COMP 354
Test Document for the project myMoney

Team PA-PK

April 2, 2018

Table 1: Team

Name	ID Number
Anne-Laure Ehresmann	27858906
Marc-Antoine Dube	40029307
Kadeem Caines	26343600
Abdel Rahman Jawhar	27192142
Keith Dion	40036340
Hrachya Hakobyan	40041555
Andrew-Smith	40034936
Dongyu Chen	27241909
Yauheni Karaniuk	40005680
Renny Xu	40005262
Wei Wang	40041116

Table 2: Revision history

Version	Date	Changes
1.0	15 March 2018	Completed test document

Contents

List of Figures

List of Tables

1 Introduction

The introduction of the document provides an overview of the entire document, briefly introducing what are its goals, and what information is to be found in it.

2 Test Plan

Describe what forms of testing you plan to do (unit, subsystem, integration), describe briefly the schedule and resources for testing, and how you designed your test cases.

Indicate which qualities (from requirements) were tested and which qualities were not tested.

2.1 System Level Test Cases

All test cases for testing at the system level.

Test Case 1

Purpose
State the purpose of the test. Indicate which requirement and which aspect of that requirement is being tested.

Input Specification
State the context for the test in terms of system state. State the input test data. You may need to mention operations invoked as well as data for the operation. You can cross-reference to actual file data specified in an appendix.

Expected Output
State the expected system response and output. You can cross-reference to actual file data specified in an appendix.

Traces to Use Cases
State which requirements (at the level of use case and scenario) are tested by this test case.

2.2 Subsystem Level Test Cases

All test cases for testing at the subsystem level.

One subsection per subsystem

Subsystem X

2.3 Unit Test cases

com.github.comp354project.model.user.UserService.createUser(User)

Table 3: createUser(User)

Tester Name	Hrachya	
Test Date	1/31/18	
Class Name	com.github.comp354project.model.user.UserService	
Method Name	createUser(User)	
Purpose	This test suite tests the creation of a user	
Use Cases	01	
Test Scenarios		
createUser_withNullUser_shouldThrow		
Input Specification	user	
	null	
Expected Output	ValidationException is thrown	
Actual Output	ValidationException is thrown	
Bug Found	false	
Purpose	No null user can be saved in the database	
testCreateUser_withInvalidUser_shouldThrow		
Input Specification	user	errors
		4
Expected Output	ValidationException is thrown 4 exceptions are thrown because missing fields: username, password, firstname,	
Actual Output	ValidationException is thrown 4 exceptions are thrown because missing fields: username, password, firstname,	
Bug Found	false	
Purpose	No empty value user can be saved in the database	
testCreateUser_withValidUser_shouldReturnUser		
Input Specification	user	
	username: USERNAME password: PASSWORD firstName: FIRSTNAME lastName: LASTNAME	
Expected Output	User ID was autogenerated upon save The saved user is the same as the inputted user	
Actual Output	User ID was autogenerated upon save The saved user is the same as the inputted user	
Bug Found	false	

Purpose	A valid user should be inserted in the database
testCreateUser_withExistingUsername_shouldThrow	
Input Specification	user
	username: USERNAME password: PASSWORD firstName: FIRSTNAME lastName: LASTNAME
Expected Output	ValidationException is thrown
Actual Output	ValidationException is thrown
Bug Found	false
Purpose	A user cannot be created if the username is already taken

com.github.comp354project.model.user.UserService.deleteBankAccount(Account)

Table 4: deleteBankAccount(Account)

Tester Name	Anne-Laure	
Test Date	3/7/18	
Class Name	com.github.comp354project.model.user.UserService	
Method Name	deleteBankAccount(Account)	
Purpose	This test suite tests the deletion of a bank account	
Use Cases	04	
Test Scenarios		
testDeleteBankAccount_withNullAccount_ShouldThrow		
Input Specification	account	
	null	
Expected Output	ValidationException is thrown	
Actual Output	ValidationException is thrown	
Bug Found	false	
Purpose	No null account can be passed to the function	
testDeleteBankAccount_withoutBeingLoggedIn_ShouldThrow		
Input Specification	account	testUser
	ID: 1 user: testUser bankName: TD type: Checking balance: 15823.12	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111
Expected Output	AuthenticationException is thrown	
Actual Output	AuthenticationException is thrown	

