

Example M5: Testing and Code Review

1. Change History

Change Date	Modified Sections	Rationale
-------------	-------------------	-----------

Nothing to show

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

2.1.1. Tests

Interface	URL	Type	Test without Mocks	Test with Mocks	Mocked Components
Get All Admins	/admins	GET	AdminController.test.ts#L9	AdminController.test.ts#L38	DB
Get Admin by ID	/admins/:id	GET	AdminController.test.ts#L19	AdminController.test.ts#L50	DB
Create Admin	/admins	POST	AdminController.test.ts#L34	AdminController.test.ts#L62	DB
Update Admin	/admins/:id	PUT	AdminController.test.ts#L49	AdminController.test.ts#L78	DB
Delete Admin	/admins/:id	DELETE	AdminController.test.ts#L61	AdminController.test.ts#L90	DB
Get All Games	/games	GET	GameController.test.ts#L12	GameController.test.ts#L28	DB
Get Game by ID	/games/:id	GET	GameController.test.ts#L22	GameController.test.ts#L40	DB
Create Game	/games	POST	GameController.test.ts#L34	GameController.test.ts#L52	DB
Update Game	/games/:id	PUT	GameController.test.ts#L49	GameController.test.ts#L64	DB
Delete Game	/games/:id	DELETE	GameController.test.ts#L61	GameController.test.ts#L76	DB
Get All Groups	/groups	GET	GroupController.test.ts#L23	GroupController.test.ts#L36	group DB
Get Group by ID	/groups/:group_id	GET	GroupController.test.ts#L30	GroupController.test.ts#L48	group DB
Create Group	/groups	POST	GroupController.test.ts#L37	GroupController.test.ts#L60	group DB
Update Group	/groups/:group_id	PUT	GroupController.test.ts#L44	GroupController.test.ts#L72	group DB
Delete Group	/groups/:group_id	DELETE	GroupController.test.ts#L51	GroupController.test.ts#L84	group DB
Join Group	/groups/:group_id/join	POST	GroupController.test.ts#L74	GroupController.test.ts#L96	group DB
Leave Group	/groups/:group_id/leave	DELETE	GroupController.test.ts#L87	GroupController.test.ts#L108	group DB
Get Group Members	/groups/:group_id/members	GET	Not available	GroupController.test.ts#L120	group DB
Get Group URL	/groups/:group_id/url	GET	GroupController.test.ts#L100	GroupController.test.ts#L132	group DB
Initiate Matchmaking	/matchmaking/initiate	POST	Not available	MatchmakingController.test.ts#L47	database, redis
Check Matchmaking Status	/matchmaking/status/:discord_id	GET	Not available	MatchmakingController.test.ts#L139	database, redis
Get All Preferences	/preferences	GET	PreferencesController.test.ts#L28	PreferencesController.test.ts#L39	database
Get Preferences by ID	/preferences/:id	GET	PreferencesController.test.ts#L36	PreferencesController.test.ts#L51	database
Create Preferences	/preferences	POST	PreferencesController.test.ts#L49	PreferencesController.test.ts#L63	database
Update Preferences	/preferences/:id	PUT	PreferencesController.test.ts#L61	PreferencesController.test.ts#L75	database
Delete Preferences	/preferences/:id	DELETE	Not available	PreferencesController.test.ts#L87	database
Get Preferences by User ID	/preferences/user/:userId	GET	PreferencesController.test.ts#L43	PreferencesController.test.ts#L99	database
Get All Reports	/reports	GET	Not available	ReportController.test.ts#L49	database

