

Dungeons & Dragons Game State Manager



Testing Policy Document
Developed for **Retro Rabbit** by Optimize
Prime



Team:

James Hertzog
Thomas Scholtz
Jason Davidson
Ruben Denner

1. Testing Process

Definition of Testing:

Testing is done to effectively and efficiently provide timely, accurate, and useful information of the current state of the application. This is to ensure the software fulfills all its requirements while remaining in a healthy condition (bug free, etc).

Description of the test process:

Unit tests:

Goals:

- Detect defective code in units
- Reduce risk of unit failure in Production

Unit tests must comply with all coding standards.

Integration:

Goals:

- Detect defects in unit interfaces
- Reduce risk of dataflow and workflow failures in Production

After every major feature commit a test should be written to validate that the feature is working and giving the expected result. These tests must be added to automatically run on our continuous integration service (Travis CI).

Quality Level to be achieved:

Tests should result in no outstanding high severity faults prior to version releases. All server side tests must succeed for server commits to be deemed ready for merging into develop. Current running instances of the development server can only be updated if all unit tests pass.

Approach to Test Process Improvement

Tests should be reviewed alongside code reviews after a sprint. Tests should be on the same quality standard as the production code released to end-users.

2. Testing tools

We use Travis CI for automated testing of the front end and the back end. Since the front end and backend are in different GitHub repositories, testing on Travis is also done separate. Each repository has a .travis.yml file that tells travis what to install and what commands to run to execute the tests.

We chose Travis CI, because it executes tests each time new code gets pushed to GitHub or when merging occurs. This means that when developers do not run tests locally before pushing or merging, the tests will run online. Travis testing results are visible for all team members which helps the entire team know the current situation of the project.

Taken from <https://travis-ci.org/COS301-OptimizePrime/COS301-DnD> (Front end)

```
The command "./flutter/bin/flutter -v build apk" exited with 0.
$ ./flutter/bin/flutter test
00:10 +6: - Character Preview Test
Start: 400.0
End: 400.0
00:10 +10: All tests passed!

The command "./flutter/bin/flutter test" exited with 0.

Done. Your build exited with 0.
```

Taken from https://travis-ci.org/COS301-OptimizePrime/COS301_DND_Backend

```
===== 7 passed in 3.27 seconds =====
===Running dart tests!===
00:00 +0: test_create_rpc_good_login
2018-07-19 19:42:45,047 - DEBUG - session - ipv4:127.0.0.1:54740
2018-07-19 19:42:45,047 - INFO - session - Create new session called! Name:mySession
Client received: SUCCESS
00:00 +1: All tests passed!

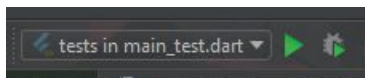
The command "make test" exited with 0.

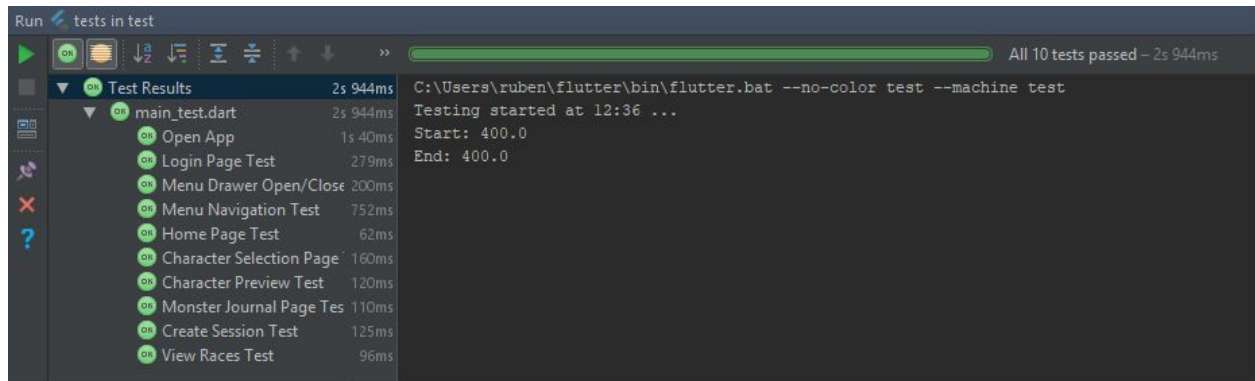
Done. Your build exited with 0.
/home/travis/.travis/job_stages: line 177: 7059 Terminated          make run
```

Front end:

The front end is developed in Android studio that provides functionality to easily run all written tests located in a folder tests. All tests are written in Dart and Android studio calls Flutter to execute the tests. We write the tests in Dart, because the source code for the project is written in Dart.

