

Software Requirements Specification

Version 1.0

September 19, 2014



Digital Photo Organizing Software

Prepared by:
Ranon Martin
rm1905@nova.edu

Prepared for:
CISC 680 – Software Engineering
Instructor: Frank J. Mitropoulos, Ph.D.
Fall 2014

i. Revision History

Date	Description	Author	Comments

ii. Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Ranon Martin	Lead Software Eng.	
	Dr. Frank J. Mitropoulos	Professor, CISC 680	

Table of Contents

i. Revision History	2
ii. Document Approval	3
1 Introduction	6
1.1 Purpose	7
1.2 Scope	7
1.3 Glossary.....	7
2 User Requirements Definition.....	10
2.1 Functional Requirements.....	11
2.1.1 <i>Photo organization</i>	11
2.1.2 <i>Photo viewing</i>	13
2.1.3 <i>Meta-data management</i>	13
2.1.4 <i>Photo sharing</i>	14
2.1.5 <i>Preference configuration</i>	15
2.1.6 <i>Search</i>	16
2.1.7 <i>Basic photo editing</i>	16
2.2 Use-cases and Scenarios.....	17
2.2.1 <i>Use-case diagram: High level view</i>	17
2.2.2 <i>Base use-case descriptions</i>	23
2.3 Non-Functional Requirements.....	80
2.3.1 <i>Compatibility</i>	80
2.3.2 <i>Reliability</i>	80
2.3.3 <i>Correctness</i>	81
2.3.4 <i>Implementation</i>	81
2.3.5 <i>Other Non-functional requirements</i>	82
2.3.6 <i>Reusability</i>	84
3 Requirements Prototyping	86
3.1 Photo organization	87
3.1.1 <i>Prototype 1.1: Create root photo album</i>	87
3.1.2 <i>Prototype 1.2: Empty slate – no photo album exists</i>	88

3.1.3 Prototype 1.3: Create new photo album	89
3.1.4 Prototype 1.4: Empty slate – no photo exists.....	90
3.1.5 Prototype 1.5: Add a photo to an existing photo album	91
3.1.6 Prototype 1.6: Root photo album with content.....	92
3.1.7 Prototype 1.7: Photo album with content and multiple photo selection	93
3.1.8 Prototype 1.8: Photo album with content and single photo selection	94
3.1.9 Prototype 1.9: Copy photo.....	95
3.1.10 Prototype 1.10: Move photo.....	96
3.1.11 Prototype 1.11: Delete photo	97
3.1.12 Prototype 1.12: Photo album with content and single photo album selection.	98
3.1.13 Prototype 1.13: Move photo album.....	99
3.1.14 Prototype 1.14: Rename photo album	100
3.1.15 Prototype 1.15: Delete photo album	101
3.2 Photo viewing	102
3.2.1 Prototype 2.1: View a single photo.....	102
3.2.2 Prototype 2.2: View an entire photo album of photos as a slideshow	103
3.2.3 Prototype 2.3: Slideshow dialog.....	104
3.3 Photo sharing	105
3.3.1 Prototype 3.1: Send a single photo as an email.....	105
3.3.2 Prototype 3.2: Send an entire photo album of photos as an email	106
3.4 Preference configuration	107
3.4.1 Prototype 4.1: Email setup	107
3.4.2 Prototype 4.2: Modify slideshow settings	108
3.5 Search	109
3.5.1 Prototype 5.1 Search results	109
3.6 Basic photo editing.....	110
3.6.1 Prototype 6.1 Edit photo image	110
4 Risk Analysis	111

1 Introduction

This section gives a description of the project scope and an overview of this Software Specification (SRS) document. The purpose of this document is also outlined and a glossary with a list of abbreviations and definitions is provided.

1.1 Purpose

This document aims to give a detailed description of the requirements for PixBox, the digital photo organizing software. It will illustrate the purpose and the complete declaration for the development of the software. The primary intent of this document is for the client's approval and for the development of the software.

1.2 Scope

The scope of this project is to develop a single user, standalone digital photo organizing software that should give the user the ability to:

- Manage photos and photo albums
- Search for a photo or photos by using a keyword or group of comma-separated keywords.
- Share one photo or an entire album of photographs via email without the use of a third-party solution.
- View a single photo or a slideshow of photos from a particular photo album.
- Configure the settings for the email and slideshow globally.
- Carryout basic photo editing techniques on photos without the use of a third-party software.

1.3 Glossary

Term	Definition
User	A person who interacts with the digital photo organizing software.
End-user	Synonym for User
PixBox	The digital photo organizing software that is to be developed.

SRS	Software Requirements Specification.
Photo	An abstraction for an image uploaded by the user.
Photo album	A container for holding the photos to keep them organized.
Album	Synonym for Photo album.
Software	The digital photo organizing software that is to be developed.
Keyword	A word that
JPEG	An image file type.
PNG	An image file type.
Sub photo album	An album that exists within an album.
Root photo album	The album that is the container for all the albums and photos for the software.
Adjacent photo album	An album that has the same direct root album.
Relative root photo album	An album that contains the album or photo being referenced.
Slideshow	A presentation of a series of photos existing in the album specified.
Slideshow properties	The slideshow duration and replay loop count.
Slideshow transition duration	The amount of time in seconds that a photo will be displayed for before the next photo is displayed.
Slideshow replay loop count	The number of times the entire slideshow will replay for.
Email settings	Settings that should exist for the email feature to function properly. Settings include: email address, password and outgoing mail server configurations.
Email fields	Email fields include: recipient's email address, email subject and message.
Email attachment	The photo or photos that will be attached to the

	email to be sent.
Search results	Photos that match the keyword or keywords specified in the search component.

2 User Requirements Definition

This section describes the specific requirements for the software project by covering all required functional and non-functional requirements. Various use cases were also illustrated, communicating the interactions necessary for a user to achieve the stated goals.

2.1 Functional Requirements

This section describes the specific features of the software project.

2.1.1 Photo organization

2.1.1.1 *Functional requirement 1.1*

The user shall be able to create a photo album.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.1.2 *Functional requirement 1.2*

The user shall be able to move a photo album from one relative root photo album to another.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.1.3 *Functional requirement 1.3*

The user shall be able to modify the name of a photo album.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.1.4 *Functional requirement 1.4*

The user shall be able to delete one or more selected photo albums.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.1.5 *Functional requirement 1.5*

The user shall be able to add a photo to an existing album.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.1.6 *Functional requirement 1.6*

The user shall be able to delete one or more selected photos.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.1.7 *Functional requirement 1.7*

The user shall be able to move one or more selected photos from one album to another.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.1.8 *Functional requirement 1.8*

The user shall be able to copy one or more selected photos from one album to another.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.2 Photo viewing

2.1.2.1 Functional requirement 2.1

The user shall be able to view a single photo.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.2.2 Functional requirement 2.2

The user shall be able to view an entire album of photos as a slideshow.

Priority: medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.2.3 Functional requirement 2.3

The user shall be able to replay the slideshow when it is finish.

Priority: medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.2.4 Functional requirement 2.4

The user shall be able to stop the slideshow at any time while it is playing.

Priority: medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.3 Meta-data management

2.1.3.1 Functional requirement 3.1

The user shall be able to add a comma-separated list of keywords to a photo.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.3.2 Functional requirement 3.2

The user shall be able to change the keywords associated with a photo.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.3.3 Functional requirement 3.3

The user shall be able to delete one or more keywords associated with a photo.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.3.4 Functional requirement 3.4

The user shall be able to view the keywords associated with a photo.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.4 Photo sharing

2.1.4.1 Functional requirement 4.1

The user shall be able to send one photo via email to a recipient.

Priority: Medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.4.2 Functional requirement 4.2

The user shall be able to send an entire photo album of photos via email to a recipient.

Priority: medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.5 Preference configuration

2.1.5.1 Functional requirement 5.1

The user shall be able to configure the email settings globally.

Priority: medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.5.2 Functional requirement 5.2

The user shall be able to configure the slideshow duration globally.

Priority: medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.5.3 Functional requirement 5.3

The user shall be able to configure the slideshow replay loop count globally.

Priority: medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.6 Search

2.1.6.1 *Functional requirement 6.1*

The user shall be able to search for a photo by using a keyword or a group of comma-separated keywords.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.7 Basic photo editing

2.1.7.1 *Functional requirement 7.1*

The user shall be able to convert a photo to black and white.

Priority: Medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.7.2 *Functional requirement 7.2*

The user shall be able to change the contrast of a photo.

Priority: Medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.7.3 *Functional requirement 7.3*

The user shall be able to change the brightness of a photo.

Priority: Medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.1.7.4 Functional requirement 7.4

The user shall be able to rotate a photo.

