Test Report

Created	@October 21, 2021 6:33 PM
Last Edited Time	@January 16, 2022 11:46 PM
Type	Technical Spec
Status	
Created By	
Last Edited By	
Stakeholders	

1. Introduction

This document contains information about test strategy, test subjects, and test results of tests that we performed. This document has 3 main parts: test plan, additional tests, and test results.

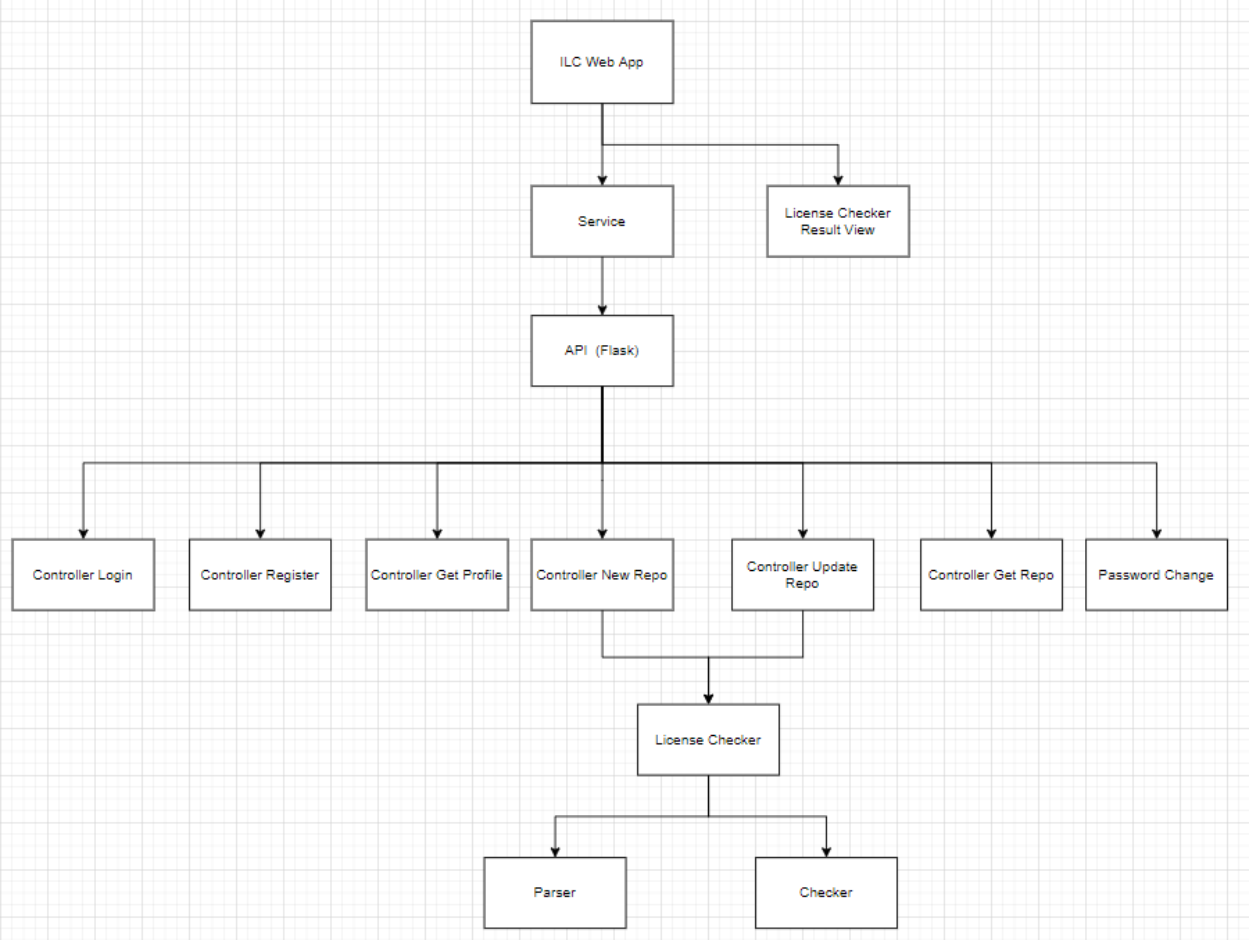
In the test plan, you will understand why we use the bottom-up approach integration test and which data is subject to integration testing. The additional test part includes which external test we needed or why we didn't. By looking at the test results, it can be understood how well the parts we implement work independently in other parts, the compatibility of these parts with each other by looking at the integration test results, and whether it will be successful if more than one user sends hundreds of requests to the system at the same time with the load test.

