CS673 Software Engineering
Team 1 - Trackr
Software Test Document

Team Member	Role(s)	Signature	Date
Weijie Liang	QA Leader	*Weijie Liang*	05/29/2022

Revision history

Version	**Author**	**Date**	**Change**
0.1	**Weijie Liang**	**05/29/2022**	**Initial Test Result Added**

• Testing Summary

- Unit Testing:

  The purpose of unit testing is to verify that each unit of the application is developed and executed as designed in the user story. Developers use JUnit5 and Mockito to develop tests and reuse unit tests when units change.

- Integration testing:

  The purpose of integration testing is to test the integrated components to verify that they work as expected. These tests are performed manually. All results will be documented in this document. Unit tests can be part of a continuous integration framework.

- System Testing:

  The purpose of system testing is to test the system as a whole. All components are integrated to verify that the system works as expected. These tests are performed automatically.

- Acceptance Testing:

  The purpose of Acceptance Testing is to verify how well the system meets major functional requirements written in the user stories and some nonfunctional requirements. These tests are performed manually.

- Regression Testing:

  The purpose of Regression Tests is to re-run functional and non-functional tests to ensure that previously developed and tested software works as expected after changes are made. These tests are performed automatically every time we open a pull request through GitHub Actions.

- ## Manual Testing Report

In this section, you will give a detailed description of each manual test case performed and the result. If this is a previous You shall list what are existing tests developed in the previous semester and what are new tests developed currently.

Here is a sample template that can be used for each test case. For system tests or acceptance tests, you may also include some screenshots.

- Test case ID, name
- New or old:
- Test items: (what do you test)
- Test priority (high/medium/low)
- Dependencies (to other test case/requirement if any):
- Preconditions: (if any)
- input data:
- Test steps:
- Postconditions:
- Expected output:
- Actual output:
- Pass or Fail:
- Bug id/link: (this should link to your github issue id)
- Additional notes:

(You can use an additional spreadsheet for this section as well)
CS673_STD_TestCase_team1

Test case name	New or Old	Test item	Test priority	Dependencies	Preconditions
Create user account	New	Add a new user account	High	None	None
Login	New	Use existing user information to login	High	None	None
Test invalid scenarios 1(Create Duplicate user names)	New	Test the corresponding function with a value that does not match the input conditions	High	None	None
Test invalid scenarios 2(wrong login info)	New	Test the corresponding function with a value that does not match the input conditions	High	None	None

Test case name	Input data	Test steps	Postconditions

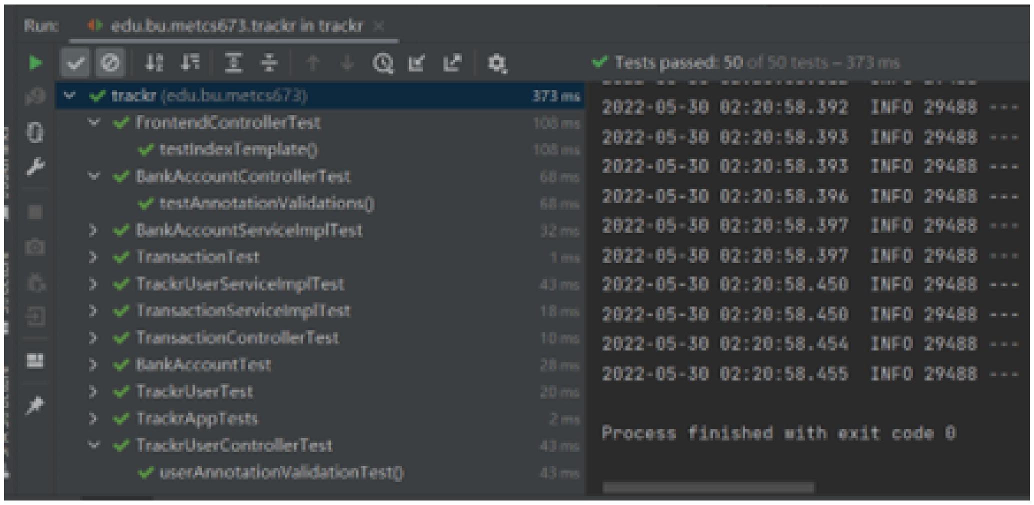
Create user account	First name(WEIJIE); Last name(LIANG); Username(WEL157); Email address(weijiel@bu.edu); Password(123456)	1. Start the page by entering: https://trackr-dev.herokuapp.com/ 2. Fill in the input box with the appropriate information 3. Click “Sign Up!” Button 4. See If a new account is successfully created	The user is able to log in with the created account
Login	Username(WEL157); Password(123456)	1. Start the page by entering: https://trackr-dev.herokuapp.com/ 2. Click the “Login” Button at the top of the page 3. Fill in the input box with the appropriate information 4. Click the “Login” Button 5. See if the account login successfully	The user login to their own account
Test invalid scenarios 1(Create Duplicate user names)	First name(WEIJIE); Last name(LIANG); Username(WEL157); Email address(weijiel@bu.edu); Password(123456)	1. After the former registration, click the “Home” button. 2. Fill in the input box with the appropriate information 3. Click “Sign Up!” Button 4. See If a new account is successfully created	The account should not be created
Test invalid scenarios 2(wrong login info)	Username(WEL157); Password(987654)	1. Start the page by entering: https://trackr-dev.herokuapp.com/ 2. Click the “Login” Button at the top of the page 3. Fill in the input box with the appropriate information 4. Click the “Login” Button 5. See if the account login successfully	The authentication should not be passed