Interface	URL	Type	Test without Mocks	Test with Mocks	Mocked Components
Get Report by ID	/reports/:id	GET	Not available	ReportController.test.ts#L64	database
Create Report	/reports	POST	Not available	ReportController.test.ts#L98	database
Resolve Report	/reports/:id/resolve	PUT	Not available	ReportController.test.ts#L144	database
Delete Report	/reports/:id	DELETE	Not available	ReportController.test.ts#L191	database
Get All Users	/users	GET	UserController.test.ts#L14	UserController.test.ts#L41	database
Get User by ID	/users/:id	GET	UserController.test.ts#L23	UserController.test.ts#L54	database
Create User	/users	POST	UserController.test.ts#L32	UserController.test.ts#L67	database
Update User	/users/:id	PUT	UserController.test.ts#L41	UserController.test.ts#L80	database
Delete User	/users/:id	DELETE	UserController.test.ts#L50	UserController.test.ts#L93	database
Get User Groups	/users/:id/groups	GET	UserController.test.ts#L74	UserController.test.ts#L106	database
Ban User	/users/:id/ban	PUT	UserController.test.ts#L97	UserController.test.ts#L119	database
Protect Endpoint	/auth/protect	Not available	AuthController.test.ts#L9	session	
Handle Register	/auth/register	Not available	AuthController.test.ts#L31	session, axios (DB)	
Handle LoginOrRedirect	/auth/login-or-redirect	Not available	AuthController.test.ts#L75	session, axios (DB)	
Handle Logout	/auth/logout	Not available	AuthController.test.ts#L108	session	
Handle Discord Callback	/auth/discord-callback	Not available	AuthController.test.ts#L130	session, axios (Discord API), axios (DB)	

2.1.2. Commit Hash Where Tests Run

13ed3639a32a3011a88a8c7c4c0cacd565c83c25

2.1.3. Explanation on How to Run the Tests

- 1. Navigate to [GitHub Actions](#)
- 2. Click "Deploy Test Environment"
- 3. Click "Run Workflow"
- 4. Click the launched workflow
- 5. Await test results

2.2. GitHub Actions Configuration Location

~/.github/workflows/test-deployment.yml

2.3. Jest Coverage Report Screenshots With Mocks

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	60.55	41.49	47.52	59.33	
src	8.19	20	0	8.19	
data-source.ts	0	100	100	0	1-14
index.ts	0	0	0	0	1-101
redis_client.ts	45.45	33.33	0	45.45	6,12-17
src/controllers	77.22	45.8	94.11	75.84	
AdminController.ts	69.84	0	100	67.24	10,24-27,39-40,52-53,61,74-80,94-98
AuthController.ts	89.1	85.71	90	89	105-106,112-116,154,193-194,242-243
GameController.ts	80.7	28.57	100	78.84	9,22-25,40-41,45,62-64,80-82
GroupController.ts	67.76	19.23	90	65.49	...72-180,204-205,212-221,241-244,258-261,276-298
MatchmakingController.ts	89.47	66.66	100	88.88	11-12,51,53
PreferencesController.ts	73.62	42.1	100	71.76	...70-71,84-85,93-94,106,132-133,141-143,159-161
ReportController.ts	82.5	55	100	81.33	17,52-53,73-74,81-82,108-117,134-135
UserController.ts	76.54	28.57	87.5	75.34	10,24-27,45-49,67-69,85-87,103-107,124-126
src/entity	75.49	100	0	78.75	
Admin.ts	87.5	100	0	83.33	9
Game.ts	81.81	100	0	87.5	15
Group.ts	72.22	100	0	78.57	13,19,23
GroupMember.ts	66.66	100	0	72.72	10,14,18
Preference.ts	80	100	0	83.33	19,23
Report.ts	76.47	100	0	73.33	10,14,18,25
User.ts	72.22	100	0	78.57	16,22,25
src/routes	0	100	0	0	
AdminRoutes.ts	0	100	100	0	1-18
AuthRoutes.ts	0	100	0	0	1-18
GameRoutes.ts	0	100	100	0	1-18
GroupRoutes.ts	0	100	100	0	1-26
MatchmakingRoutes.ts	0	100	100	0	1-9
PreferencesRoutes.ts	0	100	100	0	1-20
ReportRoutes.ts	0	100	100	0	1-19
UserRoutes.ts	0	100	100	0	1-22
src/services	14.14	0	0	15.38	
MatchmakingService.ts	14.14	0	0	15.38	28-42,46-51,55-83,90-206

