Test randomAmount" in "Description" text field. Input "randomAmount" in "Amount" text field. Select "Arnav Prasad" in "Split Between" section. Click "Add Expense" button on form.
5c1. The app displays an error message prompting the user to input valid values	Check dialog is opened with text: "Please fill all fields correctly".

Scenario Steps	Test Case Steps
5d. The user does not select who to split expense between	Input "Test randomAmount" in "Description" text field. Input "randomAmount" in "Amount" text field. Select "Arnav Prasad" in "Paid By" dropdown. Click "Add Expense" button on form.
5d1. The app displays an error message prompting the user to input valid values	Check dialog is opened with text: "Please fill all fields correctly".
5. The user inputs valid inputs	Input "Test randomAmount" in "Description" text field. Input "randomAmount" in "Amount" text field. Select "Arnav Prasad" in "Paid By" dropdown. Select "Arnav Prasad" in "Split Between" section.
6. The user clicks the "Add Expense" button	Click "Add Expense" button on form.
7. The user can view the new expense	Check the description "Test (randomAmount)" is present on screen. Check the amount "(randomAmount)" is present on screen. Check the paid by user "Arnav Prasad" is present on screen. Check the split between user(s) "Arnav Prasad" is present on the screen.

- **Test Logs:**

```

Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_7(AVD) - 13
Connected to process 21754 on device 'Pixel_7 [emulator-5554]'.

Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13

BUILD SUCCESSFUL in 56s
72 actionable tasks: 1 executed, 71 up-to-date

```

Test Results

✓ Test Results	26 s	1/1
✓ MyComposeTest	26 s	1/1
✓ add_expense	26 s	✓
✓ com.cpen321.usermanagement.MyComposeTest.add_expense		
Pixel_7	Passed	

- **Use Case: Creating/Assigning Project Tasks and Deadlines to Group Members (Use Case 5)**

- **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. The user opens an existing project	Click on "Test1" project to open project screen.
2. The app displays a "Task Board" button which the user clicks	Check button labelled "Task Board" is present on screen. Click button labelled "Task Board".
3. The app displays a "Create Task" button which the user clicks	Check button labelled "Create Task" is present on screen. Click button labelled "Create Task".
4. The app displays an "Create New Task" form	Check "Create New Task" form is present on screen.
5a. The user inputs an empty task name	Select "Arnav Prasad" under "Assignee" dropdown. Select "TODAY_DATE" under "Deadline". Select "In progress" under "Status" dropdown. Click "Create" button.
5a1. The app displays an error message prompting the user to input a non empty task	Check dialog is opened with text: "Task name cannot be empty".
5b. The user does not select an "Assignee"	Input "Task1" under "Task Name". Select "TODAY_DATE" under "Deadline". Select "In progress" under "Status" dropdown. Click "Create" button.
5b1. The app displays an error message prompting the user to input a non empty Assignee	Check dialog is opened with text: "Assignee cannot be empty".

Scenario Steps	Test Case Steps
5c. The user selects a date that has already passed	Input "Task1" under "Task Name". Select "Arnav Prasad" under "Assignee" dropdown. Select "In progress" under "Status" dropdown. Select "TODAY_DATE - 1" under "Deadline". Click "Create" button.
5c1. The app displays an error message prompting the user to input a future date	Check dialog is opened with text: "Please enter a future date".
5. The user inputs valid inputs	Input "TaskRandomNumber" under "Task Name". Select "Arnav Prasad" under "Assignee" dropdown. Select "In progress" under "Status" dropdown. Select "TODAY_DATE" under "Deadline".
6. The user clicks the "Create" button	Click "Create" button.
7. The user can view the new task	Check the Task Name "TaskRandomNumber" is present on screen. Check the Assignee "Arnav Prasad" is present on screen. Check the date "TODAY_DATE" is present on screen. Check the status "In progress" is present on screen.

- **Test Logs:**

```

Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_7(AVD) - 13

Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Pixel_7(AVD) - 13 Tests 1/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13

BUILD SUCCESSFUL in 57s
72 actionable tasks: 1 executed, 71 up-to-date
  
```

✓	Test Results	41s	1/1
✓	MyComposeTest	41s	1/1
✓	create_task	41s	✓
✓ com.copen321.usermanagement.MyComposeTest.create_task			
Pixel_7	Passed		

5. Automated Code Review Results

5.1. Commit Hash Where Codacy Ran

18d5b90ab464e7737273a77360d61c5ebfe8f45c

5.2. Unfixed Issues per Codacy Category

Issues

Current 0 Ignored 0

Filter by Languages ▾ Severities ▾ Categories ▾ Authors ▾ Tags ▾

All issues 0

No current issues found
We couldn't find any current issues right now.

5.3. Unfixed Issues per Codacy Code Pattern

Not applicable as there are no Codacy issues (see above image).

5.4. Justifications for Unfixed Issues

Not applicable as there are no Codacy issues (see above image).