

CS673 Software Engineering
Team 1 - Trackr
Software Test Document



<u>Team Member</u>	<u>Role(s)</u>	<u>Signature</u>	<u>Date</u>
Weijie Liang	QA Leader	<i><u>Weijie Liang</u></i>	<u>05/29/2022</u>
Jean Dorancy	Team Leader	<i><u>Jean Dorancy</u></i>	<u>06/12/2022</u>

Revision history

<u>Version</u>	<u>Author</u>	<u>Date</u>	<u>Change</u>
<u>0.1</u>	<u>Weijie Liang</u>	<u>05/29/2022</u>	<u>Initial Test Result Added</u>
<u>0.2</u>	<u>Weijie Liang</u>	<u>06/11/2022</u>	<u>Updates on Manual Test and Automated Test Report</u>
<u>0.3</u>	<u>Weijie Liang</u>	<u>06/19/2022</u>	<u>Updates on Manual Test and Automated Test Report, Added Test Metric</u>

[Testing Summary](#)

[Manuel Tests Reports](#)

[Automated Testing Reports](#)

[Testing Metrics](#)

[References](#)

[Glossary](#)

● 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

We have a sample of the manual test cases below in this document. Please go to the sheet for all the cases: [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	A user account information that has never been used before is used to successfully create a user
Login	New	Use existing user information to login	High	None	user account needs to exist to test its successful login
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	A used user account information is used in order to successfully report an error
Test invalid scenarios 2(wrong login info)	New	Test the corresponding function with a value that does not match the input conditions	High	None	An incorrect user account information is used in order to successfully report an error

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)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 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

To test our API, we will use the Mockito framework to write our automation tests. There are 215 automation tests completed so far, and they are geared toward the Bank Account, User, Transaction and Front end aspects of the functionality.

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 163, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.327 s
[INFO] Finished at: 2022-06-19T13:08:54-04:00
[INFO] -----
```

```
PASS src/main/js/account/__tests__/account.test.js
PASS src/main/js/auth/__tests__/login.test.js
PASS src/main/js/transaction/__tests__/transaction-form.test.js
PASS src/main/js/auth/__tests__/login-form.test.js
PASS src/main/js/account/__tests__/bank-account-form.test.js
PASS src/main/js/user/__tests__/profile-form.test.js
PASS src/main/js/user/__tests__/sign-up-form.test.js

Test Suites: 10 passed, 10 total
Tests: 52 passed, 52 total
Snapshots: 0 total
Time: 3.365 s
Ran all test suites matching /test/i.
```

● Testing Metrics

- Number of test cases:
 - Automated test case: 163
 - Manual test case: 28
- Number of defects: Was: High level: 5
Median level: 3
Now: High level: 0
Median level: 3
- Total man hour: 44h
- Test passing rate: 98.4%
- User story points counts: 95 points
- Unit Test Code Coverage: **As shown in the figure below**↓

Frontend Report:

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	76.34	58.69	56.09	76.08	
account	71.42	75	43.75	70.58	
account-preview.js	81.81	50	66.66	80	15,33
account.js	33.33	100	20	33.33	22-48
bank-account-form.js	100	100	100	100	
bank-account-modal.js	100	50	100	100	25
bank-account.js	25	100	0	25	11-40
confirmation-modal.js	66.66	50	50	66.66	35
auth	100	100	100	100	
login-form.js	100	100	100	100	
login.js	100	100	100	100	
logout.js	100	100	100	100	
transaction	66.66	50	46.15	66.66	
transaction-conf-modal.js	66.66	50	50	66.66	35
transaction-form.js	75	50	66.66	75	17-29,71
transaction-modal.js	100	50	100	100	19-27
transaction-preview.js	37.5	100	28.57	37.5	30,62-91
user	86.36	57.14	75	86.36	
profile-form.js	80	57.14	60	80	20,120-157
profile.js	100	100	100	100	
sign-up-form.js	100	100	100	100	

Test Suites: 10 passed, 10 total
Tests: 52 passed, 52 total

Backend Report:

Current scope: all classes

Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	92.3% (24/26)	76.1% (118/155)	65.7% (255/388)

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	100% (1/1)	63.6% (7/11)	50% (7/14)
edu.bu.metcs673.trackr.bankaccount	100% (6/6)	77.8% (28/36)	58.9% (43/73)
edu.bu.metcs673.trackr.common	66.7% (2/3)	62.5% (5/8)	63.6% (7/11)
edu.bu.metcs673.trackr.controller	100% (2/2)	42.9% (3/7)	23.1% (3/13)
edu.bu.metcs673.trackr.security	100% (3/3)	100% (13/13)	100% (76/76)
edu.bu.metcs673.trackr.transaction	100% (5/5)	83.8% (31/37)	70.1% (61/87)
edu.bu.metcs673.trackr.user	80% (4/5)	72.5% (29/40)	50.5% (56/111)