2.4. Jest Coverage Report Screenshots Without Mocks

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	0	0	0	0	
src	0	0	0	0	
data-source.ts	0	100	100	0	1-14
index.ts	0	0	0	0	1-101
redis_client.ts	0	0	0	0	1-17
src/controllers	0	0	0	0	
AdminController.ts	0	0	0	0	2-102
AuthController.ts	0	0	0	0	2-262
GameController.ts	0	0	0	0	2-86
GroupController.ts	0	0	0	0	2-298
MatchmakingController.ts	0	0	0	0	2-59
PreferencesController.ts	0	0	0	0	2-165
ReportController.ts	0	0	0	0	2-141
UserController.ts	0	0	0	0	2-130
src/entity	0	100	0	0	
Admin.ts	0	100	0	0	1-11
Game.ts	0	100	0	0	1-16
Group.ts	0	100	0	0	1-27
GroupMember.ts	0	100	0	0	1-19
Preference.ts	0	100	0	0	1-25
Report.ts	0	100	0	0	1-29
User.ts	0	100	0	0	1-26
src/routes	0	100	0	0	
AdminRoutes.ts	0	100	100	0	1-18
AuthRoutes.ts	0	100	0	0	1-18
GameRoutes.ts	0	100	100	0	1-18
GroupRoutes.ts	0	100	100	0	1-26
MatchmakingRoutes.ts	0	100	100	0	1-9
PreferencesRoutes.ts	0	100	100	0	1-20
ReportRoutes.ts	0	100	100	0	1-19
UserRoutes.ts	0	100	100	0	1-22
src/services	0	0	0	0	
MatchmakingService.ts	0	0	0	0	1-206

Test Suites: 5 passed, 5 total
 Tests: 39 passed, 39 total
 Snapshots: 0 total
 Time: 28.667 s

2.5 Justification for Low Test Coverage

As you can see, our tests are configured to run against a dockerized backend so Jest cannot properly instrument the code. Thus, we only collect code coverage in mocked tests and do not collect any coverage data from unmocked tests

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

3.1. Test Locations in Git

Non-Functional Requirement	Location in Git
Endpoints Need To Be Secured By Sessions	EndpointSecurity.test.ts#L49
Report Reasoning Can Not Exceed 500 Characters	EndpointSecurity.test.ts#L49

