

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. 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.
- 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 to not 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

Flask Service Tests

1. Flask Test 1

F1.1. F1

F1.2. Purpose: This test will ensure that the Flask service can properly fulfill requests for information about a requested valid user.

F1.3. Description: This test will make an HTTP request to the Flask service to obtain basic information about a valid user. The Flask service should return basic information about the user.

F1.4. Inputs: Valid HTTP request to get information about a user

F1.5. Expected Results: The Flask service will return a JSON object with the information about the requested user.

F1.6. Normal

F1.7. Whitebox

F1.8. Functional

F1.9. Unit

2. Flask Test 2

F2.1. F2

F2.2. Purpose: This test will ensure that the Flask service can properly fulfill requests for information about a requested invalid user.

F2.3. Description: This test will make an HTTP request to the Flask service to obtain basic information about an invalid user. The Flask service should return a message indicating that the user is invalid.

F2.4. Inputs: Valid HTTP request to get information about an invalid user

F2.5. Expected Results: The Flask service will return a JSON object with a field stating that the user is invalid.

F2.6. Abnormal

- F2.7. Whitebox
- F2.8. Functional
- F2.9. Unit

3. Flask Test 3

- F3.1. F3
- F3.2. Purpose: This test will ensure that the Flask service can properly fulfill requests for information about a requested valid subreddit.
- F3.3. Description: This test will make an HTTP request to the Flask service to obtain basic information about a valid subreddit. The Flask service should return basic information about the subreddit.
- F3.4. Inputs: Valid HTTP request to get information about a subreddit
- F3.5. Expected Results: The Flask service will return a JSON object with the information about the requested subreddit.
- F3.6. Normal
- F3.7. Whitebox
- F3.8. Functional
- F3.9. Unit

4. Flask Test 4

- F4.1. F4
- F4.2. Purpose: This test will ensure that the Flask service can properly fulfill requests for information about a requested quarantined subreddit.
- F4.3. Description: This test will make an HTTP request to the Flask service to obtain basic information about a quarantined subreddit. The Flask service should return a message indicating that the subreddit is valid but quarantined.
- F4.4. Inputs: Valid HTTP request to get information about a quarantined subreddit
- F4.5. Expected Results: The Flask service will return a JSON object with fields indicating the subreddit is valid but quarantined
- F4.6. Abnormal
- F4.7. Whitebox
- F4.8. Functional
- F4.9. Unit

5. Flask Test 5

- F5.1. F5

- F5.2. Purpose: This test will ensure that the Flask service can properly fulfill requests for information about a requested invalid subreddit.
- F5.3. Description: This test will make an HTTP request to the Flask service to obtain basic information about an invalid subreddit. The Flask service should return a message indicating that the subreddit is invalid.
- F5.4. Inputs: Valid HTTP request to get information about an invalid subreddit
- F5.5. Expected Results: The Flask service will return a JSON object with a field indicating the subreddit is invalid.
- F5.6. Abnormal
- F5.7. Whitebox
- F5.8. Functional
- F5.9. 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 user interface has proper communication with the Python Flask application that facilitates requests to the Reddit Live API.
- IT2.3. Description: This test will begin with an existing, valid user 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.
- IT2.4. Inputs: Valid user to request realtime information from
- IT2.5. Expected Results: Requested information is returned and appears in user interface with no errors or exceptions encountered
- IT2.6. Normal
- IT2.7. Blackbox
- IT2.8. Functional
- IT2.9. Integration

3. Integration Test 3

- IT3.1. IT3
- IT3.2. Purpose: This test ensures that the user interface properly handles the case in which the response from the Flask application indicates that the user is invalid.
- IT3.3. Description: This test will begin with an invalid user 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 indicate that the user is invalid and the OSCAR user will be notified of this.
- IT3.4. Inputs: Invalid user to request realtime information from
- IT3.5. Expected Results: Response indicates that the user is invalid and a popup appears notifying the OSCAR user that analysis cannot be completed for the submitted user for this reason
- IT3.6. Abnormal
- IT3.7. Blackbox
- IT3.8. Functional
- IT3.9. Integration