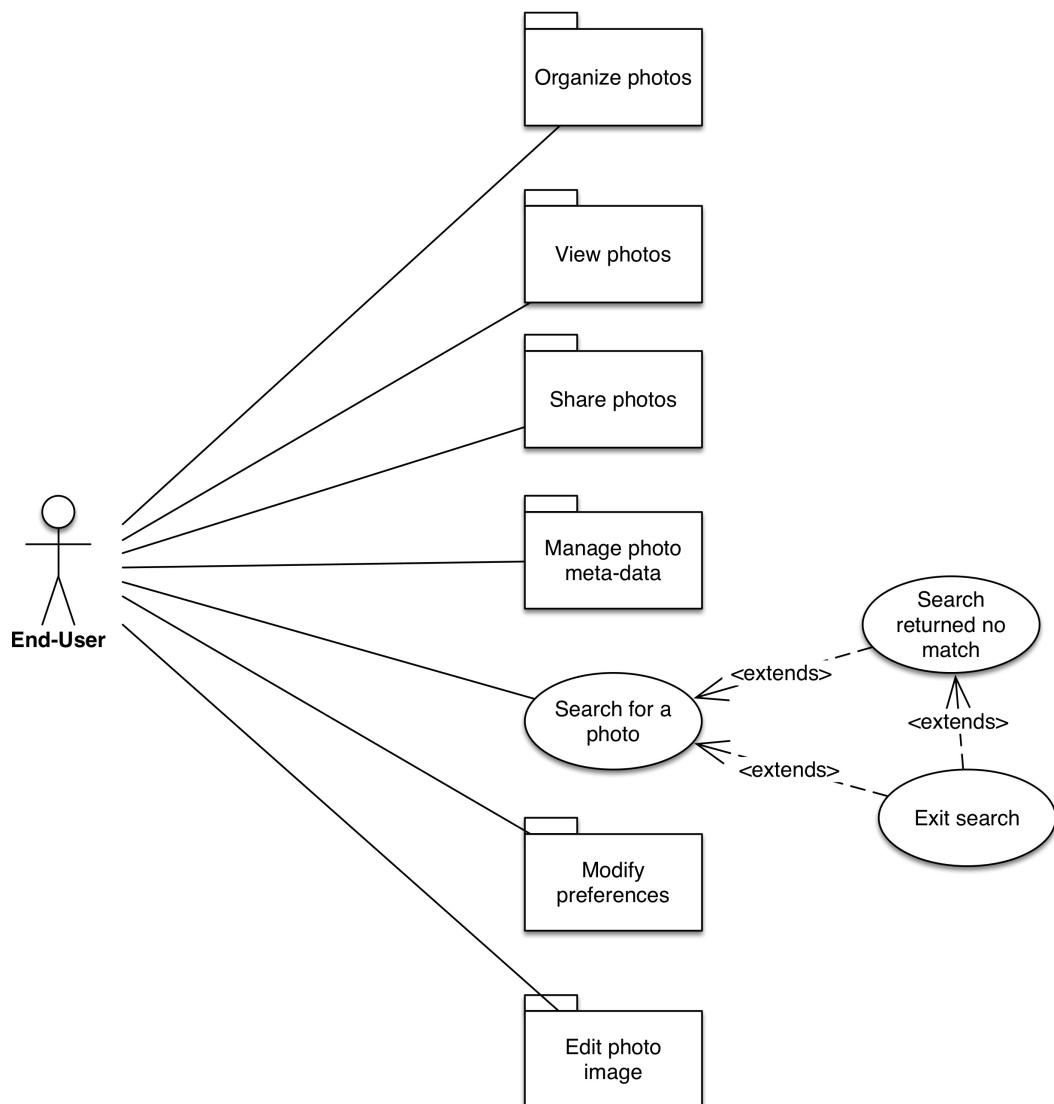
Priority: Medium

Source: Dr. Mitropoulos

Date: August 27, 2014

2.2 Use-cases and Scenarios

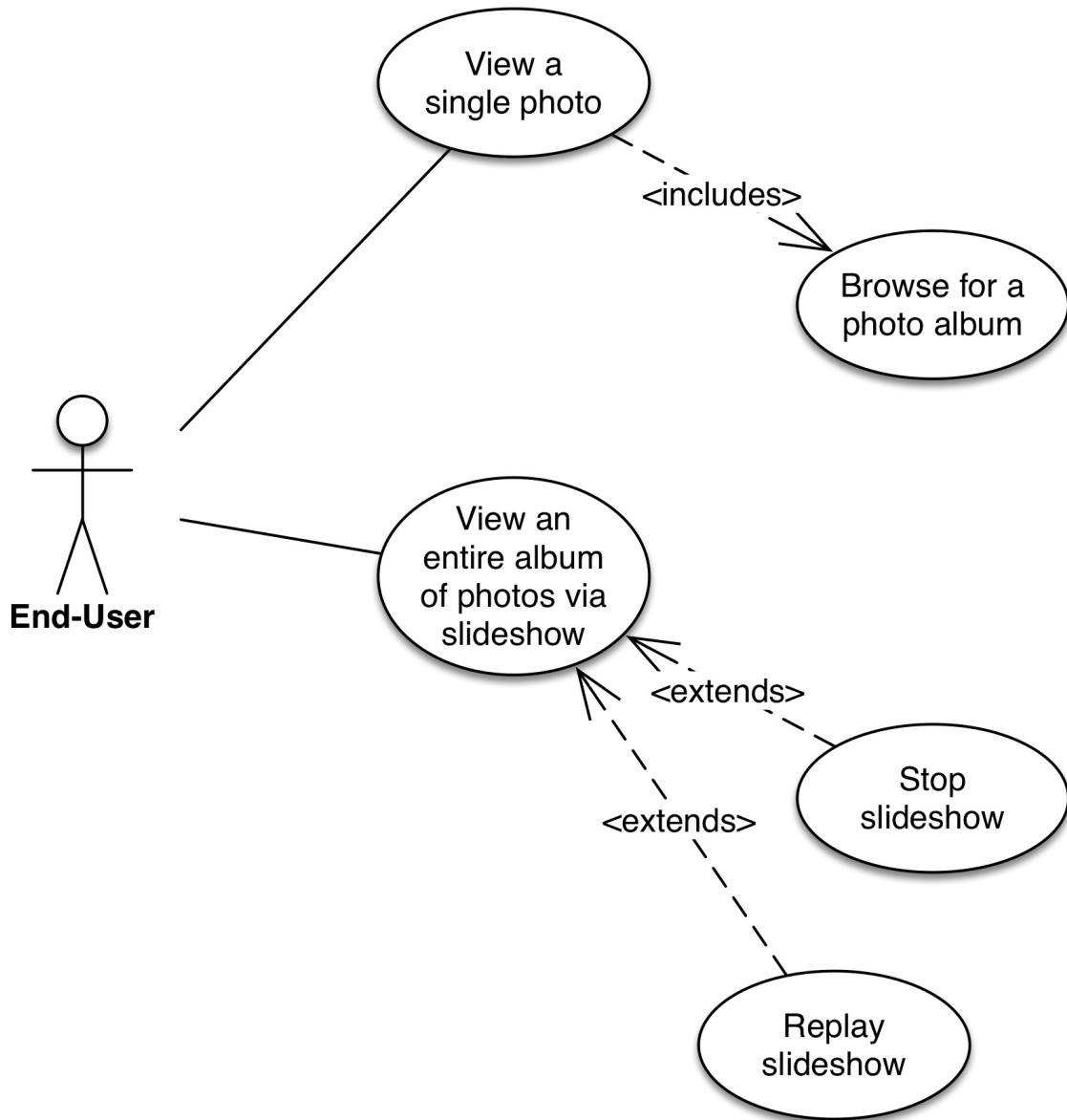
2.2.1 Use-case diagram: High level view



