

Puzzle

Mert ÇALIK - 1679406

Oğuz Can KARTAL – 1746106

Berke Çağkan TOPTAŞ – 1746387

Burak VELİOĞLU - 1679257

SOFTWARE TEST DOCUMENTATION

Version 1.0

Table of Contents

- 1. Introduction..... 3
 - 1.1. Problem Definition 3
 - 1.2. Purpose 3
 - 1.3. Scope 3
- 2. Test Cases 4
 - 2.1. Register Functional Test Cases 5
 - 2.2. Login Functional Test Cases..... 8
 - 2.3. Add Album Functional Test Cases 9
 - 2.4. Delete Album Functional Test Cases 10
 - 2.5. Add Category Functional Test Cases 11
 - 2.6. Categorize Album Functional Test Cases..... 12
 - 2.7. Share Album Functional Test Cases..... 13
 - 2.8. Like Album Functional Test Cases 13
 - 2.9. Write Comment Functional Test Cases 14
 - 2.10. Search Album Functional Test Cases 15
- 3. Performance Test 23

1. Introduction

1.1. Problem Definition

People may be collecting audio albums because of their collecting hobby or their interest in music. Audio albums in physical form are often provided with decorative covers (cover art). Since there are numerous audio albums and a new album come onto the market nearly every week keeping record of these albums is not very easy. Therefore, it is critical to keep record of all the albums collectors have.

1.2. Purpose

The purpose of this document is to test the functionality and performance of the MobColl project. This document specifies how the application should react some actions and what the performance of the application should be while reacting some of these actions. This document verifies how the design should meet the requirements specified in the SRS document through design viewpoints. The design viewpoints will cover all design elements presented before. Moreover, in this document, test environment, required hardware and software for testing use cases will be specified.

1.3. Scope

This application will allow the users to keep record of all the albums they have.

- 1) User can register to the system.
- 2) User can login the system by using a previously registered account.
- 3) User can login the system by using a Facebook account.
- 4) User can add an album to the account.
- 5) User can delete a previously created album from the account.
- 6) User can create a category.
- 7) User can categorize his/her albums.
- 8) User can share an album through Facebook.
- 9) User can like an album.

- 10) User can write comment to album and can read comments of an album.
- 11) User can search an album through the system.
- 12) User can search other users through the system.
- 13) User can become friend with other users.
- 14) User can send message to his friends.
- 15) User can save the location of an album.

In this document, these functionalities of the MobColl Project are tested.

2. Test Cases

Test Number	Test Name
1	Register Functional Test Cases
2	Login Functional Test Cases
3	Add Album Functional Test Cases
4	Delete Album Functional Test Cases
5	Create Category Functional Test Cases
6	Categorize Album Functional Test Cases
7	Share Album Functional Test Cases
8	Like Album Functional Test Cases
9	Write Comment Functional Test Cases
10	Search Functional Test Cases
11	Friendship Functional Test Cases
12	Messaging Functional Test Cases
13	Album Location Functional Test Cases

In order to test these functional test cases, a working Android based smart phone with version 2.2 and internet connection are necessary.

2.1. Register Functional Test Cases

These test cases take place in “Register Page”.

Test ID	Objective	Inputs	Outcomes
Register-01	Test registering with all the required fields are filled with appropriate values.	Username : ercanK Password : ercan12 Confirm Password : ercan12 Name : Ercan Surname : Kaya Email : ercanK@hotmail.com Birth Date : 1990 03 07 Press “Register” button.	If this username is not already in use, this user is successfully registered.
Register-02	Test registering with duplicate username.	Username : ercanK Password : ercan12 Confirm Password : ercan12 Name : Ercan Surname : Kaya Email : ercanK@hotmail.com Birth Date : 1990 03 07 Press “Register” button.	“Username already exists!” error is displayed.

Register-03	Test registering with different "password" and "confirm password" fields.	Username : ercanK Password : ercan12 Confirm Password : ercan13 Name : Ercan Surname : Kaya Email : ercanK@hotmail.com Birth Date : 1990 03 07 Press "Register" button.	"Password Conflict" error is displayed.
Register-04	Test registering with duplicate e-mail.	Username : ercanK Password : ercan12 Confirm Password : ercan12 Name : Ercan Surname : Kaya Email : ercanK@hotmail.com Birth Date : 1990 03 07 Press "Register" button.	"This e-mail is already in use!" error is displayed.
Register-05	Test registering with invalid e-mail	Username : ercanK Password : ercan12 Confirm Password : ercan12 Name : Ercan Surname : Kaya Email : ercankaya.com Birth Date : 1990 03 07 Press "Register" button.	"Invalid Email Address!" error is displayed.

