



MOBCOLL
wonderland

Software Test Document

Sinan Küçükköseler
Zeliha Şentürk
Cansu Tetik
Murat Türe
14.04.2013

Table of Contents

Table of Contents	2
Introduction.....	3
Problem Definition	
Purpose of the Document	
Scope of the Document	
Definitions	
Abbreviations	
Test Cases.....	6
Description of the Test Environments	
Register Functionality Test Cases	
Login Functionality Test Cases	
Facebook Login Functionality Test Cases	
Logout Functionality Test Cases	
Create Collection Functionality Test Cases	
Delete Collection Functionality Test Cases	
Search Collection Functionality Test Cases	
Search for Cards Functionality Test Cases	
Search for People Functionality Test Cases	
Add Friend Functionality Test Cases	
Delete Friend Functionality Test Cases	
Responding to Incoming Friend Request Functionality Test Cases	
Trade Cards Functionality Test Cases	
Responding to Trade Requests Functionality Test Cases	
Item Delete Functionality Test Cases	
Image Processing Test Cases	
Performance Test Cases	

Introduction

1.1. Problem Definition

Collecting items has been a mainstream hobby upto this date. There is a hugely wide range of items that a collection can be created. Being a collectionist though, comes with a load if it is taken seriously. Collectionists need to keep the record of the items they own, locate and store them in a proper way. They also need to communicate with othercollectionists.

There are several media collection systems in the market, however many of them lack of some serious and useful functionalities, such as:

- Recognition from device camera for easiness of adding cards to collections
- Communication between the users via social media (e.g. Facebook)
- Sharing collections with other users with friendship functionality

The MOBCOLL project will provide all general features like creating and updating the collections and having a user profile page. Also, all these special functionalities listed above will be ready to use for card collectors.

1.2. Purpose of the Document

In this document; component testing and system testing of MOBCOLL are explained in detail with consequences. Brief definitions of these testing processes are:

Component Testing: The process of showing that the component interface behaves according to its specification.

System testing: The process of testing software or hardware conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

1.3. Scope of the Document

In the test phases of the MOBCOLL, the methods of all the activities are tested with the specified scenarios. System Test Document explains and describes all the testing processes, gives details regarding the tests for each test case.

1.4. Definitions

Term	Definition
Operating System	A collection of software that manages computer hardware resources and provides common services for computer programs
Image Processing	Any form of signal processing applied to images (with the aim of recognition in this project)
Test Environment	The development environment that is to be used for the test cases

1.5. Abbreviations

Term	Definition
OS	Operating System
ADT	Android Development Tools
SDK	Software Development Kit
NTY	Not Tested Yet

Test Cases

1.6. Description of the Test Environment

To be able to use MOBCOLL, devices with a touchscreen and Android OS is required. The application will be implemented for devices having an Android version higher than 3.0 and minimum SDK version 11. In addition to these requirements a camera higher than 1.3 mega-pixel is needed for scanning images and decent image processing, and for accessing the web services, internet connection is essential. Lastly for the client local database, used for offline search feature, 5M free storage space is required.

1.7. Register Functionality Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Register #1	Testing the consequences of registering to the system with a nonexisting user name	Entering a user name that does not exist in the system	Successfully logged in and navigated to 'Login Page'	Pass
Register #2	Testing the consequences of registering to the system with a user name that already exists in the system	Entering user name that already exists in the system	Register attempt denied with a warning indicating that the user name already exists	Pass
Register #3	Testing the consequences	Entering an email that does	Register attempt denied with a	Pass

	of registering to the system with an invalid email address	not contain '@' character in it	warning indicating the email address is not valid	
--	--	---------------------------------	---	--

1.8. Login Functional Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Login #1	Testing the consequences of entering the correct user name and password	Entering correct user name and password	Successfully logged in and navigated to 'Main Page'	Pass
Login #2	Testing the consequences of entering wrong user name and password	Entering wrong user name and/or password	Login attempt denied	Pass

1.9. Facebook Login Functional Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Facebook Login #1	Testing the consequences of entering the correct Facebook user name and password	Entering correct Facebook user name and password	Successfully logged in and navigated to 'Main Page'	Pass
Facebook Login #2	Testing the consequences of entering wrong Facebook user name and password	Entering wrong Facebook user name and/or password	Login attempt denied	Pass

1.10. Logout Functional Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Logout #1	Testing the consequences of logout attempt	Pressing Logout button	Successfully logged out and navigated to 'Login Page'	Pass

1.11. Create Collection Functional Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Create Collection #1	Testing the consequences of entering a unique (among the user's collections) collection name	From 'Main Page' navigating to 'User Collections Page', entering a unique name and pressing Create Collection button	Collection successfully created	Pass
Create Collection #2	Testing the consequences of entering a nonunique (among the user's collections) collection name	From 'Main Page' navigating to 'User Collections Page', entering a nonunique name and pressing Create Collection button	Create collection attempt denied	Pass

1.12. Delete Collection Functional Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Delete Collection #1	Testing the consequences of delete collection attempt from 'User Collections Page'	From 'User Collections Page', navigating to 'Delete Collection Page', select collections to be deleted	The selected collections are deleted from the collection list	Pass
Delete Collection #2	Testing the consequences of delete collection attempt from 'Collection Page'	From 'User Collection Page', navigating to 'Collection Page' by selecting a collection and pressing the delete collection button	The selected collection is deleted from the collection list	Pass