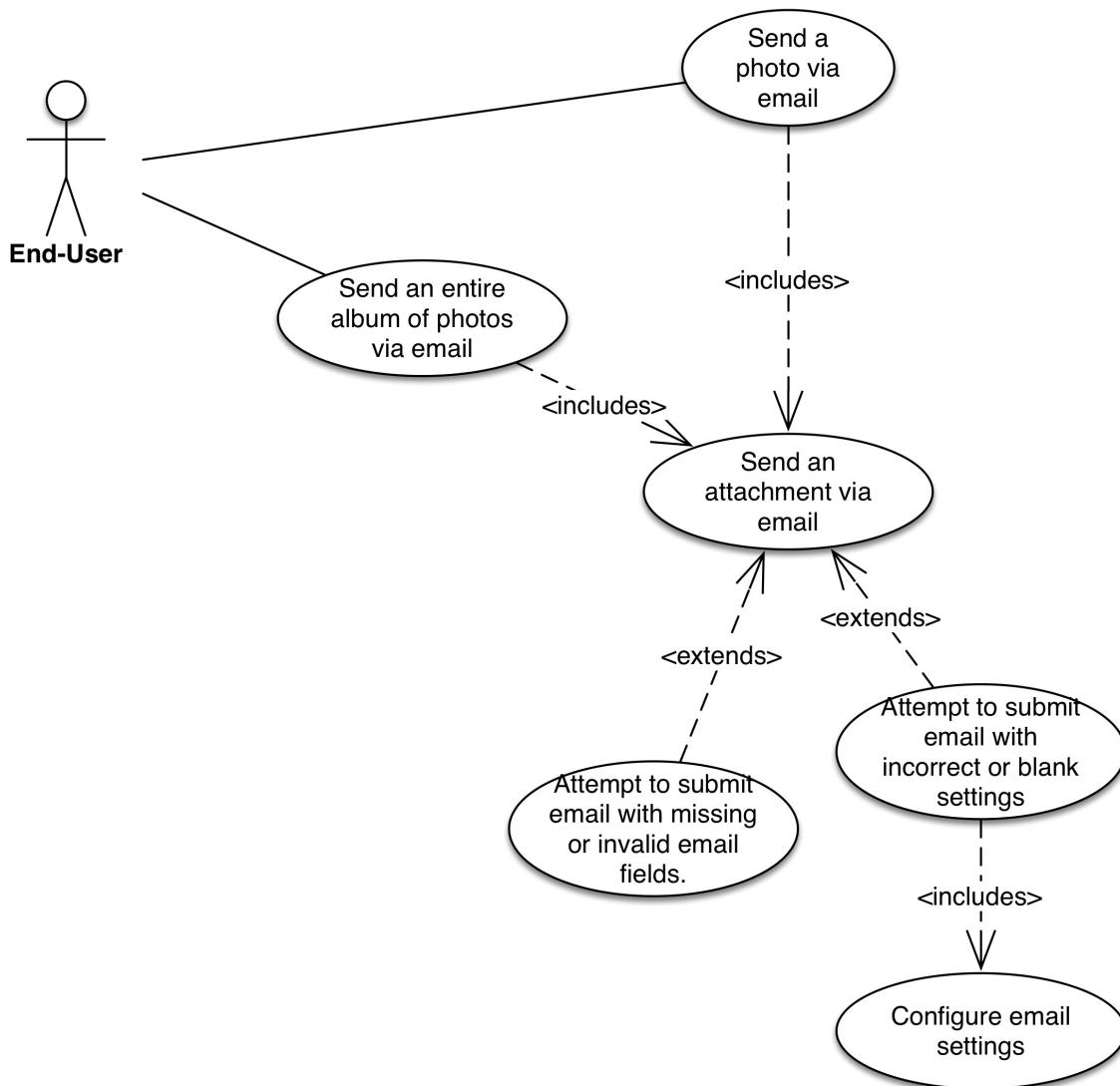
2.2.1.1 Use-case package 1: Organize photos



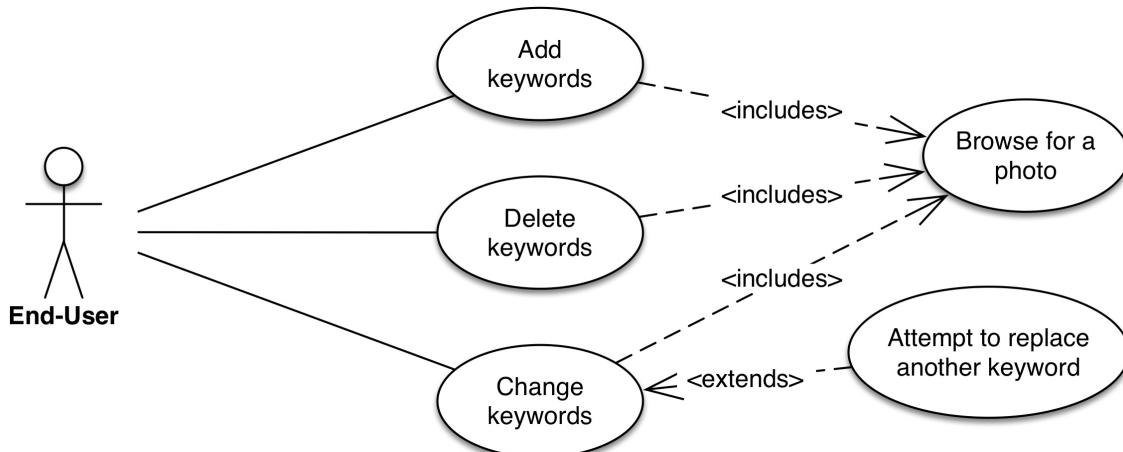
2.2.1.2 Use-case package 2: View photos



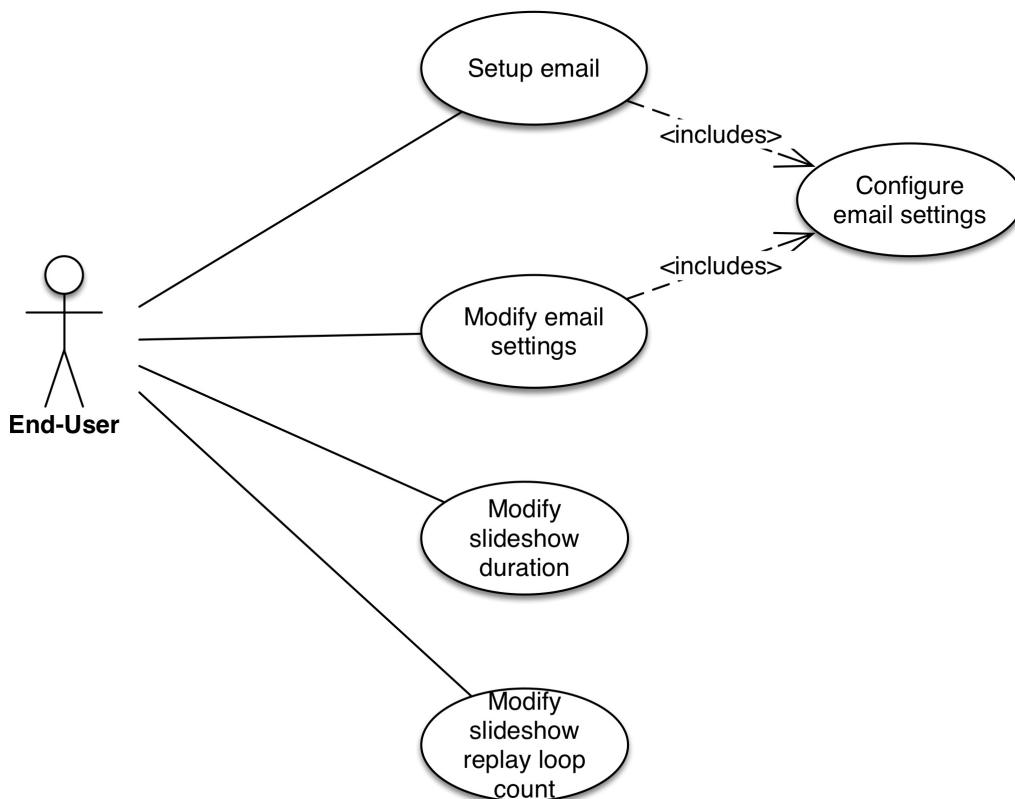
2.2.1.3 Use-case package 3: Share photos