Register-06	Test registering with invalid birth date year	Username : ercanK Password : ercan12 Confirm Password : ercan12 Name : Ercan Surname : Kaya Email : ercanK@hotmail.com Birth Date : 2222 03 07 Press "Register" button.	"Invalid Birth Date!" error is displayed.
Register-07	Test registering with invalid birth date month	Username : ercanK Password : ercan12 Confirm Password : ercan12 Name : Ercan Surname : Kaya Email : ercanK@hotmail.com Birth Date : 1990 13 07 Press "Register" button.	"Invalid Birth Date!" error is displayed.
Register-08	Test registering with invalid birth date day	Username : ercanK Password : ercan12 Confirm Password : ercan12 Name : Ercan Surname : Kaya Email : ercanK@hotmail.com Birth Date : 1990 03 32 Press "Register" button.	"Invalid Birth Date!" error is displayed.

2.2. Login Functional Test Cases

These test cases take place in “Login Page”.

Test ID	Objective	Inputs	Outcomes
Login-01	Test logging in with all the required fields are filled with appropriate values.	Username : ercanK Password : ercan12 Press “Login” button.	User logs in to system and redirected to “Profile Page”.
Login-02	Test logging in with a username that does not exists.	Username : ercankaya Password : ercan12 Press “Login” button.	“Wrong Username/Password!” error is displayed.
Login-03	Test logging in with invalid password.	Username : ercanK Password : ercan13 Press “Login” button.	“Wrong Username/Password!” error is displayed.
Login-04	Test logging in with a Facebook account.	Press “Login with Facebook” button. Fill the fields according to a Facebook account.	If Facebook account is valid, user logs in to system and redirected to “Profile Page”.

2.3. Add Album Functional Test Cases

Test ID	Objective	Inputs	Outcomes
AddA-01	Test adding a new album by the barcode of the album.	In "Profile Page", click "Add Album" button from menu. Scan the barcode of the album.	The album is added to user account.
AddA-02	Test adding an already existing album by the barcode of the album.	In "Profile Page", click "Add Album" button from menu. Scan the barcode of the album.	"Album Already Added" error is displayed.
AddA-03	Test adding a new album from the "Album Page"	In "Album Page", click the cover photo of the album.	The album is added to user account.
AddA-04	Test adding an already existing album from the "Album Page"	In "Album Page", click the cover photo of the album.	"Album Already Added" error is displayed.

2.4. Delete Album Functional Test Cases

These test cases take place in "Profile Page".

Test ID	Objective	Inputs	Outcomes
Delete-01	Test deleting an album from the user account.	Select "All Albums" category Click "Delete Album" button from menu Select the album desired to be deleted from account.	Album is deleted from every category that the album exists and from user's account.
Delete-02	Test deleting an album from one of user's category.	Select the category that the album desired to be deleted Click "Delete Album" button from menu Select the album desired to be deleted from account.	Album is deleted from the category that the album desired to be deleted.

2.5. Add Category Functional Test Cases

These test cases take place in “Profile Page”.

Test ID	Objective	Inputs	Outcomes
AddC-01	Test adding a new category.	Category name : Jazz Press “Add” button.	“Jazz” category is added to the category list of the user.
AddC-02	Test adding an already existing category.	Category name : Jazz Press “Add” button.	“Category Already Added” error is displayed.
AddC-03	Test adding a category without name.	Category name : Press “Add” button.	“Invalid Category Name” error is displayed.

2.6. Categorize Album Functional Test Cases

These test cases take place in “Profile Page”.

Test ID	Objective	Inputs	Outcomes
Categorize-01	Test adding a new album to a category.	Press “Add Album to Category” button from the menu Select the category that the album desired to be added. Select a new album that desired to be added to that category.	The album is added to the desired category.
Categorize-02	Test adding an already existing album to a category.	Press “Add Album to Category” button from the menu Select the category that the album desired to be added. Select an already existing album that desired to be added to that category.	“Album Already Exists” error is displayed.

2.7. Share Album Functional Test Cases

These test cases take place in “Album Page”.

Test ID	Objective	Inputs	Outcomes
Share-01	Test sharing an album through Facebook account if user logged in with Facebook account.	Press “Share” button.	Album information is shared on Facebook account.
Share-02	Test sharing an album through Facebook account if user did not login with Facebook account.	Press “Share” button.	“Login with Facebook account” error is displayed.