3.2. Test Verification and Logs

- **Performance (Response Time)**
 - **Verification:**
 - Endpoints Need To Be Secured By Sessions: We have a system that verifies users through sessions to ensure they are authenticated via Discord OAuth, which is critical for the security and viability of our platform. To enforce this, we protect key endpoints by validating sessions on incoming HTTPS requests. In our test suite, we simulate unauthorized access by making HTTPS requests without cookies, where the session data would normally be stored, and expect protected endpoints like /auth/register and /auth/logout to return a 401 Unauthorized status. While there are additional protected endpoints, these two are the most important for demonstration purposes.
 - Report Reasoning Can Not Exceed 500 Characters: Users can report other users they have formed groups with, allowing admins to review reports and help maintain a safer environment on our platform. To prevent database bloat and potential security issues, such as memory exhaustion, we enforce a 500-character limit on the report reason in the backend. To ensure this validation works as expected, we've added a test that submits a report with more than 500 characters and checks that a 400 Bad Request response is returned.

```
PORT=3000 npm test "tests/non-functional"
> backend@1.0.0 test
> jest --silent tests/non-functional

PASS tests/non-functional/EndpointSecurity.test.ts
PASS tests/non-functional/ReportReasonLength.test.ts

File                                     % Stmts % Branch % Funcs % Lines Uncovered Line #s
-----
All files                               10.96    4.08    0.99    9.77
src                                     18.03      0      0      18.03
  data-source.ts                        100      100    100      100
  index.ts                              0        0      0      0 1-101
  redis_client.ts                       0        0      0      0 1-17
src/controllers                         3.31    4.58    1.96    2.73
  AdminController.ts                   0        0      0      0 2-102
  AuthController.ts                    0        0      0      0 2-262
  GameController.ts                    0        0      0      0 2-86
  GroupController.ts                     0        0      0      0 2-298
  MatchmakingController.ts              0        0      0      0 2-59
  PreferencesController.ts              0        0      0      0 2-165
  ReportController.ts                   27.5     30     20    22.66 11-20, 25-38, 47-48, 56-96, 101-123, 128-141
  UserController.ts                     0        0      0      0 2-130
src/entity                             75.49    100      0    78.75
  Admin.ts                             87.5     100      0    83.33 9
  Game.ts                              81.81    100      0    87.5 15
  Group.ts                             72.22    100      0    78.57 13, 19, 23
  GroupMember.ts                       66.66    100      0    72.72 10, 14, 18
  Preference.ts                         80      100      0    83.33 19, 23
  Report.ts                             76.47    100      0    73.33 10, 14, 18, 25
  User.ts                              72.22    100      0    78.57 16, 22, 25
src/routes                             0      100      0      0
  AdminRoutes.ts                       0      100    100      0 1-18
  AuthRoutes.ts                        0      100      0      0 1-18
  GameRoutes.ts                        0      100    100      0 1-18
  GroupRoutes.ts                       0      100    100      0 1-26
  MatchmakingRoutes.ts                 0      100    100      0 1-9
  PreferencesRoutes.ts                 0      100    100      0 1-20
  ReportRoutes.ts                      0      100    100      0 1-18
  UserRoutes.ts                        0      100    100      0 1-22
src/services                           0        0      0      0
  MatchmakingService.ts                0        0      0      0 1-205

Test Suites: 2 passed, 2 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        6.269 s
```

- Log Output
- Chat Data Security
 - Verification: ...
 - Log Output

[Placeholder for chat security test logs]

4. Front-end Test Specification

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

frontend/app/src/androidTest/java/com/example/gameon/

4.2. Tests

Tests	Duration	Pixel_9_API_31
Test Results	5 m 57 s	6/7
FindGroupTest	4 m 33 s	1/2
testFindGroupSuccess	3 m 14 s	
testFindGroupFailure	1 m 18 s	
ReportTest	46 s	3/3
testReportSuccess	16 s	
testCancelReportFailure	13 s	
testReportReasonTooLargeFailure	16 s	
ViewExistingGroupTest	36 s	2/2
testViewExistingGroupSuccess	22 s	
testViewExistingGroupFailure	14 s	

• Use Case: Find Group

◦ Expected Behaviors:

Scenario Steps	Test Case Steps
1. User is on the main page of GameOn.	Log into GameOn and verify main page is opened.
2. User clicks the "Find Group" button.	Check that the button labelled Find Group is present on screen. Click the Find Group button.
3. The "Find Group" button becomes disabled and says "Finding".	Check that the Find Group button text switches to "Finding".
4. A matchmaking request is submitted, and the user is added to the matchmaking queue.	
5. A group is found based on the user's preferences.	
6. A popup appears stating "You have been matched with a group!"	Wait for up to a minute (timeout) and check that a matchmaking popup appears with the text: "You have been matched with a group!"
7. The new group appears in the "My Groups" section of the main page.	Check that a group with the name "Valorant Matchmaking Group" is present under "My Groups"