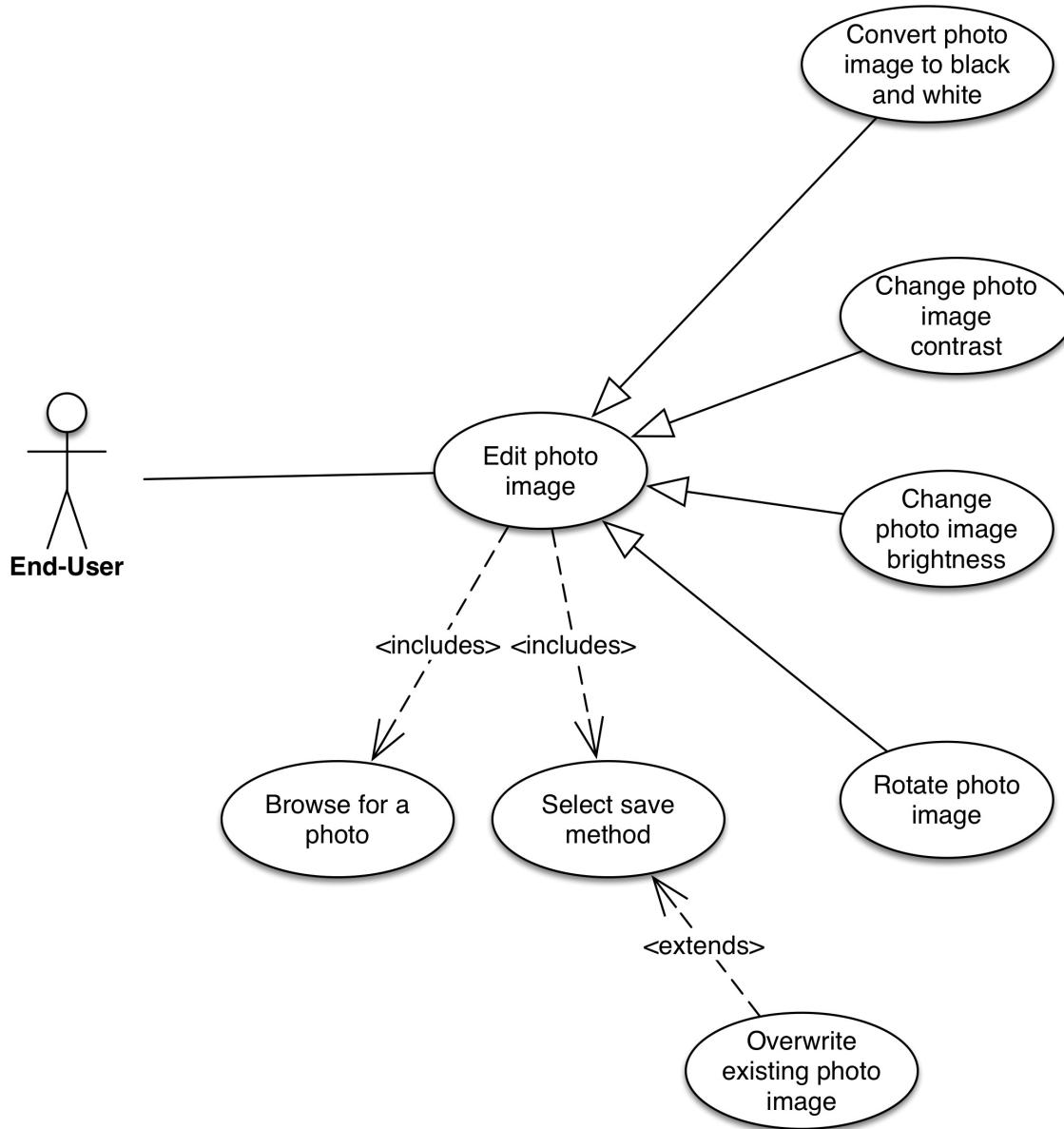
2.2.1.4 Use-case package 4: Manage photo meta-data



2.2.1.5 Use-case package 5: Modify preferences



2.2.1.6 Use-case package 6: Edit photo image



2.2.2 Base use-case descriptions

2.2.2.1 Use-case 1.1

Create photo album

Actor:

End-user

Goal:

To create a photo album to organize photos

Related use-

Generalization of:

cases:

- Use-case 1.1.1 ‘Create root photo album’
- Use-case 1.1.2 ‘Create a photo album’

Priority:

High

Brief description:

The end-user creates a photo album.

Preconditions:

The end-user already has the software open.

Flow of events:

	User actions	System responses
	1. Choose to create a new photo album.	2. Prompts end-user to specify photo album name.
	3. Specify photo album name	
	4. Submit	5. Prompt disappears.

Post conditions: A photo album is created.

Extensions: N/A

Non-behavioral requirements: N/A

Assumptions: N/A

Issues: N/A

Source: Functional requirement 1.1

2.2.2.2 Use-case 1.1.1

Create root photo album

Actor:	End-user	
Goal:	To create a root photo album when the software is started for the first time.	
Related use-cases:	Specialization of: Use-case 1.1 ‘Create photo album’	
Priority:	High	
Brief description:	The end-user is prompted to create the root photo album when the software is first started.	
Preconditions:	<ul style="list-style-type: none"> • The software is started for the first time. • There are no existing albums or photos 	
Flow of events:	User actions 1. Specify photo album name 2. Submit	System responses 1. Prompts end-user to specify root photo album name. 2. Prompt disappears.
Post conditions:	The root photo album is created.	
Extensions:	N/A	
Non-behavioral requirements:		
Assumptions:	N/A	
Issues:	N/A	
Source:	Functional requirement 1.1	

2.2.2.3 Use-case 1.1.2

Create a photo album

Actor:	End-user	
Goal:	To create a photo album to organize photos	
Related use-cases:	Specialization of: Use-case 1.1 ‘Create photo album’	
Priority:	High	
Brief description:	The end-user creates a photo album inside the root photo album.	
Preconditions:	The root photo album exists.	
Flow of events:	User actions <ol style="list-style-type: none"> 1. Choose to create a new photo album. 3. Specify photo album name 4. Submits photo album 	System responses <ol style="list-style-type: none"> 2. Prompts end-user to specify photo album name. 5. Prompt disappears.
Post conditions:	A new photo album is created.	
Extensions:	<p>3a. <u>Photo album name already exists in the same relative root photo album:</u> Use-case 1.1.3 ‘Attempt to give a photo album a name that exists’.</p>	
Non-behavioral requirements:	<ol style="list-style-type: none"> 1. The software shall allow any photo album to contain one or more photo albums. 2. The user shall not be able to create photo albums of the same name in the same root album relative to it. 	
Assumptions:	The end-user is currently in the relative root photo album	

of the desired photo album.

Issues: N/A

Source: Functional requirement 1.1

Use-case 1.1.3

Attempt to give a photo album a name that exists

Actor: End-user

Goal: To ensure that the photo album has a unique name within its relative root photo album.

Related use-cases: Extension of:

- Use-case 1.1.2 ‘Create a photo album’
- Use-case 1.4 ‘Rename a photo album’
- Use-case 1.5 ‘Move a photo album’

Priority: High

Brief description: The end-user is prompted to re-enter the photo album name, if the previously entered name already exist for an adjacent photo album.

Preconditions: An attempt was made to include a desired photo album in a relative root photo album where there is a photo album name the desired photo album’s name already exists.

Flow of events:	User actions	System responses
		1. Prompts end-user to specify photo album name.
	2. Specifies photo album name	3. System indicates that a photo album with the name specified already

exists.

- 4. Corrects the photo album name
- 5. Prompt disappears.

Post conditions: The photo album has a unique name within its relative root photo album.

Extensions: N/A

Non-behavioral requirements: Each photo album shall have a unique name within its relative root photo album.

Assumptions: N/A

Issues: N/A

Source:

Use-case 1.2

Add a photo

Actor: End-user

Goal: To add a photo an existing photo album

Related use-cases: N/A

Priority: High

Brief description: The end-user adds a photo with optional meta-data to an existing photo album.

Preconditions: The end-user is in the desired photo album, which is not the root album.

Flow of events:	User actions	System responses
	<ul style="list-style-type: none"> 1. Choose to add a photo to the current album. 	<ul style="list-style-type: none"> 2. The 'Add a photo' window appears. 3. Prompts the user to

upload a photo.

4. Chooses 'Browse...' command.
5. File open dialog appears.
6. **Include Use-case**
 - 1.2.3 'Browse for photo file'**
7. Confirm selection
8. Dialog disappears
9. Submits photo
10. The Add a photo window disappears.

Post conditions: A new photo with optional keywords and name is added to the desired photo album.

Extensions:

- 9a. If the keywords input box is selected: Use-case 1.2.1 'Add keywords when adding photo'. Rejoin at 9.
- 9b. If the name input box is selected: Use-case 1.2.2 'Add a name when adding photo'. Rejoin at 9.

Assumptions: N/A

Issues: N/A

Source: Functional requirement 1.5

2.2.2.4 Use-case 1.2.1

Add keywords when adding photo

Actor: End-user

Goal: To add a comma-separated list of keywords to the specified photo.

Related use-cases: Extension for: Use-case 1.2 'Add photo'

Priority:	High	
Brief description:	The end-user associates a comma-separated list of keywords to a photo that is being added to an album.	
Preconditions:	The desired photo to associate the keywords with is in the process of being added.	
Flow of events:	User actions	System responses
	<ol style="list-style-type: none"> 1. Select the keywords input box. 2. Specify the comma-separated list of keywords. 	
Post conditions:	The desired photo is associated with the keywords or keyword specified.	
Assumptions:	N/A	
Issues:	N/A	
Source:	Functional requirement 3.1	

2.2.2.5 Use-case 1.2.3

Browse for photo file

Actor:	End-user	
Goal:	To browse for a photo file	
Related use-cases:	N/A	
Priority:	High	
Brief description:	The end-user browses for a photo file of type JPEG or PNG.	
Preconditions:	The 'open file' dialog is open.	
Flow of events:	User actions	System responses

1. Select a directory if the desired file is not displayed.
2. Display the contents of the directory.
3. Repeat step 1 until the desired file is displayed.
4. Select the desired photo file.

