

Anna DeBrunner
Chuck Greenman
Isaac Mathers

Test Plan

Overall Test Plan

Since our project will involve several components working together to create a solution, we will first test these components individually and then test their integration. The method of testing for each individual component will depend on the technology used to implement that component. One resource we will use in this progress is PyTest to write unit tests that will encapsulate as many test cases as possible and perform them automatically. Once each component is passing its tests, we will perform integration tests to ensure the pieces are working together as expected. This will mainly be done manually through the user interface by ensuring that operations requiring each component are being completed properly.

Test Case Descriptions

Reddit Live Test Cases

1. **Reddit Live Test 1**

- RL1.1. RL1
- RL1.2. Purpose: This test will ensure that the Reddit Live API can properly obtain access to information about a submitted Reddit user.
- RL1.3. Description: This test will attempt to create a User object for a Reddit username that is known to exist and be valid. Upon successful creation of the User object, it will run the methods of the User class and ensure that no exceptions are encountered when trying to obtain certain attributes or information about the user.
- RL1.4. Inputs: Username of a Reddit user that is valid and known to exist
- RL1.5. Expected Results: The method used to check if a username is valid will return True and all other methods used to obtain comments, posting activity, etc. will return the requested information without encountering any exceptions (with the

exception of downvote and upvote methods which may return an exception indicating that access to this information is Forbidden, which is expected).

- RL1.6. Normal
- RL1.7. Whitebox
- RL1.8. Functional
- RL1.9. Unit

2. Reddit Live Test 2

- RL2.1. RL2
- RL2.2. Purpose: This test will ensure that the Reddit Live API will throw the proper exceptions when a user is submitted that is invalid or does not exist.
- RL2.3. Description: This test will attempt to create a User object for a Reddit username that is known not to exist. Upon successful creation of the User object, it will run the methods of the User class and ensure that the expected exceptions are thrown when trying to obtain certain attributes or information about the user.
- RL2.4. Inputs: Username of a Reddit user that is invalid
- RL2.5. Expected Results: The method used to check if a username is valid will return False and all other methods used to obtain comments, posting activity, etc. will throw a NotFound exception, indicating that information about this user does not exist.
- RL2.6. Abnormal
- RL2.7. Whitebox
- RL2.8. Functional
- RL2.9. Unit

3. Reddit Live Test 3

- RL3.1. RL3
- RL3.2. Purpose: This test will ensure that the Reddit Live API can properly obtain access to information about a submitted subreddit.
- RL3.3. Description: This test will attempt to create a Subreddit object for a subreddit that is known to exist and is not quarantined. Upon successful creation of the Subreddit object, it will run the methods of the Subreddit class and ensure that no exceptions are encountered when trying to obtain certain attributes or information about the subreddit.
- RL3.4. Inputs: Name of a subreddit that is known to exist

- RL3.5. Expected Results: The method used to check if a subreddit name is valid will return True and the method used to check if a subreddit is quarantined will return False. All other methods used to obtain comments, posting activity, etc. will return the requested information without encountering any exceptions.
- RL3.6. Normal
- RL3.7. Whitebox
- RL3.8. Functional
- RL3.9. Unit

4. Reddit Live Test 4

- RL4.1. RL4
- RL4.2. Purpose: This test will ensure that the Reddit Live API can properly identify a quarantined subreddit and throw the proper exceptions when information from it is requested.
- RL4.3. Description: This test will attempt to create a Subreddit object for a subreddit that is quarantined. Upon successful creation of the Subreddit object, it will run the methods of the Subreddit class and ensure that the proper exceptions are thrown when trying to obtain certain attributes or information about the subreddit.
- RL4.4. Inputs: Name of a subreddit that is known to be quarantined
- RL4.5. Expected Results: The method used to check if a subreddit name is valid will return True and the method used to check if a subreddit is quarantined will return True. All other methods used to obtain comments, posting activity, etc. will throw a Forbidden exception, indicating that access to this information is denied.
- RL4.6. Abnormal
- RL4.7. Whitebox
- RL4.8. Functional
- RL4.9. Unit

5. Reddit Live Test 5

- RL5.1. RL5
- RL5.2. Purpose: This test will ensure that the Reddit Live API will throw the proper exceptions when a subreddit is submitted that is invalid or does not exist.
- RL5.3. Description: This test will attempt to create a Subreddit object for a subreddit that is known not to exist. Upon successful creation of the Subreddit object, it will run

the methods of the Subreddit class and ensure that the expected exceptions are thrown when trying to obtain certain attributes or information about the subreddit.