4. Integration Test 4

- IT4.1. IT4

- IT4.2. Purpose: This test ensures that the user interface properly handles the case in which the response from the Flask application indicates that the subreddit is invalid.
- IT4.3. Description: This test will begin with an invalid 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 indicate that the subreddit is invalid and the OSCAR user will be notified of this.
- IT4.4. Inputs: Invalid subreddit to request realtime information from
- IT4.5. Expected Results: Response indicates that the subreddit is invalid and a popup appears notifying the OSCAR user that analysis cannot be completed for the submitted subreddit for this reason
- IT4.6. Abnormal
- IT4.7. Blackbox
- IT4.8. Functional
- IT4.9. Integration

Overall Test Case Matrix

| Test Case ID | (Ab)Normal | Black/Whitebox | Funct/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 |
| F1 | Normal | Whitebox | Functional | Unit |
| F2 | Abnormal | Whitebox | Functional | Unit |
| F3 | Normal | Whitebox | Functional | Unit |
| F4 | Abnormal | Whitebox | Functional | Unit |
| F5 | Abnormal | Whitebox | Functional | Unit |

| | | | | |
|-----|----------|----------|------------|-------------|
| IT1 | Normal | Blackbox | Functional | Integration |
| IT2 | Normal | Blackbox | Functional | Integration |
| IT3 | Abnormal | Blackbox | Functional | Integration |
| IT4 | Abnormal | Blackbox | Functional | Integration |

Results

1. Reddit Live Test 1

Input: Username = HawkSoul12

Output: Method checking validity returns true. All other methods run successfully with no exceptions.

2. Reddit Live Test 2

Input: Username = fahklfwq

Output: Method checking validity returns false. All other methods encounter the `prawcore.exceptions.NotFound` exception.

3. Reddit Live Test 3

Input: Subreddit = worldNews

Output: Method checking validity returns true. Method checking if quarantined returns false. All methods run successfully with no exceptions.

4. Reddit Live Test 4

Input: Subreddit = The_Donald

Output: Method checking validity returns true. Method checking if quarantined returns true. All other methods encounter the `prawcore.exceptions.Forbidden` exception.

5. Reddit Live Test 5

Input: Subreddit = fhajlfea

Output: Method checking validity returns false. All other methods encounter the `prawcore.exceptions.NotFound` exception.

6. Flask Test 1

Input: `http://127.0.0.1:5000/user?name=HawkSoul12`

Output: `{"is_valid": true, "comment_karma": 442, "link_karma": 2, "top_subreddits": {"AskReddit": 7, "buildapc": 5}, "popular_words": {"build": 13, "rebates": 9, "outletpc": 8, "atx": 8, "newspaper": 7, "would": 7, "price": 7, "newegg": 7, "pihole": 6, "like": 6, "parts": 6, "list": 6, "motherboard": 6, "memory": 6, "video": 6, "card": 6, "case": 6, "power": 6, "supply": 6, "total": 6,`

"mailin": 6, "also": 6, "gaming": 5, "ads": 5, "general": 5, "read": 5, "storage": 5, "drive": 5, "games": 5}, "most_linked_websites": {}, "proportion_controversial_posts": {"controversial": 1, "non-controversial": 2}, "evaluation": null}

7. Flask Test 2

Input: http://127.0.0.1:5000/user?name=fahklfwq

Output: {"is_valid": false}

8. Flask Test 3

Input: http://127.0.0.1:5000/subreddit?name=worldNews

Output: {"name": "worldNews", "description": " SC_OFF \n\n\n\nOther

Subs:\nRelated\n\n/r/News\n/r/PoliticalDiscussion\n/r/WorldEvents\n/r/GeoPolitics\n/r/Gl
obalTalk

\n/r/Business\n/r/Economics\n/r/Environment\n/r/History\n/r/HumanRights\n/r/Features\n/
r/UpliftingNews\n/r/NewsOfTheWeird\n/r/FakeNews\n/r/ID_News \n\nN.

America\n\n/r/Politics\n/r/USA\n/r/USANews\n/r/Canada\n/r/CanadaPolitics\n/r/OnGuard
ForThee\n/r/Cuba\n/r/Mexico\n/r/PuertoRico\n\nS.