Post conditions: The desired photo file is selected.

Extensions: N/A

Non-behavioral requirements: The software shall support JPEG and PNG file formats for photos.

Issues: N/A

Source: N/A

2.2.2.6 Use-case 1.3

Delete a photo album

Actor: End-user

Goal: To delete an existing photo album

Related use-cases: N/A

Priority: High

Brief description: The end-user deletes an existing photo album along with the included photos.

Preconditions: The desired photo album to be deleted already exists.

Flow of events:	User actions	System responses
------------------------	---------------------	-------------------------

1. Selects the desired
2. Display a list of

- photo album. possible commands.
3. Chooses ‘Delete...’ command.
4. Prompt the user to confirm request:
- Include use-case**
- 1.3.1 ‘Confirm destructive operation’**
5. The desired photo album disappears
6. Display success dialog.
7. Closes dialog.
8. Dialog disappears.

Post conditions: The desired photo album along with the included photos is deleted.

Extensions: N/A

Non-behavioral requirements: N/A

Assumptions: The end-user is currently in the relative root photo album of the desired photo album.

Issues: N/A

Source: Functional requirement 1.4

Use-case 1.3.1

Confirm destructive operation

Actor:

End-user

Goal:

To confirm a desired destructive operation

Related use-

N/A

cases:

Priority: High

Brief description: The end-user confirms a destructive operation.

Preconditions: A desired destructive operation is requested.

Flow of events:

User actions

System responses

1. Confirmation dialog appears and informs the user of the possible outcome.
2. Confirms operation
3. Confirmation dialog disappears

Post conditions: The desired destructive operation is confirmed.

Extensions: 2a. If the destructive operation is canceled: Use-case 1.3.2 'Abort destructive operation'. Rejoin at 3.

Non-behavioral requirements: The user shall be able to confirm or cancel destructive operations.

Issues: N/A

Source: N/A

2.2.2.7 Use-case 1.3.2

Abort destructive operation

Actor:	End-user
Goal:	To abort the previously requested destructive operation
Related use-cases:	Extension of: Use-case 1.3.1 ‘Confirm destructive operation’
Priority:	High
Brief description:	The end-user aborts a previously requested destructive operation.
Preconditions:	The system requests confirmation for a desired destructive operation is requested

Flow of events:	User actions	System responses
		1. Confirmation dialog appears. Indicates what the operation is going to do, and the operation cannot be undone.
	2. Cancel the operation	3. Confirmation dialog disappears
Post conditions:	Aborts the requested destructive operation.	
Extensions:	N/A	
Non-behavioral requirements:	N/A	
Issues:	N/A	
Source:	N/A	

2.2.2.8 Use-case 1.4

Rename a photo album

Actor: End-user
Goal: To change the name of a desired photo album
Related use-cases: N/A
Priority: High
Brief description: The end-user changes the name of a desired photo album that is already in existence.
Preconditions: The desired photo album exists.

Flow of events:	User actions	System responses
	1. Selects the desired photo album.	2. Display the list of possible commands.
	3. Chooses ‘Rename...’ command.	4. Prompts end-user to specify photo album name.
	5. Specifies the photo album name.	
	6. Submits the specified photo album name.	7. Desired photo album has the name specified.
		8. Prompt disappears
		9. Displays success dialog.
	10. Closes dialog.	11. Dialog disappears.
Post conditions:	Changed the name of the desired photo album to the name specified.	
Extensions:	<u>6a. If the name specified already exists within the same</u>	

relative root photo album: Use-case 1.1.3 ‘Attempt to give a photo album a name that exists’. Rejoin at 7.

Non-behavioral requirements:	N/A
Assumptions:	The end-user is currently in the relative root photo album of the desired photo album.
Issues:	N/A
Source:	N/A

2.2.2.9 Use-case 1.5

Move a photo album

Actor:	End-user
Goal:	To move a desired photo album from one photo album to another
Related use-cases:	N/A
Priority:	High
Brief description:	The end-user moves a desired photo album from one photo album to another.
Preconditions:	<ul style="list-style-type: none">• The desired photo album exists.• The desired new relative root photo album exists.

Flow of events:	User actions	System responses
	1. Selects the desired photo album. 3. Chooses the ‘Move...’ command. 5. Specify new relative root photo album: Include use-case 1.5.1 ‘Browse for a photo album’ 6. Confirm photo album selected	2. Displays a list of possible commands. 4. Photo album select dialog appears. 7. The desired photo album disappears from the current photo album. 8. Dialog disappears. 9. Displays success dialog. 11. Dialog disappears.
Post conditions:	Moves the desired photo album to the relative root photo album specified.	
Extensions:	<u>6a. If a photo album exists in the photo album selected that has the same name of the desired photo album to be moved:</u> Use-case 1.1.3 ‘Attempt to give a photo album a name that exists’. Rejoin at 7.	
Non-behavioral requirements:	N/A	
Assumptions:	The end-user is currently in the relative root photo album of the desired photo album.	

Issues: N/A
Source: Functional requirement 1.2

2.2.2.10 Use-case 1.5.1

Browse for a photo album

Actor: End-user
Goal: To browse for a desired photo album
Related use-cases: N/A
Priority: High
Brief description: The end-user browses for a desired photo album.
Preconditions: Photo albums and photos exist in the software.

Flow of events:	User actions	System responses
	1. Select a photo album if the desired photo album is not displayed.	2. Display the contents of the photo album.
	3. Repeat step 1 until the desired photo album is displayed.	
	4. Select a photo album.	

Post conditions: The desired photo album is found.
Extensions: N/A
Non-behavioral requirements: N/A
Issues: N/A

Source: N/A

2.2.2.11 Use-case 1.6

Delete photo

Actor: End-user

Goal: To delete one or more desired photos from a photo album

Related use-
cases: N/A

Priority: High

Brief description: The end-user deletes one or more desired photos from a photo album.

Preconditions: The desired photo or photos exist in a photo album.

Flow of events: **User actions** **System responses**

1. **Include use-case**
 - 1.6.1 ‘Browse for a photo’
2. Selects the desired photo.
3. Repeat step 2 until all the desired photos are selected.
4. Displays a list of possible commands.
5. Chooses the ‘Delete...’ command.
6. Prompt the user to confirm request:
Include use-case
 - 1.3.1 ‘Confirm destructive operation’
7. The desired photo

- | | |
|--|------------------------------|
| <p>or photos disappears.</p> <p>8. Display success message dialog.</p> <p>9. Chooses the Close command of the dialog</p> | <p>10. Dialog disappears</p> |
|--|------------------------------|

Post conditions: Deletes the desired photo or photos from a photo album.

Extensions: N/A

Non-behavioral requirements: N/A

Assumptions: N/A

Issues: N/A

Source: Functional requirement 1.6

2.2.2.12 Use-case 1.6.1

Browse for a photo

Actor: End-user

Goal: To browse for a desired photo

Related use-cases: N/A

Priority: High

Brief description: The end-user browses for a desired photo.

Preconditions: Photo albums and photos exist in the software.

Flow of events: **User actions** **System responses**

- | | |
|--|--|
| <p>1. Select a photo album if the desired photo is not</p> | <p>2. Display the contents of the photo album.</p> |
|--|--|

displayed.

3. Repeat step 1 until
the desired photo is
displayed.

Post conditions: The desired photo is found.

Extensions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: N/A

2.2.2.13 Use-case 1.7

Move photo

Actor: End-user

Goal: To move one or more desired photos from one album to another

Related use-cases: N/A

Priority: High

Brief description: The end-user moves one or more desired photos from its current photo album to the specified photo album.

Preconditions: The desired photos and both photo albums participating in this use-case exist in the software.

Flow of events: **User actions** **System responses**

1. Include use-case

1.6.1 ‘Browse for a photo’

2. Select the desired photo.
3. Repeat step 2 until all the desired photos are selected.
4. Display a list of possible commands.
5. Chooses 'Move...' command.
6. 'Photo album' open dialog appears.
7. Select the album to move the photo or photos to: **Include use-case 1.5.1 'Browse for a photo album'**
8. Confirm selection.
9. The desired photo or photos disappears from the current photo album.
10. Dialog disappears.
11. Displays success message dialog.
12. Chooses the close command of the dialog.
13. Dialog disappears.

Post conditions: The desired photo or photos are moved to the photo album specified.

Extensions: N/A

Non-behavioral N/A

requirements:

Issues: N/A

Source: Functional requirements 1.7

2.2.2.14 Use-case 1.8

Copy photo

Actor: End-user

Goal: To copy one or more desired photos from one photo album to another

Related use-cases: N/A

Priority: High

Brief description: The end-user copies one or more desired photos from its current photo album to the specified photo album.

Preconditions: The desired photos and both photo albums participating in this activity exist.