◦ Test Logs:

```
STATUS: FAILED
03-14 22:16:09.986 17778 17902 I TestRunner: started: testFindGroupSuccess(com.example.gameon.FindGroupTest)
03-14 22:16:10.109 17778 17902 I Tracing : Tracer added: class androidx.test.platform.tracing.AndroidXTracer
03-14 22:19:24.567 17778 17902 E TestRunner: failed: testFindGroupSuccess(com.example.gameon.FindGroupTest)
03-14 22:19:24.567 17778 17902 E TestRunner: at
androidx.compose.ui.test.AndroidComposeUiTestEnvironment$AndroidComposeUiTestImpl.waitForIdle(ComposeUiTest.android.k
t:441)
03-14 22:19:24.567 17778 17902 E TestRunner: at
androidx.compose.ui.test.junit4.AndroidComposeTestRule.waitForIdle(AndroidComposeTestRule.android.kt:306)
03-14 22:19:24.567 17778 17902 E TestRunner: at
androidx.test.internal.runner.junit4.statement.RunBefores.evaluate(RunBefores.java:80)
03-14 22:19:24.568 17778 17902 E TestRunner: at androidx.test.runner.AndroidJUnit4.run(AndroidJUnit4.java:123)
03-14 22:19:24.568 17778 17902 E TestRunner: at
androidx.test.internal.runner.TestExecutor.execute(TestExecutor.java:68)
03-14 22:19:24.568 17778 17902 E TestRunner: at
androidx.test.internal.runner.TestExecutor.execute(TestExecutor.java:59)
03-14 22:19:24.568 17778 17902 E TestRunner: at
androidx.test.runner.AndroidJUnitRunner.onStart(AndroidJUnitRunner.java:463)
03-14 22:19:24.571 17778 17902 I TestRunner: finished: testFindGroupSuccess(com.example.gameon.FindGroupTest)
androidx.compose.ui.test.ComposeTimeoutException: Condition still not satisfied after 120000 ms
at
androidx.compose.ui.test.AndroidComposeUiTestEnvironment$AndroidComposeUiTestImpl.waitForIdle(ComposeUiTest.android.k
t:441)
at androidx.compose.ui.test.junit4.AndroidComposeTestRule.waitForIdle(AndroidComposeTestRule.android.kt:306)
at com.example.gameon.FindGroupTest.testFindGroupSuccess(FindGroupTest.kt:120)
```

• Use Case: Find Group

◦ Expected Behaviors:

Scenario Steps	Test Case Steps
1. User is on the main page of GameOn.	Log into GameOn and verify main page is opened.
2. User clicks the "Find Group" button.	Check that the button labelled Find Group is present on screen. Click the Find Group button.
3. The "Find Group" button becomes disabled and says "Finding".	Check that the Find Group button text switches to "Finding".
4. A matchmaking request is submitted, and the user is added to the matchmaking queue.	
5. A group is not found based on the user's preferences within the timeout of 1 minute.	
6. A popup appears stating "Matchmaking timed out. Please try again."	Wait for up to a minute (timeout) and check that a matchmaking popup appears with the text: "Matchmaking timed out. Please try again."

◦ Test Logs:

```
STATUS: PASSED
03-14 22:19:25.966 17778 17902 I TestRunner: started: testFindGroupFailure(com.example.gameon.FindGroupTest)
03-14 22:20:44.757 17778 17902 I TestRunner: finished: testFindGroupFailure(com.example.gameon.FindGroupTest)
03-14 22:20:44.758 17778 17902 I TestRunner: passed: testFindGroupFailure(com.example.gameon.FindGroupTest)
```

• Use Case: Navigate to Existing Group

◦ Expected Behaviors:

Scenario Steps	Test Case Steps
1. User is on the main page of GameOn.	Log into GameOn and verify main page is opened.
2. User clicks on an existing group from the "My Groups" section.	Check that the expected test group "Valorant Matchmaking Group" is present on screen. Click the "Valorant Matchmaking Group" button.
3. The View Existing Groups page opens, displaying the group name, group members, and a "Go to Discord Group" button.	Check that the title "Valorant Matchmaking Group" is present. Check that the test group member "sanhal23" is present. Check that the "Go to Discord Group" is present.
4. User clicks the "Go to Discord Group".	Click the "Go to Discord Group" button
5. Chrome opens the Discord web version and displays the newly created group with the added members.	Check that text "discord.com/channels/" is present in the URL, to verify browser has launched.