Bug Found	false		
Purpose	A user that is not authenticated cannot delete his accounts		
testDeleteBankAccount_withoutProperAuthorisation_ShouldThrow			
Input Specification	testUser	user2	testAccount
	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111	username: username password: password firstName: firstname lastName: lastname ID: 999	ID: 1 user: testUser bankName: TD type: Checking balance: 15823.12
Expected Output	AuthorisationException is thrown		
Actual Output	AuthorisationException is thrown		
Bug Found	false		
Purpose	A user cannot modify the accounts of another user		
testDeleteBankAccount_WithProperAuthorisation_ShouldSucceed			
Input Specification	testUser	testAccount	invocationCount
	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111	ID: 1 user: testUser bankName: TD type: Checking balance: 15823.12	1
Expected Output	Execution of the deletion of the account once		
Actual Output	Execution of the deletion of the account once		
Bug Found	false		
Purpose	An authenticated user should succeed in deleting his own bank accounts		

com.github.comp354project.model.user.UserService.deleteUser(User)

Table 5: deleteUser(User)

Tester Name	Abed Jawhar
Test Date	3/13/18
Class Name	com.github.comp354project.model.user.UserService
Method Name	deleteUser(User)
Purpose	This test suite tests the creation of a user
Use Cases	02
Test Scenarios	

testDeleteUser_withNullUser_shouldThrow				
Input Specification	user			
	null			
Expected Output	ValidationException is thrown			
Actual Output	ValidationException is thrown			
Bug Found	false			
Purpose	A null user can't be deleted			
testDeleteUser_withNonexistantUser_shouldThrow				
Input Specification	testUser			
	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111			
Expected Output	ValidationException is thrown			
Actual Output	ValidationException is thrown			
Bug Found	false			
Purpose	A user that does not exist can't be deleted			
testDeleteUser_withExistingtUser_shouldSucceed				
Input Specification	testUser	returnSize		
	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111	0		
Expected Output	The number of users with ID 1 is 0			
Actual Output	The number of users with ID 1 is 0			
Bug Found	false			
Purpose	A valid user should be deleted			
testDeleteUser_withExistingtUser_shouldDeleteAssociatedAccounts				
	testUser	testAccount	returnSize	deleteAccountInvoc

Input Specification	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111	ID: 1 user: testUser bankName: TD type: Checking balance: 15823.12	0	1
Expected Output	The number of users with ID 1 is 0 Delete account should be invocated 1 time			
Actual Output	The number of users with ID 1 is 0 Delete account should be invocated 1 time			
Bug Found	false			
Purpose	A valid user should be deleted and his accounts also			

com.github.comp354project.model.user.UserService.updateUser(User)

Table 6: updateUser(User)

Tester Name	Abed Jawhar
Test Date	3/13/18
Class Name	com.github.comp354project.model.user.UserService
Method Name	updateUser(User)
Purpose	This test suite tests the update of a user
Use Cases	02
Test Scenarios	
testUpdateUser_withNullUser_shouldThrow	
Input Specification	user
	null
Expected Output	ValidationException is thrown
Actual Output	ValidationException is thrown
Bug Found	false
Purpose	A null user can’t be updated
testUpdateUser_withNonexistenttUser_shouldThrow	
	testUser

Input Specification	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111			
Expected Output	ValidationException is thrown			
Actual Output	ValidationException is thrown			
Bug Found	false			
Purpose	A user that does not exist can't be updated			
testUpdateUser_withValidUser_shouldSucceed				
Input Specification	testUser	firstName	lastName	password
	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111	Abed	jawhar	admin2
Expected Output	The firstName is updated to 'Abed' The lastName is updated to 'jawhar' The password is updated to 'admin2'			
Actual Output	The firstName is updated to 'Abed' The lastName is updated to 'jawhar' The password is updated to 'admin2'			
Bug Found	false			
Purpose	A valid user should be updated			

com.github.comp354project.model.auth.SessionManager.login(username, password)

Table 7: login(username, password)

Tester Name	Hrachya
Test Date	2/7/18
Class Name	com.github.comp354project.model.auth.SessionManager
Method Name	login(username, password)
Purpose	This test suite tests the login of a user
Use Cases	02
Test Scenarios	

testLogin_withInvalidCredentials_shouldThrow					
Input Specification	username	password			
Expected Output	ValidationException is thrown				
Actual Output	ValidationException is thrown				
Bug Found	false				
Purpose	A user with invalid credentials should not be able to login				
testLogin_withValidCredentials_shouldReturnUser					
Input Specification	testUser	username	password	loggedIn	authenticateInvoc
	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111			true	1
Expected Output	The method authenticate is invoked 1 time The user is logged in The logged in user is equal to 'testUser' object				
Actual Output	The method authenticate is invoked 1 time The user is logged in The logged in user is equal to 'testUser' object				
Bug Found	false				
Purpose	A user with valid credentials should be able to login				
testLogin_withLoggedInUser_shouldThrow					
Input Specification	username	password			
Expected Output	UserLoggedInException is thrown				
Actual Output	UserLoggedInException is thrown				
Bug Found	false				
Purpose	A user that is already logged in should not be able to login again				

com.github.comp354project.model.auth.SessionManager.logout()

Table 8: logout()