Flow of events:	User actions	System responses
------------------------	---------------------	-------------------------

1. Include use-case

 1.6.1 ‘Browse for a photo’

2. Select the desired photo.

3. Repeat step 2 until all the desired photos are selected.

5. Chooses ‘Copy...’ command.

4. Display a list of possible commands.

6. Photo album open dialog appears.

7. Select the album to copy the photo or photos to: **Include**
use-case 1.5.1
'Browse for a photo album'
8. Confirm selection.
9. Dialog disappears.
10. Displays success message dialog.

11. Choses the close command of the dialog.
12. Dialog disappears.

Post conditions: The desired photo or photos are copied to the photo album specified.

Extensions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirements 1.8

2.2.2.15 Use-case 2.1

View a single photo

Actor:	End-user	
Goal:	To view the desired photo.	
Related use-cases:	N/A	
Priority:	High	
Brief description:	The end-user views the desired photo.	
Preconditions:	The desired photo exists.	
Flow of events:	User actions	System responses
	1. Include use-case 1.6.1 ‘Browse for a photo’ 2. Select and open the desired photo.	
	3. Display photo. 4. Display associated keywords.	
Post conditions:	A large view of the desired photo along with its associated keywords and name is displayed to the end-user.	
Extensions:	N/A	
Non-behavioral requirements:	N/A	
Issues:	N/A	
Source:	<ul style="list-style-type: none"> • Functional requirement 2.1 • Functional requirement 3.4 • Functional requirement 3.8 	

2.2.2.16 Use-case 2.2

View an entire album of photos via slideshow

Actor:	End-user														
Goal:	To view an entire album of photos in a slideshow format.														
Related use-cases:	N/A														
Priority:	High														
Brief description:	The end-user views all the photos in the desired album via a slideshow mechanism.														
Preconditions:	<ul style="list-style-type: none"> • The desired photo album exists. • The desired photo album has one or more photos. 														
Flow of events:	<table border="0"> <thead> <tr> <th style="text-align: center;">User actions</th> <th style="text-align: center;">System responses</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">1. Select the desired photo album.</td> <td style="vertical-align: top;">2. Display a list of possible commands.</td> </tr> <tr> <td style="vertical-align: top;">3. Chooses 'View as slideshow...' command.</td> <td style="vertical-align: top;">4. Slideshow window appears.</td> </tr> <tr> <td></td> <td style="vertical-align: top;">5. Display a photo from the desired photo album.</td> </tr> <tr> <td></td> <td style="vertical-align: top;">6. Repeat step 5 until the last photo is reached.</td> </tr> <tr> <td></td> <td style="vertical-align: top;">7. Dialog with a list of operations appears.</td> </tr> <tr> <td style="vertical-align: top;">8. Chooses 'Finish...' command.</td> <td style="vertical-align: top;">9. Slideshow window with dialog disappears.</td> </tr> </tbody> </table>	User actions	System responses	1. Select the desired photo album.	2. Display a list of possible commands.	3. Chooses 'View as slideshow...' command.	4. Slideshow window appears.		5. Display a photo from the desired photo album.		6. Repeat step 5 until the last photo is reached.		7. Dialog with a list of operations appears.	8. Chooses 'Finish...' command.	9. Slideshow window with dialog disappears.
User actions	System responses														
1. Select the desired photo album.	2. Display a list of possible commands.														
3. Chooses 'View as slideshow...' command.	4. Slideshow window appears.														
	5. Display a photo from the desired photo album.														
	6. Repeat step 5 until the last photo is reached.														
	7. Dialog with a list of operations appears.														
8. Chooses 'Finish...' command.	9. Slideshow window with dialog disappears.														

Post conditions:	<ul style="list-style-type: none">• The entire photo album of photos is being displayed to the end-user in a slideshow format.• Returns end-user to the relative root photo album of the desired photo album after the slideshow is completed.
Extensions:	<p><u>5a. If the end-user chooses the Stop command before the slideshow is completed:</u> Use-case 2.2.1 ‘Stop slideshow’.</p> <p><u>8a. If the end-user chooses the Replay command instead:</u> Use-case 2.2.2 ‘Replay slideshow’. Rejoin at 7.</p>
Non-behavioral requirements:	<ul style="list-style-type: none">• The slideshow shall have an effect that places each photo on top of the previous photo.• The slideshow shall be context sensitive.• The slideshow shall start with the first photo in the album.
Assumptions:	The end-user is currently in the relative root photo album of the desired photo album.
Issues:	N/A
Source:	<ul style="list-style-type: none">• Functional requirement 2.2• Functional requirement 2.3• Functional requirement 2.4

2.2.2.17 Use-case 2.2.1

Stop slideshow

Actor:	End-user	
Goal:	To stop the slideshow while it is displaying photos.	
Related use-cases:	Extension of: use-case 2.2 'View an entire album of photos via slideshow'	
Priority:	High	
Brief description:	The end-user stops the slideshow while it is displaying the photos of the desired photo album.	
Preconditions:	The viewing of photos of a photo album via slideshow is in progress.	
Flow of events:	User actions	System responses
	1. Chooses the 'Stop...' command.	2. Stops slideshow at the current photo. 3. Slideshow window disappears. 4. Returns the end-user to the relative root photo album of the desired photo album.
Post conditions:	Stops the slideshow and returns the end-user to the relative root photo album.	
Extensions:	N/A	
Non-behavioral requirements:	N/A	
Issues:	N/A	
Source:	Functional requirement 2.4	

2.2.2.18 Use-case 2.2.2

Replay slideshow

Actor:	End-user							
Goal:	To replay the slideshow after it ends.							
Related use-cases:	Extension of: use-case 2.2 'View an entire album of photos via slideshow'							
Priority:	High							
Brief description:	The end-user replays the slideshow that displays the photos of the desired photo album after it ends.							
Preconditions:	<ul style="list-style-type: none"> • The slideshow displaying all the photos of the desired photo album ends. • The slideshow dialog with a list of possible slideshow operations is displayed. 							
Flow of events:	<table border="0"> <thead> <tr> <th>User actions</th> <th>System responses</th> </tr> </thead> <tbody> <tr> <td>1. Chooses the Replay command.</td> <td>2. Display a photo from the desired photo album.</td> </tr> <tr> <td></td> <td>3. Repeat step 2 until the last photo is reached.</td> </tr> </tbody> </table>	User actions	System responses	1. Chooses the Replay command.	2. Display a photo from the desired photo album.		3. Repeat step 2 until the last photo is reached.	
User actions	System responses							
1. Chooses the Replay command.	2. Display a photo from the desired photo album.							
	3. Repeat step 2 until the last photo is reached.							
Post conditions:	Replays the slideshow that displays all the photos of the desired photo album.							
Extensions:	N/A							
Non-behavioral requirements:	<ul style="list-style-type: none"> • The slideshow shall have an effect that places each photo on top of the previous photo. • The slideshow shall be context sensitive. • The slideshow shall start with the first photo listed in the album. 							
Issues:	N/A							

Source: Functional requirement 2.3

2.2.2.19 Use-case 3.1

Send a photo via email

Actor: End-user

Goal: To send a photo to a recipient as an email attachment.

Related use-cases: N/A

Priority: High

Brief description: The end-user sends any desired photo to a recipient as an email attachment.

Preconditions: • The desired photo exists.

Flow of events: **User actions** **System responses**

1. **Include use-case**

1.6.1 ‘Browse for a photo’

2. Select the desired photo.

3. Display the list of possible commands.

4. Choose the ‘Send as email...’ command.

5. Send an email with the desired photo as an attachment:

Include Use-case
3.1.1 ‘Send an attachment via email’

Post conditions: An email is sent to the recipient specified with the desired photo as an attachment.

Extensions: N/A
Non-behavioral requirements: N/A
Issues: N/A
Source: Functional requirement 4.1

2.2.2.20 Use-case 3.1.1

Send an attachment via email

Actor: End-user
Goal: To send an email with an attachment to a recipient.
Related use-cases: N/A
Priority: High
Brief description: The end-user sends an email with an attachment to a recipient.
Preconditions:

- The desired attachment is specified.
- The send specified attachment via email function is initiated.

Flow of events:	User actions	System responses
		<ol style="list-style-type: none"> 1. Email dialog appears. 2. Specify the required email fields. 3. Submit <ol style="list-style-type: none"> 4. Email dialog disappears. 5. Display success message dialog

6. Closes dialog 7. Dialog disappears.

Post conditions: An email is sent to the specified recipient with the desired attachment.

Extensions: 3a. The email settings are not configured or are incorrect:
 Use-case 3.1.2 ‘Attempt to submit email with incorrect or blank settings’. Rejoin at 4.

3b. One or more of the required email fields are missing or invalid: Use-case 3.1.3 ‘Attempt to submit email with missing or invalid email fields’. Rejoin at 4.