Test case name	Expected output	Actual output	Pass or Fail	Bug id/link	Additional notes
Create user account	Account successfully created!	Account successfully created!	Pass	None	None
Login	Successfully authenticated!	Successfully authenticated!	Pass	None	None
Test invalid scenarios 1(Create Duplicate user names)	Invalid USERNAME value. Please use another value.	Invalid USERNAME value. Please use another value.	Pass	None	Also tested the case of different user names but the same mailbox, account added successfully, need to follow up to discuss whether to reject duplicate mailboxes
Test invalid scenarios 2(wrong login info)	Invalid Login Credentials	Invalid Login Credentials	Pass	None	None

● Automated Testing Report

Describe briefly the automated testing you have done, including where the test code resides in your code repository, what test frameworks are used, and the screen shots or generated testing report.

To test our API, we will use the Mockito framework to write our automation tests. There are 50 automation tests completed so far, and they are geared toward the Bank Account, User, and Transaction aspects of the functionality.



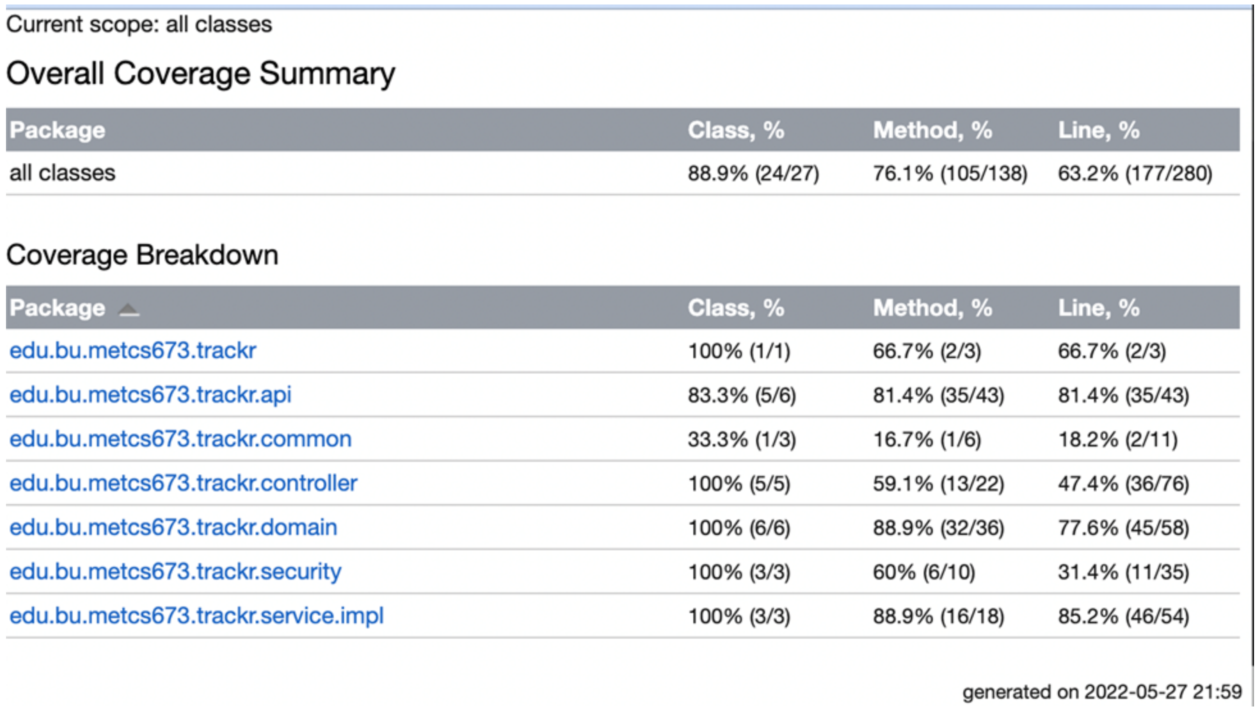
```
Run:  edu.bu.metcs673.trackr in trackr
✓ Tests passed: 50 of 50 tests – 373 ms

✓ trackr (edu.bu.metcs673)                     373 ms
  ✓ FrontendControllerTest                     108 ms
    ✓ testIndexTemplate()                      108 ms
  ✓ BankAccountControllerTest                   68 ms
    ✓ testAnnotationValidations()               68 ms
  ✓ BankAccountServiceImplTest                  32 ms
  ✓ TransactionTest                              1 ms
  ✓ TrackrUserServiceImplTest                   43 ms
  ✓ TransactionServiceImplTest                  18 ms