America\n\n/r/Argentina\n/r/Brasil\n/r/Chile\n/r/Colombia\n/r/Ecuador\n/r/Guyana\n/r/Nica
ragua\n/r/Uruguay\n/r/Venezuela\n\nEurope\n\n/r/Armenia\n/r/Azerbaijan\n/r/Belgium\n/r/
Bosnia\n/r/Bulgaria\n/r/Croatia\n/r/Denmark\n/r/Europe\n/r/France\n/r/Georgia\n/r/Germa
ny\n/r/Greece\n/r/Hungary\n/r/Ireland\n/r/Italy\n/r/Moldova\n/r/TheNetherlands\n/r/Poland
\n/r/Polaska\n/r/Portugal\n/r/Romania\n/r/Russia\n/r/Scotland\n/r/Serbia\n/r/Spain\n/r/Swe
den\n/r/Switzerland\n/r/Turkey\n/r/UnitedKingdom\n/r/UKPolitics\n/r/Ukraina\n/r/Ukraine\
n/r/UkrainianConflict\n\nAsia\n\n/r/Afghanistan\n/r/Bangladesh\n/r/China\n/r/India\n/r/Kaz
akhstan

\n/r/Malaysia\n/r/NorthKoreaNews\n/r/Pakistan\n/r/Philippines\n/r/Singapore\n/r/Thailand\
n/r/Turkey\n\nMiddle

East\n\n/r/Assyria\n/r/Iran\n/r/Iranian\n/r/Iraq\n/r/Israel\n/r/Kurdistan\n/r/LevantineWar\n/r/
MiddleEastNews\n/r/MideastPeace\n/r/Palestine\n/r/Syria\n/r/Yemen\n/r/YemeniCrisis\n\
nAfrica\n\n/r/Africa\n/r/SouthAfrica\n\nOceania\n\n/r/Australia\n/r/Fijian\n/r/NewZealand\
r/Oceania\n/r/Westpapua\n\n\n\n\nFilter out dominant topics:\nDisplay COVID-19

submissions\nFilter COVID-19\nDisplay Trump submissions\nFilter Trump\nDisplay
Russia submissions\nFilter Russia\nDisplay North Korea submissions\nFilter North
Korea\nDisplay Hong Kong submissions\nFilter Hong Kong\nDisplay Israel/Palestine
submissions\nFilter Israel / Palestine\nDisplay all submissions\nFilter all dominant

topics\n Live Thread: COVID-19 Outbreak\n\n\nWelcome!\n/r/worldnews is for major news from around the world except US-internal news / US politics\nSee all of our AMA events here\n\n\n\nWorldnews Rules\nDisallowed submissions\n\nUS internal news/US politics\nEditorialized titles\nMisleading titles\nEditorials, opinion, analysis\nFeature stories\nNon-English articles\nImages, videos or audio clips\nPetitions, advocacy, surveys\nAll caps words in titles\nBlogspam (if stolen content/direct copy)\nTwitter, Facebook, Tumblr\nOld news (\u22651 week old) articles\n\nSee the wiki for details on each rule\nDisallowed comments\n\nBigotry / Other offensive content\nPersonal attacks on other users\nMemes/GIFs\nUnlabeled NSFW images/videos\nURL shorteners\n\nSee the wiki for details on each rule\nContinued or outstandingly blatant violation of the submission or commenting rules will result in you being temporarily banned from the subreddit without a warning.\n\nPlease don't ever feed the trolls.\nDownvote, report and move on.\n\n\nWhat moderators do and can't do\nMessage the moderators\n\n\n\nSticky Posts\n\n\u2022 A list of all recent stickied posts.\n\n SC_ON ", "sub_count": 23794555, "is_valid": true, "is_quarantined": false, "popular_words": {"coronavirus": 38, "covid19": 19, "world": 16, "trump": 11, "says": 10, "health": 10, "new": 10, "us": 9, "china": 9, "people": 8, "cases": 8, "million": 7, "pandemic": 6, "first": 6, "time": 6, "lockdown": 5, "response": 5, "workers": 5, "say": 5, "patients": 5, "brazil": 5, "minister": 5, "day": 5, "uk": 5}, "most_linked_websites": {"reuters.com": 7, "independent.co.uk": 5, "businessinsider.com": 5, "theguardian.com": 4, "news.sky.com": 3, "forbes.com": 3, "timesofisrael.com": 3, "vice.com": 2, "newsweek.com": 2, "hongkongfp.com": 2, "cnn.com": 2, "edition.cnn.com": 2, "abc.net.au": 2, "cnbc.com": 2, "scmp.com": 2, "cbc.ca": 2, "dw.com": 2, "taiwannews.com.tw": 2, "firstpostofindia.com": 2, "france24.com": 2, "wjla.com": 1, "thelondoneconomic.com": 1, "npr.org": 1, "extra.ie": 1, "videogamer.com": 1, "climatechangenews.com": 1, "nationalpost.com": 1, "thehill.com": 1, "news.com.au": 1, "e.vnexpress.net": 1, "time24.news": 1, "usatoday.com": 1, "apnews.com": 1, "techcrunch.com": 1, "thesun.co.uk": 1, "sg.news.yahoo.com": 1, "euractiv.com": 1, "cbsnews.com": 1, "rnz.co.nz": 1, "aljazeera.com": 1, "standard.co.uk": 1, "thescottishsun.co.uk": 1, "9news.com.au": 1, "nytimes.com": 1, "express.co.uk": 1, "japantimes.co.jp": 1, "economictimes.indiatimes.com": 1, "nypost.com": 1, "uk.reuters.com": 1, "globalnews.ca": 1, "stuff.co.nz": 1, "amp.theguardian.com": 1, "dailymail.co.uk": 1, "euobserver.com": 1, "straitstimes.com": 1, "9now.nine.com.au": 1,