◦ Test Logs:

```
STATUS: PASSED
03-14 22:21:37.175 17778 17902 I TestRunner: started:
testViewExistingGroupSuccess(com.example.gameon.ViewExistingGroupTest)
03-14 22:21:59.986 17778 17902 I TestRunner: finished:
testViewExistingGroupSuccess(com.example.gameon.ViewExistingGroupTest)
03-14 22:21:59.987 17778 17902 I TestRunner: passed:
testViewExistingGroupSuccess(com.example.gameon.ViewExistingGroupTest)
```

• Use Case: Navigate to Existing Group

◦ Expected Behaviors:

Scenario Steps	Test Case Steps
1. User is on the main page of GameOn.	Log into GameOn and verify main page is opened.
2. User tries to click on an existing group from "My Groups" but the list is empty.	Check that the test group "Valorant Matchmaking Group" is initially present. Clear the group list dynamically during runtime. Attempt to find the group in the UI.

Scenario Steps	Test Case Steps
3. User attempts to select the now non-existent group.	Try to click on "Valorant Matchmaking Group" after it has been removed.
4. System displays an error message instead of proceeding.	Check for the "No groups found" text on screen.
5. User cannot proceed to the View Existing Group page.	

◦ Test Logs:

```
STATUS: PASSED
03-14 22:21:59.989 17778 17902 I TestRunner: started:
testViewExistingGroupFailure(com.example.gameon.ViewExistingGroupTest)
03-14 22:22:14.192 17778 17902 I TestRunner: finished:
testViewExistingGroupFailure(com.example.gameon.ViewExistingGroupTest)
03-14 22:22:14.193 17778 17902 I TestRunner: passed:
testViewExistingGroupFailure(com.example.gameon.ViewExistingGroupTest)
```

• Use Case: Submit Report

◦ Expected Behaviors:

Scenario Steps	Test Case Steps
1. User is on the main page of GameOn.	Log into GameOn and verify main page is opened.
2. User clicks the "Submit a Report" button.	Check that the button labelled Submit a Report is present on screen. Click the Find Group button.
3. User is now on the Reports page of GameOn.	Check that the button labelled Submit Report is present on screen and is not enabled.
4. User selects a group from the Group dropdown menu.	Check that the dropdown menu labelled Group is present on screen. Click on the dropdown menu and check that at least one dropdown value exists. Click on the first dropdown option, check that the dropdown menu is closed, and check that the selected value is now displayed on the dropdown. Lastly, check that the Submit Report button is still not enabled.
5. User selects a user to report from the User dropdown menu.	Check that the dropdown menu labelled User is present on screen. Click on the dropdown menu and check that at least one dropdown value exists. Click on the first dropdown option, check that the dropdown menu is closed, and check that the selected value is now displayed on the dropdown. Lastly, check that the Submit Report button is still not enabled.
6. User types a reason for reporting the selected user in the Reason text input	Check that the text input labelled Reason is present on screen. Click on the text input and input a string of text with a length less than 500 characters. Lastly, check that the Submit Report button is now enabled.
7. User submits the report by clicking the "Submit Report" button and is taken back to the main page.	Click Submit Report button.
8. User clicks the "View Reports" button and navigates to the List Reports page of GameOn.	Check that the button labelled View Reports is present on screen and click that button.
9. User looks at the list of reports and confirms that the submitted report is present at the bottom of the list.	Check that the list of submitted reports is displayed and has at least one item. Check that the last item in this list displays the username of the user who had been reported during this sequence.

◦ Test Logs:

```
STATUS: PASSED
03-14 22:20:46.112 17778 17902 I TestRunner: started: testReportSuccess(com.example.gameon.ReportTest)
03-14 22:21:02.942 17778 17902 I TestRunner: finished: testReportSuccess(com.example.gameon.ReportTest)
03-14 22:21:02.943 17778 17902 I TestRunner: passed: testReportSuccess(com.example.gameon.ReportTest)
```

• Use Case: Submit Report

◦ Expected Behaviors:

Scenario Steps	Test Case Steps
1. User is on the main page of GameOn.	Log into GameOn and verify main page is opened.