2. Test Plan

2.1 Testing Strategy

In our project, we use Bottom-up testing strategies. One of the reasons we employ the bottom-up testing strategy is that it allows us to solve minor problems first, then merge them all together to create a full solution. We were able to execute our tests in parallel with our development because we employed a similar technique while designing the project.

Another reason we employ the bottom-up testing strategy is that when dealing with serious defects at the end of the program, the Bottom-Up Integration testing approach comes in handy. ILC Licensing Checker is a program that checks for license dependency issues using the GitHub Repo Url. However, we made it a web application. In other words, while being the most important aspect of the project, the ILC tool is located at the bottom of the ILC WebApp Project.



2.2 Test Subjects

Integration Test License Checker:

Input: Url of the repository and a boolean for the inclusion of development dependency

Output: License dependencies tree in JSON Format and License Inconsistencies Array

Integration Test Controller New Repo:

Input: Url of the repository, a boolean for the inclusion of development dependency, user credentials

Output: Successfully save new repository on the database.

Integration Test Controller Update Repo:

Input: Url of repo that comes from Request

Output: Success message.

Integration Test API (FLASK) :

Login:

Input: Mail and password

Output: Login success or invalid mail/password message

If login is successful, access to License Checker and profile.

Register:

Input: Mail and password

Output: Successful registration or unsuccessful message.

Saving user credentials to the database.

New repo:

Input: Request with project URL and user credentials

Output: Dependency graph result and saving to database.

Integration Test ILC WebApp :

Each use case should be tested.

Input: Request from UI

Output: A bunch of operation

2.3 Black Box Testing

2.4 White Box Testing

Parser

Test Number	Description	Excepted Result
1	Valid JS repo without including development dependencies	Valid JSON that shows all dependencies

2	JS + Python repo w/o dev. dep.	Valid JSON that finds licenses and dependencies if possible
3	JS + Python repo with dev. dep	Valid JSON that finds licenses and dependencies if possible
4	Repo with no license information	Valid JSON including error tags

Checker

Test Number	Description	Excepted Result
1	JSON that has no Error value as input	Valid JSON that shows dependency graph and Valid String array that contains incompatible packages.
2	JSON that has some Error value as input	Valid JSON that shows dependency graph and Valid String array that contains incompatible packages.

Backend API Request Testing

Register

Test Number	Description	Excepted Response Status Code	Test result
1	Missing one or both of the email or password parameters.	400, Email and Password is required, not register	Passed the test.
2	When trying to register with a previously registered email and valid password parameter.	400, Not register.	Passed the test.
3	New register, with unused email and valid password	201, successful registration.	Passed the test.

Login

Test Number	Description	Excepted Response Status Code	Test result
1	Request with correct email and password.	200, logged-in.	Passed the test.
2	Request with the wrong email.	404, No logged-in, user not found.	Passed the test.
3	Request with the wrong password.	404, No logged-in.	Passed the test.

4	Request with the wrong email and password.	404, No logged-in.	Passed the test.
5	Missing email parameter.	400, No logged-in, email is required	Passed the test.
6	Missing password parameter.	400, No logged-in, password is required	Passed the test.