2.8. Like Album Functional Test Cases

These test cases take place in “Album Page”.

Test ID	Objective	Inputs	Outcomes
Like-01	Test liking an album which has not been liked before.	Press “Like” button.	Album like count is incremented and this information is added to user information.
Like-02	Test liking an album which has been liked before.	Press “Like” button.	Album like count stays the same.

2.9. Write Comment Functional Test Cases

These test cases take place in “Comment Page”.

Test ID	Objective	Inputs	Outcomes
Comment-01	Test writing a comment with appropriate length.	Comment: Very good album. Press “Send Comment” button.	Comment is successfully added to the comment list of this album.
Comment-02	Test writing a comment with non-appropriate length.	Comment length > 150 Press “Send Comment” button.	“Message is too long” error is displayed.
Comment-03	Test writing an empty comment.	Comment : Press “Send Comment” button.	“Invalid Comment” error is displayed.

2.10. Search Album Functional Test Cases

These test cases take place in "Search Page".

Test ID	Objective	Inputs	Outcomes
Search-01	Test searching an album which exists in database by the barcode of the album.	Select "Barcode" tab from search page Press "Search" button. Scan the barcode of the album.	Desired album is shown in results.
Search-02	Test searching an album which does not exist in database by the barcode of the album.	Select "Barcode" tab from search page Press "Search" button. Scan the barcode of the album.	Empty list is displayed.
Search-03	Test searching an album which exists in database by the name of the album.	Select "Album" tab from search page Album Name: Overexposed Press "Search" button.	Albums which contain the searched string in their name are shown in results.
Search-04	Test searching an album which does not exist in database by the name of the album.	Select "Album" tab from search page Album Name: Ovedads Press "Search" button.	Empty list is displayed.

Search-05	Test searching an album by album name with empty string.	Select "Album" tab from search page Album Name: Press "Search" button.	"Invalid Album Name" error is displayed.
Search-06	Test searching an album in database by the artist name of the album which exists.	Select "Artist" tab from search page Artist Name: Rihanna Press "Search" button.	Albums which contain the searched string in their artist name are shown in results.
Search-07	Test searching an album in database by the artist name of the album which does not exist.	Select "Artist" tab from search page Artist Name: Overasdads Press "Search" button.	Empty list is displayed.
Search-08	Test searching an album by artist name with empty string.	Select "Artist" tab from search page Artist Name: Press "Search" button.	"Invalid Artist Name" error is displayed.

Search-09	Test searching a user in database by the name of the user who exists.	<p>Select "User" tab from search page</p> <p>User Name: Mert</p> <p>Press "Search" button.</p>	Users who contain the searched string in their name are shown in results.
Search-10	Test searching a user in database by name of the user who does not exist.	<p>Select "User" tab from search page</p> <p>User Name: Merasas</p> <p>Press "Search" button.</p>	Empty list is displayed.
Search-11	Test searching a user by name with empty string.	<p>Select "User" tab from search page</p> <p>User Name:</p> <p>Press "Search" button.</p>	"Invalid User Name" error is displayed.

2.11. Friendship Functional Test Cases

Test ID	Objective	Inputs	Outcomes
Friendship-01	Test displaying 'Add Friend' button.	Press the name of a user who is not a friend.	'Add Friend' button should be displayed in the page.
Friendship-02	Test displaying 'Friendship request has sent.' Message.	Press the name of a user who is not a friend, but a friendship request has been sent.	'Friendship request has sent' message should be displayed in the page.
Friendship-03	Test displaying friendship requests.	Select 'Requests' tab from Friends Page.	Names of users who have sent a friendship request are displayed.
Friendship-04	Test accepting a friendship request.	Press the name of a user in 'Requests' tab in Friends Page. Press 'Yes'.	Name of the user is deleted from 'Requests' tab and added to 'Friends' tab and they become friend.

Friendship-05	Test rejecting a friendship request.	Press the name of a user in 'Requests' tab in Friends Page. Press 'No'.	Name of the user is deleted from 'Requests' tab.
Friendship-06	Test displaying friends.	Select 'Friends' tab from Friends Page.	Names of the friends are displayed.
Friendship-07	Test displaying the albums of a friend.	Press the name of a user in 'Friends' tab in Friends Page. Press 'Albums' Button.	Albums of the friend are displayed.
Friendship-08	Test displaying the friends of a friend.	Press the name of a user in 'Friends' tab in Friends Page. Press 'Friends' Button.	Friends of the friend are displayed.