Scenario Steps	Test Case Steps
2. User clicks the "Submit a Report" button.	Check that the button labelled Submit a Report is present on screen. Click the Find Group button.
3. User is now on the Reports page of GameOn.	Check that the button labelled Submit Report is present on screen and is not enabled.
4. User cancels the process by clicking the "Cancel" button	Check that the button labelled Cancel is displayed. Click the Cancel button.
5. User is back on the main page of GameOn.	Verify that the main page is opened.

◦ Test Logs:

```
STATUS: PASSED
03-14 22:21:04.310 17778 17902 I TestRunner: started: testCancelReportFailure(com.example.gameon.ReportTest)
03-14 22:21:18.089 17778 17902 I TestRunner: finished: testCancelReportFailure(com.example.gameon.ReportTest)
03-14 22:21:18.090 17778 17902 I TestRunner: passed: testCancelReportFailure(com.example.gameon.ReportTest)
```

• Use Case: Submit Report

◦ Expected Behaviors:

Scenario Steps	Test Case Steps
1. User is on the main page of GameOn.	Log into GameOn and verify main page is opened.
2. User clicks the "Submit a Report" button.	Check that the button labelled Submit a Report is present on screen. Click the Find Group button.
3. User is now on the Reports page of GameOn.	Check that the button labelled Submit Report is present on screen and is not enabled.
4. User selects a group from the Group dropdown menu.	Check that the dropdown menu labelled Group is present on screen. Click on the dropdown menu and check that at least on dropdown value exists. Click on the first dropdown option, check that the dropdown menu is closed, and check that the selected value is now displayed on the dropdown. Lastly, check that the Submit Report button is still not enabled.
5. User selects a user to report from the User dropdown menu.	Check that the dropdown menu labelled User is present on screen. Click on the dropdown menu and check that at least on dropdown value exists. Click on the first dropdown option, check that the dropdown menu is closed, and check that the selected value is now displayed on the dropdown. Lastly, check that the Submit Report button is still not enabled.
6. User types a reason for reporting the selected user in the Reason text input	Check that the text input labelled Reason is present on screen. Click on the text input and input a string of text with a length more than 500 characters. Lastly, check that the Submit Report button is now enabled.
7. User clicks the Submit Report button.	Click Submit Report button.
8. User can now see the Reason text input turn red and display an error message beneath it declaring the number of characters input should be less than 500.	

◦ Test Logs:

```
STATUS: PASSED
03-14 22:21:19.497 17778 17902 I TestRunner: started:
testReportReasonTooLargeFailure(com.example.gameon.ReportTest)
03-14 22:21:35.867 17778 17902 I TestRunner: finished:
testReportReasonTooLargeFailure(com.example.gameon.ReportTest)
03-14 22:21:35.867 17778 17902 I TestRunner: passed: testReportReasonTooLargeFailure(com.example.gameon.ReportTest)
```