  ✓ TransactionControllerTest                   10 ms
  ✓ BankAccountTest                             28 ms
  ✓ TrackrUserTest                              20 ms
  ✓ TrackrAppTests                               2 ms
  ✓ TrackrUserControllerTest                    43 ms
    ✓ userAnnotationValidationTest()            43 ms

2022-05-30 02:20:58.392  INFO 29488 ---
2022-05-30 02:20:58.393  INFO 29488 ---
2022-05-30 02:20:58.393  INFO 29488 ---
2022-05-30 02:20:58.396  INFO 29488 ---
2022-05-30 02:20:58.397  INFO 29488 ---
2022-05-30 02:20:58.397  INFO 29488 ---
2022-05-30 02:20:58.450  INFO 29488 ---
2022-05-30 02:20:58.450  INFO 29488 ---
2022-05-30 02:20:58.454  INFO 29488 ---
2022-05-30 02:20:58.455  INFO 29488 ---

Process finished with exit code 0
```

The screenshots above show the results of the automated tests we have performed.

- ## Testing Metrics

In this section, you shall report any metrics used for the evaluation, e.g. # of test cases, test coverage, defects rate, etc.

Unit Test Code Coverage:



Current scope: all classes

Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	88.9% (24/27)	76.1% (105/138)	63.2% (177/280)

Coverage Breakdown

Package	Class, %	Method, %	Line, %
edu.bu.metcs673.trackr	100% (1/1)	66.7% (2/3)	66.7% (2/3)
edu.bu.metcs673.trackr.api	83.3% (5/6)	81.4% (35/43)	81.4% (35/43)
edu.bu.metcs673.trackr.common	33.3% (1/3)	16.7% (1/6)	18.2% (2/11)
edu.bu.metcs673.trackr.controller	100% (5/5)	59.1% (13/22)	47.4% (36/76)
edu.bu.metcs673.trackr.domain	100% (6/6)	88.9% (32/36)	77.6% (45/58)
edu.bu.metcs673.trackr.security	100% (3/3)	60% (6/10)	31.4% (11/35)
edu.bu.metcs673.trackr.service.impl	100% (3/3)	88.9% (16/18)	85.2% (46/54)

generated on 2022-05-27 21:59

- ## References
- ## Glossary