System Test Plan for

Smart Locker

Project Team:

Wangyi Zhu
Yangyang Li
Haishuo Zhang
Kaiqi Li
Zheng Gao

Contact Person:

Wangyi Zhu
wangyiz@sfu.ca

Submitted to:

Dr. Andrew Rawicz – ENSC 440W
Steve Whitmore – ENSC 305W
School of Engineering Science
Simon Fraser University

Issue Date:

October 14, 2014

Test Suite	Test Group	Test Case
9.1 PHP Server and MySQL Server Testing	9.1.1 Android Application Interface	9.1.1.1 Input any username and a password pair, click login, the interface should receive the username-password pair with a command to check the credentials.
		9.1.1.2 Perform an account registration, locker request, password change request and unlock request, and verify the interface receives the corresponding data and command.
	9.1.2 Query Generating Logic	9.1.2.1 Input a non- existent command and invalid parameter, and the Logic should output an error “Command not found”.
		9.1.2.2 Input a valid command and invalid parameter, including wrong type, missing parameter, extra parameter and out-of-range parameter, the Logic should output an error “Invalid parameter”.
		9.1.2.3 Input every valid command with valid parameter in turn, the Logic should yield the corresponding SQL queries for each command.
	9.1.3 MySQL Server Interface	9.1.3.1 Input a query and the MySQL server should receive the query.
	9.1.4 MySQL Server Testing	9.1.4.1 Query the MySQL server to find the number of matching username-password pair in the User table, should return 1 for existing username-password pair and 0 for non-existing ones.
		9.1.4.2 Query the MySQL server to create new row in the User table to store a new username that does not already exist with its password and email address, the new row should be created with the new information.
		9.1.4.3 Repeat the above test with an existing username, an error should be returned.

		9.1.4.4 Query the MySQL server to create new row in the Rental table to store a new user-locker rental that does not exist yet, the new row should be created with the new information.
		9.1.4.5 Repeat the above test with an existing username-lockerID pair, an error should be returned.
		9.1.4.6 Query the MySQL server to remove a row identified by an existing username-lockerID pair, the row should be gone from the table.
		9.1.4.7 Repeat the above test with a non-existing username-lockerID pair, an error should be returned.
		9.1.4.8 Query the MySQL server to retrieve a status on a row defined by an existing username-lockerID pair and the status should be returned.
		9.1.4.9 Repeat the above test with a non-existing username-lockerID pair, an error should be returned.
		9.1.4.10 Repeat the tests for the Rental table on the Locker table.
9.2 Physical Component Testing	9.2.1 Testing Physical Locker	9.2.1.1 Applying 50N forces on both inside and outside of locker body to check if the locker is strong enough to protect the things store inside.
		9.2.1.2 Open locker's door without installing electromagnetic lock to see if it can close by itself properly. If the door is stuck in the middle that means there must be some unexpected fraction on the edge. Double check each side of the door to see if there is some pop-up spot on one edge, remove this by using tools.
	9.2.2 Testing for Electromagnetic	9.2.2.1When the electromagnetic lock is inactivated, try to open the locker by using 60 lb force. In this case, the locker

	lock	door should not be opened. If the locker's door has been opened, check in details which part is falling. Correct the failing part and repeat this test.
		9.2.2.2 Set the default time limit for closing lock to 10 seconds. Test it by opening the lock immediately after sending an activation message. If the door can be open, check if the connection between lock and Arduino is correct or not. Then check if these component is powered. Then check if there is any logical problem in the controlling code. The last thing is check if the lock is still able to use.
		9.2.2.3 Set the default time limit for closing lock to 10 seconds. Test it by opening the lock 10 seconds later after sending an activation message. If the locker can be open, double check this function by closing the door for another 5 seconds. Try opening it again this time. If the door can still be open, check the connection between lock and arduino is correct or not. Then check if these component is powered. Then check if there is any logical problem in the controlling code. The last thing is check if the lock is still able to use.
9.3 NFC Reader Testing	9.3.1 Accepting and denying cases test	9.3.1.1 Using any smartphones to send correct opening message to the NFC Reader and check if the switch is closed and the green LED is on. In Process
		9.3.1.2 Using any smartphones to send incorrect opening message to the NFC Reader and check if the switch remains open and the red LED is on. In Process
		9.3.1.3 Using any smartphones to send 20 random opening message to the NFC Reader and check if the NFC Reader's reaction is correct. Pending
	9.3.2 Different smartphones test	9.3.2.1 Using Nexus 5, Samsung Galaxy to repeat the Cases test. The NFC Reader should work perfectly with different phones. Pending

9.4 Software Application Testing	9.4.1 GUI Testing	9.4.1.1 Each features such as buttons, text fields and text areas are functioning as intended and directing user to the correct activity.
		9.4.1.2 Each features should be able to adapt various unexpected actions by users as many as possible. For example, when users have typed in unintended symbols in text field, or when users have clicked the button more than once, the system would not crash, otherwise, an error message should display to notify user about the situation.
		9.4.1.3 Time to connect different activities should not exceed 5s, otherwise, a timeout error message is needed it to notify users that something has gone wrong.
		9.4.1.4 Application is stable so that it does not crash and quit unexpectedly.
	9.4.2 Functionality Testing	9.4.2.1 Software application should correctly establish connections between GUI and server.
		9.4.2.2 Clicking button(i.e. login and register) should correctly send out user information typed in text field from GUI to server
		9.4.2.3 User information should be correctly verified(i.e. login) or stored(i.e. register) and be sent back from server to GUI to allow user to proceed the next activity.
	9.4.3 Tag Reader Testing	9.4.3.1 Component should be able to read the NDEF message in a NFC tag when the Andrino device is tapping the tag.
	9.4.4 Host-based Card Emulator	9.4.4.1 Component should be able to output incomming key in a NDEF message and the NFC reader have to be able to read the NDEF.