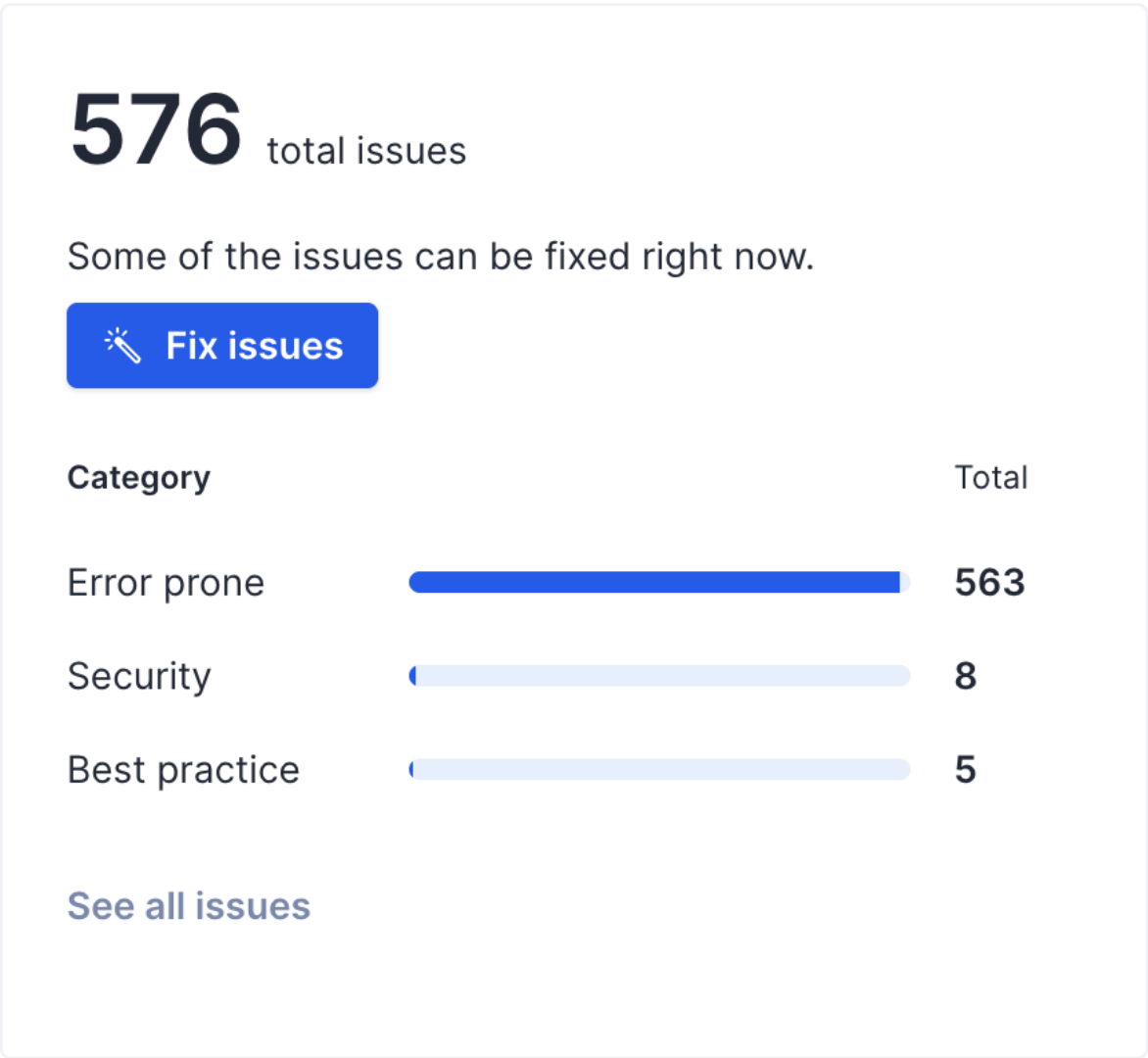
5. Automated Code Review Results

5.1. Commit Hash Where Codacy Ran

35b0261

5.2. Unfixed Issues per Codacy Category

Issues breakdown



5.3. Unfixed Issues per Codacy Code Pattern

	All issues	576
	Code patterns	
	@typescript eslint: No unsafe call	201
	@typescript eslint: No unsafe member ...	177
	@typescript eslint: No unsafe assignm...	115
	One method should have one responsibi...	18
	@typescript eslint: No unsafe return	12
	No unused vars	7
	@typescript eslint: No unused vars	7
	@typescript eslint: No unsafe argument	7
	The more parameters a function has the ...	7
	@typescript eslint: No non null assertion	5
	@typescript eslint: No misused promises	4
	Security node: Detect dangerous redirects	4
	Security: Detect object injection	4
	@typescript eslint: No explicit any	2
	The caught exception is too generic. Pref...	2
	Others	4

5.4. Justifications for Unfixed Issues

Upon integrating Codacy, over 600 issues were flagged. Due to limited time and the need to prioritize critical tasks, we focused on developing and stabilizing the testing framework to ensure core functionality. We attempted to fix some issues initially, but doing so caused delays in our testing framework development. As a result, we made a team decision not to address the remaining issues.

- ...