Profile

Test Number	Description	Excepted Response Status Code	Test result
1	Get request profile with true token parameter.	200, returns profile details.	Passed the test.
2	Get request profile with false token parameter.	401, Token is required.	Passed the test.
3	Missing token parameter.	401, Missing Authorization Header.	Passed the test.

Repo

Test Number	Description	Excepted Response Status Code	Test result
1	Get request repo with true token parameter.	200, returns repo details.	Passed the test.
2	Get request repo with false token parameter.	401, Token is required.	Passed the test.
3	Post request repo with true token and wrong URL(random string).	400, correct url must be entered.	Test Failed. URL accuracy is assumed to be checked on the frontend.
4	Post request repo with true token and wrong URL(URL but not Github URL).	400, correct url must be entered.	Test Failed. URL accuracy is assumed to be checked on the frontend.
5	Post request with missing URL parameter.	400, URL is required.	Passed the test.

Old Repo

Test	Description	Excepted	Test result

Number		Response Status Code	
1	Get request oldrepo with true token and true repoId(But false URL posted before) parameters.	200, OldRepo inconsistency not found.	Test Failed. The repo that was previously added with the wrong URL should not return.
2	Get request oldrepo with true token and repoId parameters.	200, Return oldrepo info.	Passed the test.
3	Post request oldrepo with true token and repoId parameters.	200, Update oldrepo info.	Passed the test.

Change Password

Test Number	Description	Excepted Response Status Code	Test result
1	Post request change password with true token and true old password parameters.	200, Succesfully change password.	Passed the test.
2	Post request change password with true token and wrong old password parameters.	404, Old password is wrong.	Passed the test.
3	Post request change password with missing one or both of the email or password parameters.	400, Old password, and new password are required.	Passed the test.

Web-app Testing

Login - Signup

Test Number	Description	Excepted Response Status Code	Test Result
1	Missing one or both of the email or password parameters.	Alert message displays as “You should enter valid email and password”	Passed the test
2	Entering password less than 8 character	Alert message displays as “Password should be at least 8 character”	Passed the test
3	Entering invalid email	Alert message displays as “Enter the valid email“	Passed the test
4	Entering valid email and password for signup	Alert message displays as “Signup is successful”	Passed the test

5	Entering wrong email or password for login	Alert message displays as “Email or password is wrong”	Passed the test
6	Entering correct email and password for login	Alert message displays as “Login succeed”	Passed the test

License Checker

Test Number	Description	Excepted Response Status Code	Test Result
1	Missing repository url parameters	Alert message displays as ”Enter valid url”	Passed the test
2	Missing other url instead of url parameters	The tree is not showed	Passed the test
3	Entering valid url of repository	Alert message displays as “Url is added” and the tree of license is showed	Passed the test

Change Password

Test Number	Description	Excepted Response Status Code	Test Result
1	Missing passwords	Alert message displays as ”Enter passwords”	Passed the test
2	Entering wrong old password	Alert message displays as “Wrong password”	Passed the test
3	Mismatching new password and confirm new password	Alert message displays as “Passwords are not matching”	Passed the test
4	Entering correct old password and matching new passwords	Alert message displays as “Password is changed” and login page is opened	Passed the test

Old Repositories

Test Number	Description	Excepted Response Status Code	Test Result
1	Clicking invalid url of repository	The tree is not showed	Passed the test
2	Clicking valid url of repository	The tree of license is showed	Passed the test
3	Missing url parameters	Alert message displays as “Invalid url”	Passed

	for update		the test
4	Entering same url of respositories for updating	Alert message displays as “The url is updated” and the tree of license of new project is showed	Passed the test

3. Additional Tests

Security is not a concern for ILC License checker. It is a tool that helps Software Engineers. It doesn't contain any essential information. For this reason, we did not perform any security test on ILC License Checker.
The speed of ILC depends on the communication between GitHub and web app. In fact, it is proportional to how fast Github returns us information that determines performance. Since the license incompatibility testing process is a situation that does not require immediate resolution, we did not perform any performance testing.