Taken from Android Studio:





Back end:

```

===Running python tests!===
===== test session starts =====
platform linux -- Python 3.6.5, pytest-3.5.1, py-1.5.3, pluggy-0.6.0
benchmark: 3.1.1 (defaults: timer=time.perf_counter disable_gc=False min_rounds=5 min_time=0.000005 max_time=1.0 calibration_precision=10 warmup=False warmup_iterations=100000)
rootdir: /home/thomas/Repos/COS301_DND_Backend/test, inifile:
plugins: benchmark-3.1.1
collected 25 items

test_sessions.py ..... [100%]

----- benchmark: 1 tests -----
-----
Name (time in s)      Min      Max      Mean  StdDev   Median     IQR  Outliers    OPS    Rounds    Itera
tions
-----
test_create_rpc_good_login  1.8277  1.9431  1.8834  0.0496   1.9023   0.0825      2;0  0.5309         5
1
-----

Legend:
  Outliers: 1 Standard Deviation from Mean; 1.5 IQR (InterQuartile Range) from 1st Quartile and 3rd Quartile.
  OPS: Operations Per Second, computed as 1 / Mean
===== 25 passed in 112.59 seconds =====
===Running dart tests!===
00:00 +0: test_create_rpc_good_login
Client received: SUCCESS
00:02 +1: All tests passed!

```

3. Test cases

Front end:

Test cases are located in the test folder of the repository.

(<https://github.com/COS301-OptimizePrime/COS301-DnD/tree/master/test>)

These tests include:

- 1) Open App:
 - a) Opens the Application to see if it runs.
- 2) Login Page Test

- a) Opens the Login Page to see if it loads correctly.
- 3) Menu Drawer Open/Close Test:
 - a) Tests if the menu drawer opens and closes correctly.
- 4) Menu Navigation Test:
 - a) Test the menu navigation functionality by tapping on all the icons in the drawer and testing if the app correctly navigates to all pages.
- 5) Home Page Test:
 - a) Tests if the Home Page loads correctly
- 6) Character Selection Page Test:
 - a) Tests if the Character Selection Page loads correctly.
 - b) Tests tap on Character functionality that shows more details about the Character.
- 7) Character Details Page Test:
 - a) Tests if the Character Details Page loads correctly.
 - b) Checks if character information shown is correct.
- 8) Character Creation Page Test:
 - a) Tests navigation to character creation.
 - b) Tests if page loads correctly.
 - c) Creates a character and tests functions of page.
- 9) Character Preview Test:
 - a) Tests if a preview of the Character pops up when a swipe is made on a Character and if preview is correct.
- 10) Monster Journal Page Test:
 - a) Tests if the Monster Journal Page loads correctly.
 - b) Tests tap on Monster functionality that shows more details about the Monster.
- 11) Create Session Test:
 - a) Tests functionality when a user taps on the "Create Session" button.
- 12) View Races Page:
 - a) Tests if the Monster Journal Page loads correctly.
 - b) Tests tap on Monster functionality that shows more details about the Monster.

4. History

The Travis builds for front end test can be found at

<https://travis-ci.org/COS301-OptimizePrime/COS301-DnD/builds>.

✗ in_session	fixed characters not saving	✗ #112 failed	🕒 4 min 1 sec
🔌 rdenner		🔗 78de4d6	📅 a day ago
✗ character_creat	Updated character conversions.	✗ #111 failed	🕒 3 min 34 sec
🔌 JasonDav		🔗 77aef43	📅 a day ago
✓ in_session	minor fix	✓ #110 passed	🕒 3 min 27 sec
🔌 rdenner		🔗 0e5a1e5	📅 a day ago
✓ in_session	characters now appear in party	✓ #109 passed	🕒 3 min 39 sec
🔌 rdenner		🔗 9e3ec8e	📅 a day ago
✓ in_session	characters now appear in sessions	✓ #108 passed	🕒 4 min 7 sec
🔌 rdenner		🔗 f2701b5	📅 2 days ago
✓ in_session	fixed widget tests	✓ #107 passed	🕒 3 min 56 sec
🔌 rdenner		🔗 b6506a4	📅 2 days ago
✗ Monster_Journ	Added search bar to new monster journal	✗ #106 failed	🕒 3 min 36 sec
🔌 James		🔗 6ca3dfb	📅 2 days ago
✗ Monster_Journ	Added 428 monsters to new journal	✗ #105 failed	🕒 4 min 16 sec

For backend Travis builds go to

https://travis-ci.org/COS301-OptimizePrime/COS301_DND_Backend/builds.

✓ develop	Updated all exceptions to print sql error to l	🔄 #119 passed	🕒 3 min 28 sec
👤 Thomas Scholtz		🔗 a9f13db ↗	📅 2 days ago
✓ develop	Removed lock file.	🔄 #118 passed	🕒 3 min 40 sec
👤 Thomas Scholtz		🔗 559fd44 ↗	📅 2 days ago
✓ master	Merge pull request #42 from COS301-Optimi	🔄 #117 passed	🕒 3 min 44 sec
👤 Thomas Scholtz		🔗 830e41a ↗	📅 2 days ago
✓ develop	Added gender and level attributes to charac	🔄 #115 passed	🕒 5 min 27 sec
👤 Thomas Scholtz		🔗 9c896a9 ↗	📅 2 days ago
✓ master	Merge branch 'master' of github.com:COS30	🔄 #114 passed	🕒 4 min 3 sec
👤 Thomas Scholtz		🔗 eb5b377 ↗	📅 3 days ago
✓ master	Merge pull request #40 from COS301-Optimi	🔄 #113 passed	🕒 4 min 32 sec
👤 Thomas Scholtz		🔗 f77c7ab ↗	📅 3 days ago
✓ develop	Ready functionality should be working and s	🔄 #111 passed	🕒 3 min 36 sec
👤 Thomas Scholtz		🔗 2eeb771 ↗	📅 3 days ago