"bbc.co.uk": 1, "arabnews.com": 1, "news.mongabay.com": 1, "newsinfo.inquirer.net": 1, "theglobeandmail.com": 1, "politico.eu": 1, "riotimesonline.com": 1, "buzzfeednews.com": 1}, "proportion_controversial_posts": {"controversial": 0, "non-controversial": 100}}

9. Flask Test 4

Input: http://127.0.0.1:5000/subreddit?name=The_Donald

Output: {"name": "The_Donald", "is_valid": true, "is_quarantined": true}

10. Flask Test 5

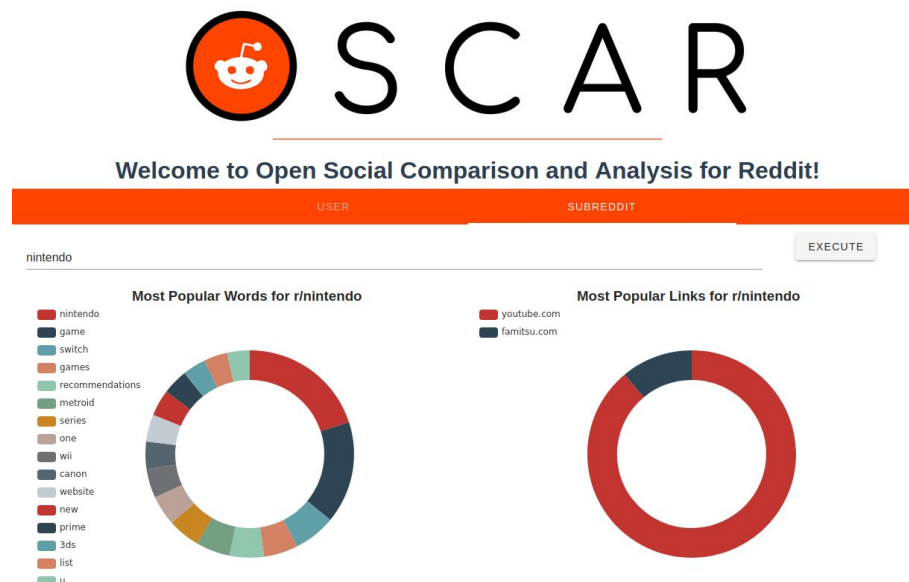
Input: http://127.0.0.1:5000/subreddit?name=fhajlfea