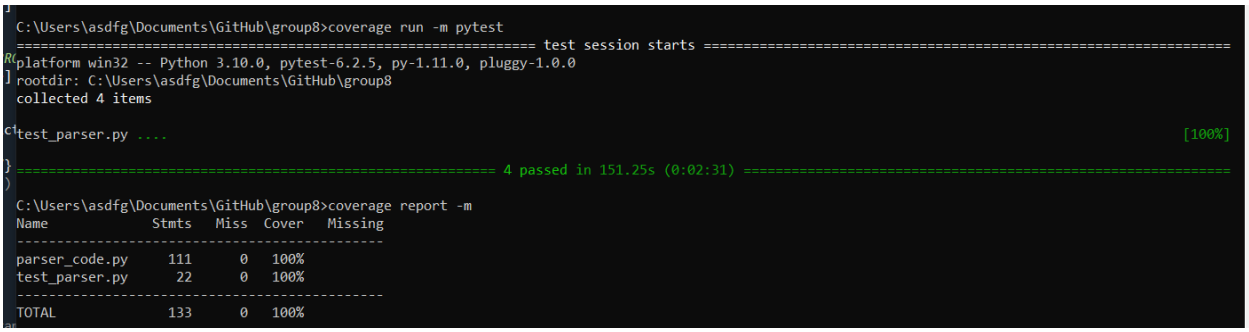
3.1 Load Test

We have also performed Load test for the backend server by using Locust. The target of Locust is load-testing websites and checking the number of concurrent users a system can handle. The result of the load test is also in Test Results.

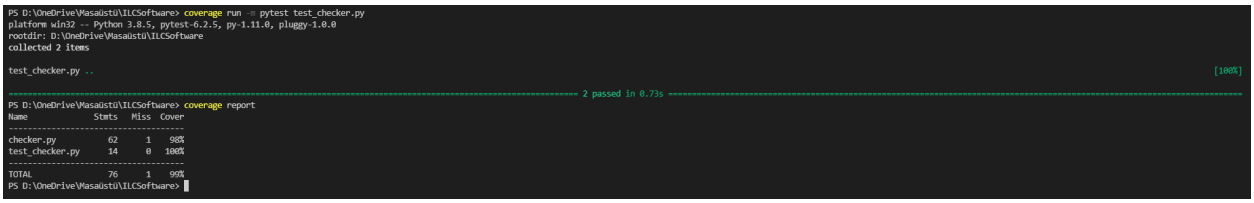
4. Test Results

4.1 Parser Unit Test

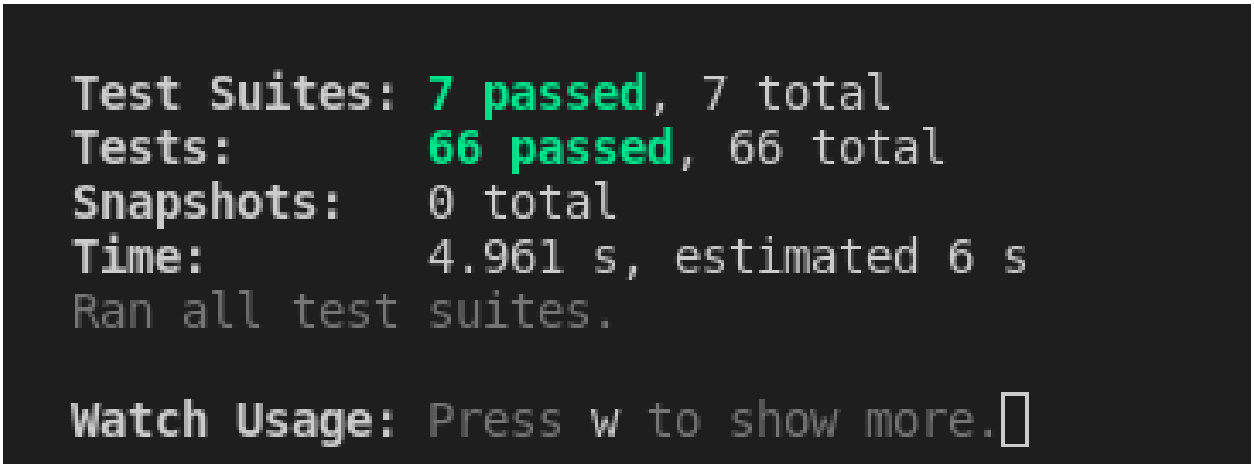
100 % coverage and it passed all cases.