1.13. Search for Cards Functionality Test Case

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Search for Cards #1	Testing the consequences of searching for a card	Entering a search query in the search area provided on top of each page	Search results according to the query are listed in 'Search Results' page	Pass

1.14. Search for People Functionality Test Case

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Search for People #1	Testing the consequences of searching for people	Entering a search query in the search area provided on 'Search for People' page which can be accessed from the 'Search for People' button provided in 'Profile Page'	Search results according to the query are listed in 'Search for People' page	Pass

1.15. Add Friend Functionality Test Case

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Add Friend #1	Testing the consequences of sending a friend request to another user	Choosing a user from the list created by Search For People action and after being directed to that user's page adding him/her using the button 'Add as Friend'	A friend request is successfully sent to the intended user, and added to the current user's pending friend requests	Pass

1.16. Delete Friend Functionality Test Case

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Delete Friend #1	Testing the consequences of deleting a friend	Choosing a user from the list of friends and after being directed to that user's page unfriending him/her using the button 'Unfriend'	The intended user successfully removed from the list of friends	Pass

1.17. Responding to Incoming Friend Requests Test Case

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Respond Friend Request #1	Testing the consequences of accepting a friend request	After being directed to the 'Incoming Friend Requests Page' by clicking on the 'Incoming Friend Requests' button provided on the 'Main Page', tapping the accept button	Successfully added to the list of friends, and deleted from the incoming requests of the current user, and from the pending requests of the user who sent the request	Pass
Respond Friend Request #2	Testing the consequences of rejecting a friend request	After being directed to the 'Incoming Friend Requests Page' by clicking on the 'Incoming Friend Requests' button provided on the 'Main Page', tapping the reject button	Request successfully deleted from the incoming friend requests of the current user, and from the pending friend requests of the user who sent the request	Pass

1.18. Trade Cards Functionality Test Case

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Trade Cards #1	Testing the consequences of sending a trade request for a friend	After being directed to the 'Trade Cards' page by using the 'Trade Cards' button from a friend's page selecting the cards that are wished to be traded	A trade request is successfully sent to the intended user, and added to the current user's pending trade requests	Pass

1.19. Responding to Incoming Trade Requests Functionality Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Respond Trade Request #1	Testing the consequences of accepting a trade request	After being directed to the 'Incoming Trade Requests Page' by clicking on the 'Incoming Trade Requests' button provided on the 'Main Page', tapping the accept button	The trade operation is successfully operated between the users, and deleted from the incoming trade requests of the current user, and from the pending trade requests of the user who sent the request	Pass
Respond Trade Request #2	Testing the consequences of rejecting a trade request	After being directed to the 'Incoming Trade Requests Page' by clicking on the 'Incoming Trade Requests' button provided on the 'Main Page', tapping the reject button	The trade operation is successfully rejected and deleted from the incoming trade requests of the current user, and from the pending trade requests of the user who sent the request	Pass

1.20. Item Delete Functional Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Item Delete #1	Testing the consequences of delete item attempt	From 'Collection Page', navigating to multiple selection page for items.	The selected items are successfully deleted	Pass

1.21. Image Processing Test Cases

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Image Upload To Server	To test the functionality of client to server image upload	From “Main Page” navigate to “Scan Page” , click on the screen to take the image and send it to the server	Text “Upload successful” is shown in the client. In server computer, underneath /tmp folder, the image is saved.	Pass
Card Boundary Extraction On Server	To test the feature of extracting card boundaries in server	From “Main Page” navigate to “Scan Page” , click on the screen to take the image and send it to the server	Server will save the boundary highlighted image in /tmp folder with name “output.jpg”	Pass

1.22. Performance Tests

Test Case Identifier	Objective	Inputs	Output	Pass / Fail
Login Server Response Test	Under controlled circumstances , test the response of login feature from server	: Server must not be in the same network . Client must be connected to an internet with an upload speed higher than 512 Kbps and download speed higher than 2 Mbps. User tries to login with a true name and password.	User logins within 1 s.	Pass
Login Server Stress Test	Under controlled circumstances , test the response of login feature from server	Server must not be in the same network . Client must be connected to an internet with an upload speed higher than 512 Kbps and download speed higher than 2 Mbps.20 User tries to login with a true name and password	User logins within 3 s.	Pass
Boundary Extraction Response Test	Under controlled circumstances , test the response of the boundary extraction from server	Server must not be in the same network . Client must be connected to an internet with an upload speed higher than 512 Kbps and download speed higher than 2 Mbps. User sends a image to server to have it boundary extracted.	The boundary extracted image is saved to the /tmp folder of the server machine within 4 seconds.	Pass

Image Load From Server Stress Test	Under controlled circumstances , test the loading times of the images	Server must not be in the same network . Client must be connected to an internet with an upload speed higher than 512 Kbps and download speed higher than 2 Mbps. 20 users navigates to a card page that has not been cached at local database	Every user should see the card image within 5 seconds	Pass
Create Collection Server Stress Test	Under controlled circumstances , test the loading times of the images	Server must not be in the same network . Client must be connected to an internet with an upload speed higher than 512 Kbps and download speed higher than 2 Mbps. 20 users navigates to user collections page and tries to create a new collection.	Every user should see that the collection is created within 3 seconds	Pass