Non-behavioral requirements: N/A

Issues: N/A

Source:

- Functional requirement 4.1
- Functional requirement 4.2

2.2.2.21 Use-case 3.1.2

Attempt to submit email with incorrect or blank settings

Actor: End-user

Goal: To prompt the user to configure the email settings.

Related use-cases: Extension of: use-case 3.1.1 ‘Send an attachment via email’

Priority: High

Brief description: This use-case prompts the end-user to configure the email settings, when an attempt has been made to send an email with the email settings not configured or has incorrect values.

Preconditions:

- An attempt has been made to send an email.

- The email settings are not configured or have incorrect values.

Flow of events:	User actions	System responses
		1. A warning dialog appears, informing the end-user of the issue.
	2. Choose the close command on the warning dialog.	3. Dialog disappears.
		4. Include use-case 6.1.1 ‘Configure email settings’
Post conditions:	Email settings configuration is completed.	
Extensions:	N/A	
Non-behavioral requirements:	N/A	
Issues:	N/A	
Source:	N/A	

2.2.2.22 Use-case 3.1.3

Attempt to submit email with missing or invalid email fields

Actor:	End-user
Goal:	To prompt the user to correct the missing or invalid email fields.
Related use-cases:	Extension of: use-case 3.1.1 ‘Send an attachment via email’
Priority:	High

Brief description: This use-case prompts the end-user to correct the missing or invalid email fields encountered when an attempt has been made to send an email with an attachment.

Preconditions:

- An attempt has been made to send an email.
- The email fields have missing or invalid values.

Flow of events:	User actions	System responses
	<p>2. Choose to close command on the dialog.</p>	<ol style="list-style-type: none"> 1. A warning dialog appears, informing the end-user of the issue. 3. Dialog disappears. 4. The email fields or field with invalid or missing value are identified. 5. Correct email fields. 6. Submit

Post conditions: Email fields are complete and valid.

Extensions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: N/A

2.2.2.23 Use-case 3.2

Send an entire photo album of photos via email

Actor:

End-user

Goal:

To send all photos of the desired photo album to a recipient as an email attachment.

Related use-cases:

N/A

Priority:

High

Brief description: The end-user sends all the photos of any desired photo album to a recipient as an email attachment.

Preconditions:

- The desired photo album exists.
- One or more photos exist in the desired photo album.

Flow of events:

User actions

1. Select the desired photo album.

3. Choose the ‘Send as email...’ command.

System responses

2. Display the list of possible commands.

4. Send an email with all the photos of the desired photo album as an attachment: **Include**

Use-case 3.1.1

‘Send an

attachment via email’

Post conditions:

An email is sent to the recipient specified with all the photos of the desired photo album as an attachment.

Assumptions:	The end-user is currently in the relative root photo album of the desired photo album.
Extensions:	N/A
Non-behavioral requirements:	N/A
Issues:	N/A
Source:	Functional requirement 4.2

2.2.2.24 Use-case 4.1

Add keywords

Actor:	End-user	
Goal:	To associate a keyword or a comma-separated list of keywords with the desired photo.	
Related use-cases:	N/A	
Priority:	High	
Brief description:	The end-user associates a keyword or a comma-separated list of keywords with the desired photo that already exists.	
Preconditions:	<ul style="list-style-type: none"> • The desired photo exists. 	
Flow of events:	User actions	System responses
	<ol style="list-style-type: none"> 1. Include use-case 1.6.1 ‘Browse for a photo’ 2. Select to open the desired photo. 3. Specify keywords. 	

4. Submit 5. Dialog disappears.

Post conditions: A keyword or a comma-separated list of keywords is associated with the desired photo.

Assumptions:

Extensions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 3.1

2.2.2.25 Use-case 4.2

Change keyword

Actor: End-user

Goal: To change a keyword associated with the desired photo.

Related use-cases: N/A

Priority: High

Brief description: The end-user changes a keyword associated with the desired photo.

Preconditions:

- The desired photo exists.
- The desired photo already has one or more keywords

Flow of events:	User actions	System responses
	1. Include use-case 1.6.1 ‘Browse for a photo’	

2. Select to open the desired photo.
3. Select the desired keyword
4. Specify the new keyword.
5. Submit

Post conditions: The desired keyword associated with the desired photo is changed.

Assumptions:

Extensions: 8a. If another keyword is selected: Use-case 4.2.1 'Attempt to replace another keyword'. Rejoin at 8.

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 3.2

2.2.2.26 Use-case 4.2.1

Attempt to replace another keyword

Actor: End-user

Goal: To replace another keyword associated with the desired photo.

Related use-cases: Extension of: Use-case 4.2 'Change Keyword'

Priority: High

Brief description: The end-user replaces another keyword associated with the desired photo.

- Preconditions:**
- The desired photo already has more than one keywords
 - The change of a keyword associated with a photo is in progress.
 - A keyword is selected after another keyword has been replaced.

Flow of events:	User actions	System responses
	<ol style="list-style-type: none"> 1. Select the desired keyword 2. Specify the new keyword. 	

Post conditions: The desired keyword associated with the desired photo has been replaced.

Assumptions:

Extensions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source:

2.2.2.27 Use-case 4.3

Delete keyword

Actor: End-user

Goal: To delete one or more keywords associated with the desired photo.

Related use-cases: N/A

Priority: High

Brief description: The end-user a keyword associated with the desired photo.

Preconditions:

- The desired photo exists.
- The desired photo already has one or more keywords

Flow of events:	User actions	System responses
	<ol style="list-style-type: none"> 1. Include use-case 1.6.1 ‘Browse for a photo’ 2. Select to open the desired photo. 3. Select the desired keywords 4. Delete 5. Submit 	

Post conditions: The desired keyword or keywords associated with the desired photo are deleted.

Assumptions: N/A

Extensions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 3.3

2.2.2.28 Use-case 5

Search for a photo

Actor:	End-user										
Goal:	To find a photo in the current or sub photo album based on a keyword or a comma-separated list of keywords.										
Related use-cases:	N/A										
Priority:	High										
Brief description:	The end-user searches the current and sub photo albums for a photo that has the specified keyword or, one or more of the keywords existing the specified comma-separated list of keywords.										
Preconditions:	<ul style="list-style-type: none"> • The current photo album is not the root photo album. • The current photo album is not empty. • The one or more photos in the software have a keyword associated to it. 										
Flow of events:	<table border="0"> <thead> <tr> <th>User actions</th> <th>System responses</th> </tr> </thead> <tbody> <tr> <td>1. Select the desired photo album.</td> <td>2. Display the search component.</td> </tr> <tr> <td>3. Specify the desired keyword or keywords.</td> <td></td> </tr> <tr> <td>4. Submit</td> <td>5. Search component retains the specified keyword or keywords.</td> </tr> <tr> <td></td> <td>6. Photo album content disappears</td> </tr> </tbody> </table>	User actions	System responses	1. Select the desired photo album.	2. Display the search component.	3. Specify the desired keyword or keywords.		4. Submit	5. Search component retains the specified keyword or keywords.		6. Photo album content disappears
User actions	System responses										
1. Select the desired photo album.	2. Display the search component.										
3. Specify the desired keyword or keywords.											
4. Submit	5. Search component retains the specified keyword or keywords.										
	6. Photo album content disappears										

7. Display search results.

Post conditions:	The photos matching the desired keyword or keywords are displayed.
Assumptions:	The end-user is currently in the relative root photo album of the desired photo album.
Extensions:	<u>7a. If there were no photos that match the specified keyword or keywords:</u> Use-case 5.1 ‘Search returned no match’. <u>7b. If an attempt has been made to exit the search results:</u> Use-case 5.2 ‘Exit search’.
Non-behavioral requirements:	N/A
Issues:	N/A
Source:	Functional requirement 6.1

2.2.2.29 Use-case 5.1

Search returned no match

Actor:	End-user
Goal:	To inform the end-user that there were no photos that matched the specified keyword or keywords.
Related use-cases:	Extension to: Use-case 5 ‘Search for a photo’
Priority:	High
Brief description:	This use-case informs the end-user that there were no photos that matched the specified keyword or keywords and displays no photo.