Tester Name	Hrachya
Test Date	2/7/18
Class Name	com.github.comp354project.model.auth.SessionManager
Method Name	logout()
Purpose	This test suite tests the function to logout

Use Cases	02			
Test Scenarios				
testLogin_withInvalidCredentials_shouldThrow				
Input Specification	username	password	isLoggedIn	currentUser
			false	null
Expected Output	After logout, the login status should be false After logout, the current user should be null			
Actual Output	After logout, the login status should be false After logout, the current user should be null			
Bug Found	false			
Purpose	A user should be completely logged out of the application			

com.github.comp354project.model.auth.AuthenticationService.authenticate(username, password)

Table 9: authenticate(username, password)

Tester Name	Hrachya		
Test Date	2/3/18		
Class Name	com.github.comp354project.model.auth.AuthenticationService		
Method Name	authenticate(username, password)		
Purpose	This test suite tests the authentication of the user		
Use Cases	01		
Test Scenarios			
testAuthenticate_withInvalidUsernameOrPassword_shouldThrow			
Input Specification	username	password	
	null	null	
Expected Output	ValidationException is thrown The number of Validation Errors is equal to 2		
Actual Output	ValidationException is thrown The number of ValidationErrors is equal to 2		
Bug Found	false		
Purpose	A user with invalid credentials should not be able to authenticate		
testAuthenticate_withNonexistentUsername_shouldThrow			
Input Specification	username	password	
	usernam e	password	
Expected Output	ValidationException is thrown		
Actual Output	ValidationException is thrown		
Bug Found	false		
Purpose	A user witha nonexistent username should not be able to authenticate		
testAuthenticate_withIncorrectPassword_shouldThrow			
	testUser	username	password

Input Specification	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 11 1111	admin	INCORRECT_PASSWORD
Expected Output	ValidationException is thrown		
Actual Output	UserLoggedInException is thrown		
Bug Found	false		
Purpose	Authenticat ion with a valid username but an incorrect password should fail		
testAuthenticate_withCorrectCredentials_shouldReturnUser			
Input Specification	testUser	username	password
	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 11 1111	admin	admin
Expected Output	The authentication is successful and the authenticated u ser is returned The authenticated user is equal to the 'testUser' object		
Actual Output	The authentication is successful and the authenticated use r is returned The authenticated user is equal to the 'testUser' object		
Bug Found	false		
Purpose	Authentication with a valid username but an incorrect password should fail		

com.github.comp354project.model.account.AccountService.addAccount (request, user)

Table 10: addAccount(request, user)

Tester Name	Hrachya
Test Date	2/7/18
Class Name	com.github.comp354project.model.account.AccountService
Method Name	addAccount(request, user)
Purpose	This test suite tests the functionality of adding a new bank account
Use Cases	03
Test Scenarios	

testAddAccount_withInvalidParameters_shouldThrow		
Input Specification	request	accountOwner
	null	null
Expected Output	ValidationException is thrown The number of Validation Errors is equal to 2	
Actual Output	ValidationException is thrown The number of ValidationErrors is equal to 2	
Bug Found	false	
Purpose	Adding an account with invalid request or user should fail	
testAddAccount_withNonexistentRemoteAccount_shouldThrow		
Input Specification	request	accountOwner
	ID: 1	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111
Expected Output	AccountDoestNotExistException is thrown	
Actual Output	AccountDoestNotExistException is thrown	
Bug Found	false	
Purpose	A request f or adding a nonexistent account should fail	
testAddAccount_withInvalidUser_shouldThrow		
Input Specification	request	accountOwner
	ID: 1	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111
Expected Output	ValidationException is thrown	
Actual Output	ValidationException is thrown	
Bug Found	false	
Purpose	Adding an account with an invalid owner should throw	
testAddAccount_withExistingAccount_shouldThrow		
	request	accountOwner

Input Specification	ID: 1	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111	
Expected Output	AccountExistsException is thrown		
Actual Output	AccountExistsException is thrown		
Bug Found	false		
Purpose	Adding an already existing account should throw		
testAddAccount_withValidAccount_shouldReturnValidAccount			
Input Specification	request	accountOwner	expectedAccount
	ID: 1	ID: 1 firstName: Hrachya lastName: Hakobyan username: admin password: admin email: sample@email.com address: address phone: 111111	ID: 1 user: accountOwner bankName: TD type: Checking balance: 15823.12
Expected Output	The account is fetched and persisted in the database T he persisted account is equal to the 'expectedAccount' object The returned account is equal to the 'expectedAccount' object		
Actual Output	The account is fetched and persisted in the database The persisted account is equal to the 'expectedAccount' object The returned account is equal to the 'expectedAccount' object		
Bug Found	false		
Purpose	Adding a va lid account with a valid owner must succeed		

com.github.comp354project.model.account.AccountService.deleteAccount(account)

