MacGo – Test Report

Group 10

Kartikay Dani - 1230669
David Elsonbaty - 1141043
Simon Quach - 1055287
Barane Paramanathan - 1135924

3/29/2015

Contents

1 *List of Tables*

2 *Change History*

Version	Date	Author	Comments
-	-	danik, elsonbd, quachsh, paramab	Original Content
0	March 29 2015	danik, elsonbd, quachsh, paramab	Initial check In

Table 1 - Change History

3 *Introduction*

3.1 PURPOSE

The purpose of this document is to describe how different test are performed on MacGo in order to ensure the app's security and data authenticity of the users using MacGo.

3.2 SCOPE

We will be testing three applications of our MacGO app. We will be doing the same tests for both the IOS and Android versions of our main app. Our scanner app and cloud code will be tested with different test cases.

The testing report primary focuses on the results by means of functional (unit) tests. Exhaustive testing is the only technique that guarantees program validity and correctness. The successes of the test cases performed by our apps and outlined in this report will provide us with confidence that the implementation is correct.

3.3 DOCUMENT STRUCTURE

This document is broken out into two major sections – Functional Testing and Unit Testing, which are described as follows.

- Functional Testing – Describes how different areas of MacGo are been tested functionality wise.
- Unit Testing – Describes how different units inside MacGo are tested using automated.

4 *Functional Testing*

4.1 REGRESSION TESTING

Regression testing is done on backend cloud data, which involves different test cases to test the following features:

Scenario	No. Of Users	Expected Result	Result	Time (sec)
User creates token	500	Token should be created immediately	Token is immediately created	3
User cancels token before 2 minutes		Token should be invalid	Token is invalid	2
User holds onto token for more than 2 minutes		Token should be invalid	Token is invalid	1
User holds onto token for less than 2 minutes		Token should be valid	Token is valid	1

Table 2 - Token Expiry Test Case (Backend)

Scenario	No. Of Users	Expected Result	Result
Registered user requests to see percentage of money put into specific category	500	Percentage should be shown on app	Percentage is shown

Table 3 - Data Analytics Test Case (Backend)

4.2 USABILITY TESTING FOR MACGO

Five participants were recruited of varying technical levels and were required to perform a series of common tasks that user of MacGo would do. Following are the test results:

User Login – Login Screen					
	Task difficulty	**Time Taken (in sec)**	**Comments**	**Expected Output**	**Output**
Participant 1	Easy	1.2	N/A	User is successfully login or error notification message is displayed.	As Expected
Participant 2	Easy	1.4	Nice User Interface		
Participant 3	Easy	1.3	Easy to use		
Participant 4	Easy	1.5	N/A		
Participant 5	Easy	1.5	Nice User Interface		

Table 4 – Usability Testing for User Login in MacGo

Make Purchase – Main Screen					
	Task difficulty	**Time Taken (in sec)**	**Comments**	**Expected Output**	**Output**
Participant 1	Easy	2	Easy to access	To Generate random QR Code	As Expected
Participant 2	Easy	3	N/A		
Participant 3	Easy	2	One tap purchase		
Participant 4	Easy	3	N/A		
Participant 5	Easy	2	Good Design		

Table 5 - Usability testing on making a purchase in MacGo

Check Purchase History – Account History Screen					
	Task difficulty	**Time Taken (in sec)**	**Comments**	**Expected Output**	**Output**
Participant 1	Easy	3	N/A	Give the list of purchase history happened on the account.	4/5 Test Cases passed. App crashes on multiple clicks on the same purchase.
Participant 2	Easy	3	Should also add time of the purchase specific time.		
Participant 3	Easy	3	N/A		
Participant 4	Easy	3	N/A		
Participant 5	Easy	4	N/A		

Table 6 - Usability testing for checking purchase history

Item History – Item History Screen					
	Task difficulty	**Time Taken (in sec)**	**Comments**	**Expected Output**	**Output**
Participant 1	Easy	3	N/A	Give the list of items specific to purchase.	As Expected
Participant 2	Easy	3	N/A		
Participant 3	Easy	2	N/A		
Participant 4	Easy	4	N/A		
Participant 5	Easy	2	N/A		

Table 7 - Usability testing for item history screen

4.3 USABILITY TESTING FOR SCANNER APP

Task 1 – Adding/Subtracting items			
	Task difficulty	Time Taken (in sec)	Comments
Participant 1	Easy	2.15s	N/A
Participant 2	Easy	2.57s	Have category of items for easier access in future
Participant 3	Easy	3.07s	N/A
Participant 4	Easy	2.12s	Categorize items listed
Participant 5	Easy	1.23s	N/A

Table 8 - Usability Testing for Adding Items in Scanner App

Task 2 – Start purchase (after adding in items)			
	Task difficulty	Time Taken (in sec)	Comments
Participant 1	Easy	1.42s	Make start button bigger
Participant 2	Easy	1.68s	N/A
Participant 3	Easy	1.59s	Use “begin purchase” instead of “start”
Participant 4	Easy	2.11s	N/A
Participant 5	Easy	1.88s	N/A

Table 9 - Usability testing for finalizing purchase on Scanner App

5 *Unit Testing*

5.1 MACGO APP

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
1	Login	MacId and Password	Fetch items from database	Network call passes and returns user object	As expected.	PASS
2	Get Updated Balance	Purchase total is set to $0	Fetch balance from backend.	Network call passes and returns $320.20 (Current Balance)	As expected.	PASS
3	Creating new token	No QR Code is read	New Token object is created with random string(primary key of the table)	Random QR Code	As expected.	PASS
4	Making Token Invalid manually	Token is valid	Token Id (primary key of the table)	Token becomes false	As expected.	PASS
5	Getting List of Purchases History	-	UserId	List of purchases is displayed.	As expected.	PASS
6	Getting List of Items specific to purchase	-	Purchase Id, UserId	List of items is displayed specific to the purchase	As expected.	PASS

Table 10 - Unit Testing for MacGo

5.2 SCANNER APP

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
1	Item Table Testing	No items in table	Fetch items from database	Network call passes and returns a minimum of 5 objects (items)	As expected.	PASS
2	Purchase Total Calculation Testing	Purchase total is set to $0	Quantity of items times the price of items	Sum of price of total items being purchased	As expected.	PASS
3	Camera Read Testing	No QR Code is read	MacGo generated QR code	Unique string of QR code	As expected.	PASS
4	Purchase Table Testing	Empty purchase table	Capture string of unique QR code	Network call passes and returns a minimum of 1 object (item)	As expected.	PASS

Table 11 - Unit Test Cases for Scanner App

Total Specific Systems for MacGo Scanner Passed:	4
Total Specific Systems for MacGo Scanner Failed:	0

Table 12 - Final result of unit test cases on Scanner App