4.2 Checker Unit Test



4.3 FrontEnd Integration Test



4.4 Backend API Request Test Result:

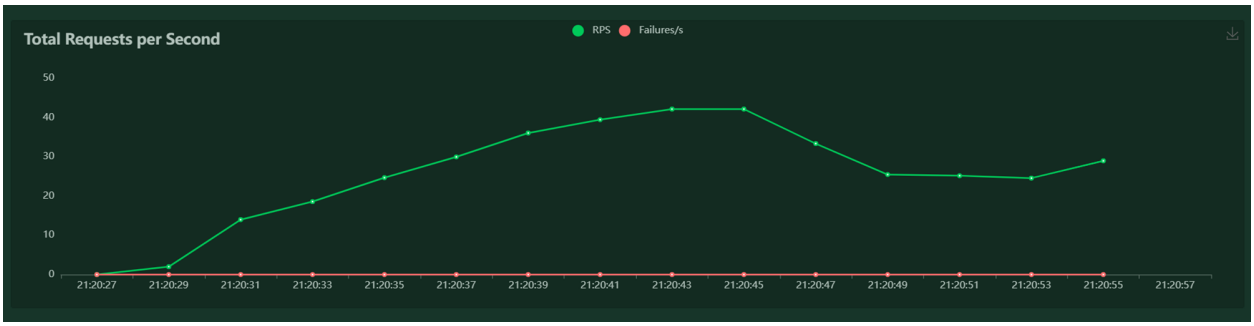
```
(swengvenv) PS C:\Users\ahmet\OneDrive\Masaüstü\Kodlama Projeleri\Software Engineering\group8> python backend_api_test.py
.........F........FF.
======================================================================
FAIL: test_1 (__main__.TestOldRepo)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\ahmet\OneDrive\Masaüstü\Kodlama Projeleri\Software Engineering\group8\backend_api_test.py", line 180, in test_1
    self.assertEqual(resp1.status_code, 200)
AssertionError: 500 != 200

======================================================================
FAIL: test_3 (__main__.TestRepo)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\ahmet\OneDrive\Masaüstü\Kodlama Projeleri\Software Engineering\group8\backend_api_test.py", line 144, in test_3
    self.assertEqual(resp1.status_code, 400)
AssertionError: 500 != 400

======================================================================
FAIL: test_4 (__main__.TestRepo)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\ahmet\OneDrive\Masaüstü\Kodlama Projeleri\Software Engineering\group8\backend_api_test.py", line 156, in test_4
    self.assertEqual(resp1.status_code, 400)
AssertionError: 500 != 400

----------------------------------------------------------------------
Ran 21 tests in 29.304s

FAILED (failures=3)
(swengvenv) PS C:\Users\ahmet\OneDrive\Masaüstü\Kodlama Projeleri\Software Engineering\group8>
```