Table 11: deleteAccount(account)

Tester Name	Anne-Laure
Test Date	3/5/18
Class Name	com.github.comp354project.model.account.AccountService
Method Name	deleteAccount(account)
Purpose	This test suite tests the functionality of removing a user's bank account
Use Cases	04
Test Scenarios	

testDeleteAccount_withNullAccount_shouldThrow	
Input Specification	account
	null
Expected Output	ValidationException is thrown
Actual Output	ValidationException is thrown
Bug Found	false
Purpose	Delete a nu ll account should fail
testDeleteAccount_withAccountWithNullID_shouldThrow	
Input Specification	account
	ID: null user: null bankName: type: balance: 0
Expected Output	ValidationException is thrown
Actual Output	ValidationException is thrown
Bug Found	false
Purpose	Deleting an account with null ID should fail
testDeleteAccount_withNonExistentAccount_shouldThrow	
Input Specification	account
	ID: 1 user: accountOwner bankName: TD type: Checking balance: 15823.12
Expected Output	ValidationException is thrown
Actual Output	ValidationException is thrown
Bug Found	false
Purpose	Deleting an nonexistent account should fail
testDeleteAccount_withValidAccount_shouldSucceed	
Input Specification	account
	ID: 1 user: accountOwner bankName: TD type: Checking balance: 15823.12
Expected Output	The account is deleted from the database
Actual Output	The account is deleted from the database
Bug Found	false
Purpose	Deleting an existing account should succeed
testDeleteAccount_withValidAccount_shouldDeleteAllAssociatedTransactionsAndAccount	
	account

Input Specification	ID: 1 user:accountOwner bankName: TD type: Checking balance: 15823.12 transactions: [object Object]	
Expected Output	The account is deleted from the database All the associated transactions are deleted from the database	
Actual Output	The account is deleted from the database All the associated transactions are deleted from the database	
Bug Found	false	
Purpose	Deleting an existing account should delete all associated transactions	

com.github.comp354project.model.account.TransactionService.update TransactionCategory(transactionID, category)

Table 12: updateTransactionCategory(transactionID, category)

Tester Name	Hrachya	
Test Date	3/4/18	
Class Name	com.github.comp354project.model.account.TransactionService	
Method Name	updateTransactionCategory(transactionID, category)	
Purpose	This test suite tests the functionality of updating the category of a transaction	
Use Cases	08	
Test Scenarios		
testUpdateCategory_withNullTransactionID_shouldThrow		
Input Specification	transactionID	category
	null	Leisure
Expected Output	ValidationException is thrown	
Actual Output	ValidationException is thrown	
Bug Found	false	
Purpose	Updating a null transaction ID should fail	
testUpdateCategory_withNonexistentTransaction_shouldThrow		
Input Specification	transactionID	category
	111111	Leisure
Expected Output	ValidationException is thrown	
Actual Output	ValidationException is thrown	
Bug Found	false	
Purpose	Updating a nonexistent transaction should fail	
testUpdateCategory_withNullCategory_shouldSucceed		

Input Specification	transactionID	category
	10	nu ll
Expected Output	The ’category’ of the transaction with the specified ID is set to null	
Actual Output	The ’category’ of the transaction with the specified ID is set to null	
Bug Found	false	
Purpose	Updating th e category of a valid transaction to null must succeed	
testUpdateCategory_withEmptyCategory_shouldSucceed		
Input Specification	transactionID	category
	10	
Expected Output	The ’category’ of the transaction with the specified ID is set to ”	
Actual Output	The ’category’ of the transaction with the specified ID is set to ”	
Bug Found	false	
Purpose	Updating th e category of a valid transaction to an empty string must succeed	
testUpdateCategory_withValidCategory_shouldSucceed		
Input Specification	transactionID	category
	10	Leisure
Expected Output	The ’category’ of the transaction with the specified ID is set to ’Leisure’	
Actual Output	The ’category’ of the transaction with the specified ID is set to ’Leisure’	
Bug Found	false	
Purpose	Updating the category of a valid transaction must succeed	
testUpdateCategory_withInvalidCategory_shouldThrow		
Input Specification	transactionID	category
	10	AAAAAAAAAAAAAAAAA
Expected Output	ValidationException is thrown	
Actual Output	ValidationException is thrown	
Bug Found	false	
Purpose	Updating th e category of a valid transaction to a an invalid value as determined by the bus iness rules must fail	

All test cases for testing at the unit level.

One subsection per unit

Unit X

3 Test Results

List the tests, indicating which passed and which did not pass. List requirements indicating the percentage of tests that passed for that requirement.

4 References

A Description of Input Files

Describe/include test data from input files.

B Description of Output Files

Describe/include test expected output that are output files.