RL5.4. Inputs: Name of a Subreddit that is known not to exist

RL5.5. Expected Results: The method used to check if a subreddit is valid will return False and all other methods used to obtain comments, posting activity, etc. will throw a NotFound exception, indicating that information about this subreddit does not exist.

RL5.6. Abnormal

RL5.7. Whitebox

RL5.8. Functional

RL5.9. Unit

Database Test Cases

1. Database Test 1

DB1.1. DB1

DB1.2. Purpose: This test will ensure that the database properly returns user data when requested.

DB1.3. Description: This test will query user data from the database. The test will begin with an existing database. A username will be passed to the database and all data related to that user will be packaged and sent back.

DB1.4. Inputs: Reddit username that is known to be present in the database

DB1.5. Expected Results: All information related to that user currently in the database

DB1.6. Normal

DB1.7. Blackbox

DB1.8. Functional

DB1.9. Unit

2. Database Test 2

DB2.1. DB2

DB2.2. Purpose: This test will ensure that the database properly returns subreddit data when requested.

DB2.3. Description: This test will query subreddit data from the database. This test will begin with an existing database. A subreddit name and various specifications

(how posts should be sorted, for example) will be passed to the database and the requested data will be packaged and sent back.

- DB2.4. Inputs: Subreddit that is known to be in the database and type of data from subreddit requested
- DB2.5. Expected Results: The requested information will be packaged and outputted.
- DB2.6. Normal
- DB2.7. Blackbox
- DB2.8. Functional
- DB2.9. Unit

3. Database Test 3

- DB3.1. DB3
- DB3.2. Purpose: This test will ensure that the database properly returns submission data when requested.
- DB3.3. Description: This test will attempt to query submission data from the database. This test will begin with an existing database. A submission id will be passed to the database and the requested data will be packaged and sent back.
- DB3.4. Inputs: Submission id that is known to be in the database
- DB3.5. Expected Results: The requested information will be packaged and outputted.
- DB3.6. Normal
- DB3.7. Blackbox
- DB3.8. Functional
- DB3.9. Unit

Integration Tests

1. Integration Test 1

- IT1.1. IT1
- IT1.2. Purpose: This test ensures that the user interface has proper communication with the Python Flask application that facilitates requests to the Reddit Live API.
- IT1.3. Description: This test will begin with an existing, valid subreddit being input to the user interface for analysis. This analysis requires real time data to be requested from the Reddit Live API via the Python Flask application. The returned information will be displayed in the user interface, either in its raw form or after some analysis has been applied.

- IT1.4. Inputs: Valid subreddit to request realtime information from
- IT1.5. Expected Results: Requested information is returned and appears in user interface with no errors or exceptions encountered
- IT1.6. Normal
- IT1.7. Blackbox
- IT1.8. Functional
- IT1.9. Integration

2. Integration Test 2

- IT2.1. IT2
- IT2.2. Purpose: This test ensures that the database properly accesses and adds data to itself when given updated user information for it to store.
- IT2.3. Description: The test will begin with an existing database with user data. The Reddit Live API will send the database the most up to date information on a given user. The database needs to make sure it adds all the data properly without overwriting previous data. If the Reddit user is not previously in the database, it should be added.
- IT2.4. Inputs: Reddit user data
- IT2.5. Expected Results: Any new information that is not already in the database about the user will be added.
- IT2.6. Normal
- IT2.7. Blackbox
- IT2.8. Functional
- IT2.9. Integration

Overall Test Case Matrix

Test Case ID	(Ab)Normal	Black/Whitebox	Funcct/Perform	Unit/Integr.
RL1	Normal	Whitebox	Functional	Unit
RL2	Abnormal	Whitebox	Functional	Unit
RL3	Normal	Whitebox	Functional	Unit
RL4	Abnormal	Whitebox	Functional	Unit
RL5	Abnormal	Whitebox	Functional	Unit
DB1	Normal	Blackbox	Functional	Unit
DB2	Normal	Blackbox	Functional	Unit
DB3	Normal	Blackbox	Functional	Unit
IT1	Normal	Blackbox	Functional	Integration
IT2	Normal	Blackbox	Functional	Integration