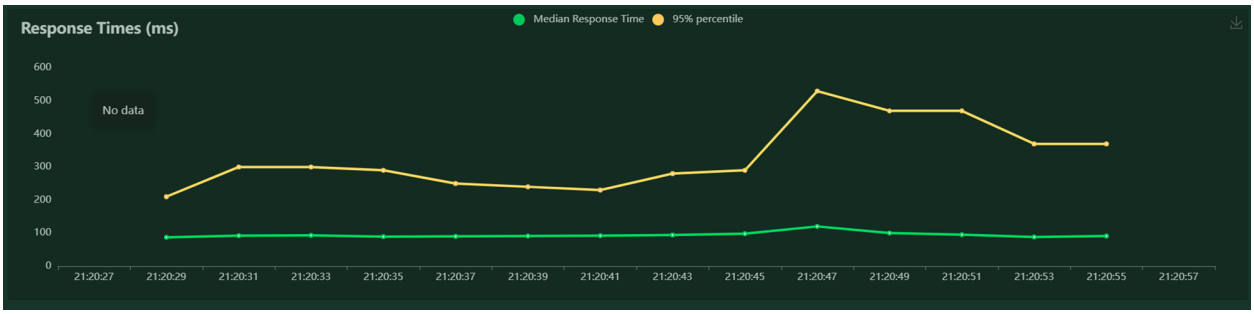
4.5 Load Test

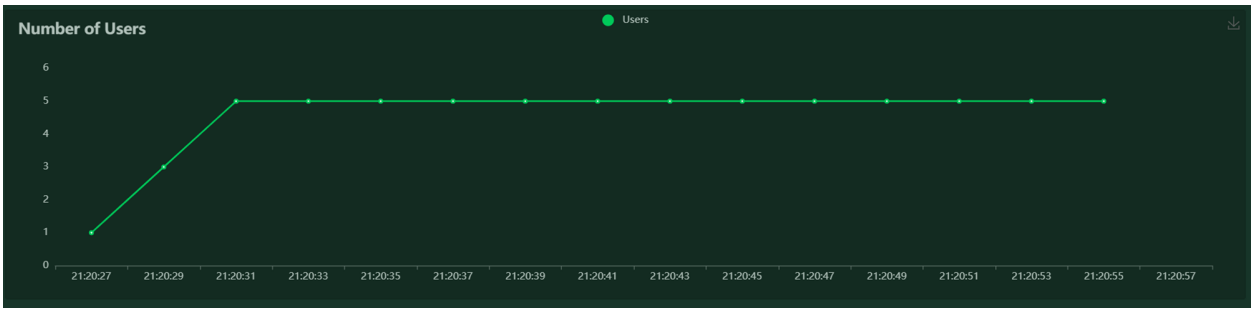
Totally one thousand and twenty-two requests are sent and none of them failed. Average response time is obtained as one hundred and thirty-six milliseconds. During the load test, requests per second reached up to forty.

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/profile	512	0	93	220	135	71	2662	68	15.7	0
GET	/repo	510	0	91	210	137	69	2660	3	15.8	0
	Aggregated	**1022**	**0**	**92**	**210**	**136**	**69**	**2662**	**36**	**31.5**	**0**

Throughout the test, median response time stays stable as can be seen in the above graph. In the below graph, the number of concurrent users is shown.


Total Requests per Second
RPS
Failures/s
50
40
30
20
10
0
21:20:27
21:20:29
21:20:31
21:20:33
21:20:35
21:20:37
21:20:39
21:20:41
21:20:43
21:20:45
21:20:47
21:20:49
21:20:51
21:20:53
21:20:55
21:20:57


Response Times (ms)
Median Response Time
95% percentile
No data
600
500
400
300
200
100
0
21:20:27
21:20:29
21:20:31
21:20:33
21:20:35
21:20:37
21:20:39
21:20:41
21:20:43
21:20:45
21:20:47
21:20:49
21:20:51
21:20:53
21:20:55
21:20:57


Number of Users
Users
6
5
4
3
2
1
0
21:20:27
21:20:29
21:20:31
21:20:33
21:20:35
21:20:37
21:20:39
21:20:41
21:20:43
21:20:45
21:20:47
21:20:49
21:20:51
21:20:53
21:20:55
21:20:57