2.12. Messaging Functional Test Cases

Test ID	Objective	Inputs	Outcomes
Messaging-01	Test going to Send Message Page from a friend.	Press the name of the friend in the Friend Info Page.	User is directed to the Send Message Page.
Messaging-02	Test going to Send Message Page from a user who is not friend.	Press the name of the user who is not friend in the Friend Info Page.	User is not directed to the Send Message Page and stay in Friend Info Page.
Messaging-03	Test sending message to a friend.	In the Send Message Page, write the message. Press 'Send Message' button.	Message is sent to friend.
Messaging-04	Test reading incoming messages.	In the User Page, press the 'Messages' button. Select the 'Inbox' tab.	The messages that are sent to the user are displayed.
Messaging-05	Test reading outgoing messages.	In the User Page, press the 'Messages' button. Select the 'Outbox' tab.	The messages that are sent from the user are displayed.

2.13. Album Location Functional Test Cases

Test ID	Objective	Inputs	Outcomes
Location-01	Test going to Location Page of a not owned album.	In Album Page press 'Info' button.	'Album is not owned!' message is displayed.
Location-02	Test going to Location Page of an already owned album.	In Album Page press 'Info' button.	User is directed to the Location Page.
Location-03	Test displaying 'Write Information Here'.	In the Album Page of an album of which location has not been saved before, press 'Info' button.	User is directed to the Location Page. 'Write Information Here' hint is written in textbox.
Location-04	Test displaying location of the album.	In the Album Page of an album of which location has been saved before, press 'Info' button.	User is directed to the Location Page. Previously saved location of album is written in textbox.

Location-05	Test updating location of the album.	In the Album Page of an album of which location has been saved before, press 'Info' button. Change the location of the album. Press 'Update' button.	Location of album is updated.
-------------	--------------------------------------	--	-------------------------------

3. Performance Test

Test ID	Objective	Inputs	Outcomes
Performance-01	Providing login service to 1 user has 1Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 200ms.
Performance-02	Providing login service to 5 users have 1Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 300ms.
Performance-03	Providing login service to 20 users have 1Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 500ms.
Performance-04	Providing login service to 50 users have 1Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 1s.
Performance-05	Providing login service to 1 user has 8Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 100ms.

Performance-06	Providing login service to 5 users have 8Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 150ms.
Performance-07	Providing login service to 20 users have 8Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 250ms.
Performance-08	Providing login service to 50 users have 8Gbit internet.	Login to system(both with program interface and Facebook)	Login to system within 400ms.
Performance-09	Providing login service to 1 user has 3G internet.	Login to system(both with program interface and Facebook)	Login to system within 200ms.
Performance-10	Providing login service to 5 users have 3G internet.	Login to system(both with program interface and Facebook)	Login to system within 300ms.

Performance-11	Providing login service to 20 users have 3G internet.	Login to system(both with program interface and Facebook)	Login to system within 500ms.
Performance-12	Providing login service to 50 users have 3G internet.	Login to system(both with program interface and Facebook)	Login to system within 1s.
Performance-13	Providing image load service to 1 user has 1Gbit internet.	Try to view album page	Get album page within 250ms.
Performance-14	Providing image load service to 5 users have 1Gbit internet.	Try to view album page	Get album page within 500ms.
Performance-15	Providing image load service to 20 users have 1Gbit internet.	Try to view album page	Get album page within 800ms.

Performance-16	Providing image load service to 50 users have 1Gbit internet.	Try to view album page	Get album page within 1s.
Performance-17	Providing image load service to 1 user has 8Gbit internet.	Try to view album page	Get album page within 150ms.
Performance-18	Providing image load service to 5 users have 8Gbit internet.	Try to view album page	Get album page within 300ms.
Performance-19	Providing image load service to 20 users have 8Gbit internet.	Try to view album page	Get album page within 400ms.
Performance-20	Providing image load service to 50 users have 8Gbit internet.	Try to view album page	Get album page within 800ms.

Performance-21	Providing image load service to 1 user has 3G internet.	Try to view album page	Get album page within 250ms.
Performance-22	Providing image load service to 5 users have 3G internet.	Try to view album page	Get album page within 500ms.
Performance-23	Providing image load service to 20 users have 3G internet.	Try to view album page	Get album page within 800ms.
Performance-24	Providing image load service to 50 users have 3G internet.	Try to view album page	Get album page within 1s.