Output: {"name": "fhajlfea", "is_valid": false}

11. Integration Test 1

Input: Subreddit = nintendo

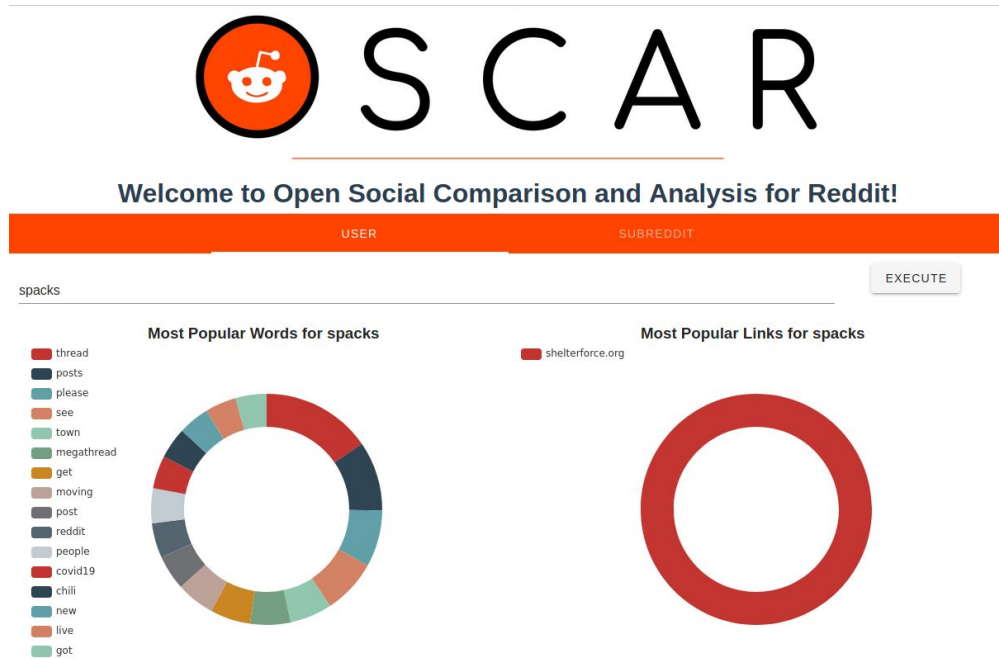
Output:



12. Integration Test 2

Input: User = spacks

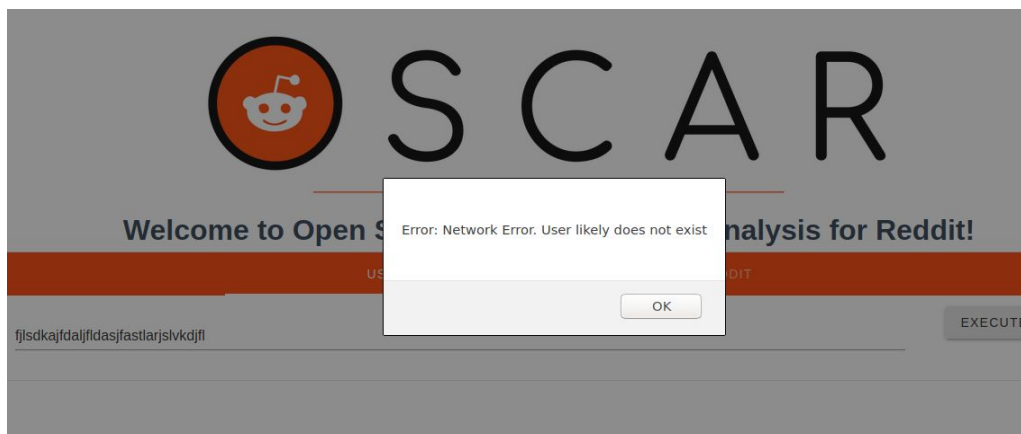
Output:



13. Integration Test 3

Input: User = random string of letters

Output:



14. Integration Test 4

Input: Subreddit = random string of letters

Output:



SCAR

Welcome to C

Error: Network Error. Subreddit likely does not exist or is quarantined.

s for Reddit!

OK

sadfdajsfjlsdalfjdfasdalfkjlf

EXECUTE