Preconditions:	<ul style="list-style-type: none"> • A search operation has been carried out. • There were no photo or photos that matched the specified keyword or keywords. 	
Flow of events:	User actions	System responses
		<ol style="list-style-type: none"> 1. Inform the user of the current state. 2. Display empty search result screen.
Post conditions:	The user is informed that there were no photos that matched the specified keyword or keywords and display no photos.	
Assumptions:	The end-user is currently in the desired photo album.	
Extensions:	<u>2a. If an attempt has been made to exit the search results:</u> Use-case 5.2 ‘Exit search’	
Non-behavioral requirements:	N/A	
Issues:	N/A	
Source:	N/A	

2.2.2.30 Use-case 5.2

Exit search

Actor:	End-user
Goal:	To exit the current search results.
Related use-cases:	Extension to: <ul style="list-style-type: none"> • Use-case 5 ‘Search for a photo’ • Use-case 5.1 ‘No photos matched the specified keyword or keywords’

Priority:	High	
Brief description:	The end-user exits the search results of the previous search operation.	
Preconditions:	<ul style="list-style-type: none"> • A search operation has been carried out. • The search results from the previous search operation are being displayed. 	
Flow of events:	User actions	System responses
	<ol style="list-style-type: none"> 1. Chooses the clear command of the search component. 	<ol style="list-style-type: none"> 2. The specified keyword or keywords are removed from the search component.
		<ol style="list-style-type: none"> 3. Search results disappear
		<ol style="list-style-type: none"> 4. Displays photo album content.
Post conditions:	The results from the previous search operation disappear, leaving the original content of the current photo album displayed. The search component is cleared.	
Assumptions:	The end-user is currently in the desired photo album.	
Non-behavioral requirements:	N/A	
Issues:	N/A	
Source:	N/A	

2.2.2.31 Use-case 6.1

Setup email

Actor: End-user
Goal: To configure the email feature with the required settings for the first time.
Related use-cases: N/A
Priority: High
Brief description: The end-user configures the email settings for the first time so that the software will be able to send emails without using a third-party application.

Preconditions: The email settings are being configured for the first time.

Flow of events:	User actions	System responses
	1. Chooses the 'preferences...' command.	2. Preference dialog appears.
	3. Chooses the 'Setup email...' command.	4. Include use-case 6.1.1 'Configure email settings'

Post conditions: The email feature has been setup and configured with the required and valid values.

Assumptions:

Non-behavioral requirements: N/A

Issues:

N/A

Source: Functional requirement 5.1

2.2.2.32 Use-case 6.1.1

Configure email settings

Actor:	End-user				
Goal:	To prompt the user to configure the email settings.				
Related use-cases:	N/A				
Priority:	High				
Brief description:	This use-case prompts the end-user to configure the email settings by specifying the values for the required email settings to enable the software to send emails without using a third-party application.				
Preconditions:	This use-case has been included by another use-case.				
Flow of events:	User actions	System responses			
		1. Email settings form appears.			
	2. Specify email settings.	4. Save settings			
	3. Submit	5. Display success message dialog.			
	6. Choose close command on dialog.	7. Dialog disappears			
Post conditions:	The email settings are completed with the required and valid values.				
Assumptions:	The email settings specified are correct.				
Extensions:	N/A				
Non-behavioral	N/A				

requirements:

Issues: N/A

Source: N/A

2.2.2.33 Use-case 6.2

Modify email settings

Actor: End-user

Goal: To modify the email settings that is already in the software.

Related use- N/A

cases:

Priority: High

Brief description: The end-user modifies the email settings that are already in the software.

Preconditions: The email settings are not being configured for the first time.

Flow of events:	User actions	System responses
	1. Chooses the 'preferences...' command.	2. Preferences dialog appears.
	3. Chooses the 'Modify email settings ...' command.	4. Include use-case 6.1.1 'Configure email settings'
Post conditions:	The one or more of the email settings are replaced with new values.	

Assumptions:

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 5.1

2.2.2.34 *Use-case 6.3*

Modify slideshow duration

Actor: End-user

Goal: To modify the existing slideshow duration.

Related use-cases: N/A

Priority: High

Brief description: The end-user modifies the existing slideshow duration.

Preconditions: Slideshow duration exists.

Flow of events:	User actions	System responses
	1. Chooses the 'preferences...' command.	2. Preferences dialog appears.
	3. Chooses the 'Modify slideshow settings ...' command.	4. Slideshow settings appear.
	5. Specifies the slideshow duration.	
	6. Submit	7. Dialog disappears.

- 8. Display success message dialog.
- 9. Chooses the close command on the dialog.
- 10. Dialog disappears.

Post conditions: The slideshow duration setting has the newly specified value.

Assumptions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 5.2

2.2.2.35 Use-case 6.4

Modify slideshow replay loop count

Actor: End-user

Goal: To modify the existing slideshow replay loop count.

Related use-cases: N/A

Priority: High

Brief description: The end-user modifies the existing slideshow replay loop count.

Preconditions: Slideshow replay loop count exists.

Flow of events:	User actions	System responses
	<ul style="list-style-type: none"> 1. Chooses the ‘preferences...’ command. 	<ul style="list-style-type: none"> 2. Preference dialog appears.

3. Chooses the
‘Modify slideshow
settings ...’
command.
4. Slideshow settings
appear.
5. Specifies the
slideshow replay
loop count.
6. Submit
7. Dialog disappears.
8. Display success
message dialog.
9. Chooses the close
command on the
dialog.
10. Dialog disappears.

Post conditions: The slideshow replay loop count setting has the newly specified value.

Assumptions: N/A

**Non-behavioral
requirements:** N/A

Issues: N/A

Source: Functional requirement 5.3

2.2.2.36 Use-case 7.1

Edit photo's image

Actor:	End-user	
Goal:	To edit the desired photo's image.	
Related use-cases:	Generalization of:	
	<ul style="list-style-type: none"> • Use-case 7.1.1 'Convert photo's image to black and white' • Use-case 7.1.2 'Change photo's image contrast' • Use-case 7.1.3 'Change photo's image brightness' • Use-case 7.1.4 'Rotate photo's image' 	
Priority:	High	
Brief description:	The end-user edits a desired photo's image.	
Preconditions:	The desired photo exists.	
Flow of events:	User actions	System responses
	<ol style="list-style-type: none"> 1. Include use-case 1.6.1 'Browse for a photo' 2. Select the desired photo. 4. Choose the 'Edit...' command. 6. Edit photo's image 8. Submit 	
	<ol style="list-style-type: none"> 3. Display the list of possible commands. 5. The edit window with the desired photo's image appears. 7. Display the edited photo's image in the edit window instead of the original. 9. Include Use-case 	

7.1.5 'Select save method'

Post conditions: The desired photo's image is edited based on the desired editing method.

Assumptions:

Non-behavioral requirements: N/A

Issues: N/A

Source:

- Functional requirement 7.1
- Functional requirement 7.2
- Functional requirement 7.3
- Functional requirement 7.4

2.2.2.37 Use-case 7.1.1

Convert photo's image to black and white

Actor: End-user

Goal: To convert the desired photo's image to black and white.

Related use-cases: Specialization of: Use-case 7.1 'Edit photo image'

Priority: High

Brief description: The end-user edits the desired photo's image by converting it to black and white.

Preconditions: The desired photo exists.

Flow of events: **User actions** **System responses**

1. Include use-case

1.6.1 'Browse for a photo'

2. Select the desired photo.
3. Display the list of possible commands.
4. Choose the 'Edit...' command.
5. The edit window with the desired photo's image appears.
6. Choose the 'Convert to black and white...' command.
7. The original photo's image disappears and the edited photo's image appears
8. Submit
9. **Include Use-case 7.1.5 'Select save method'**

Post conditions: The desired photo's image is converted to black and white.

Assumptions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 7.1

2.2.2.38 Use-case 7.1.2

Change photo's image contrast

Actor:	End-user	
Goal:	To change the contrast of the desired photo's image.	
Related use-cases:	Specialization of: Use-case 7.1 'Edit photo'	
Priority:	High	
Brief description:	The end-user edits the desired photo's image by changing its contrast by a specified value.	
Preconditions:	The desired photo exists.	
Flow of events:	User actions 1. Include use-case 1.6.1 'Browse for a photo' 2. Select the desired photo. 4. Choose the 'Edit...' command. 6. Choose the 'Change contrast' command. 6.b.Specifies a contrast value. 8. Submit	System responses 3. Display the list of possible commands. 5. The edit window with the desired photo's image appears. 6.a. Prompt the user for a contrast value. 6.c. Prompt disappears. 7. The original photo's image disappears and the edited photo's image appears 9. Include Use-case 7.1.5 'Select save'

method'

Post conditions:	The contrast of the desired photo's image is changed based on the specified contrast value.
Assumptions:	N/A
Non-behavioral requirements:	N/A
Issues:	N/A
Source:	Functional requirement 7.2

2.2.2.39 Use-case 7.1.3

Change photo's image brightness

Actor:	End-user	
Goal:	To change the brightness of the desired photo's image.	
Related use-cases:	Specialization of: Use-case 7.1 'Edit photo's image '	
Priority:	High	
Brief description:	The end-user edits the desired photo's image by changing its brightness by a specified value.	
Preconditions:	The desired photo exists.	
Flow of events:	User actions	System responses
	1. Include use-case 1.6.1 ‘Browse for a photo’	
	2. Select the desired photo.	3. Display the list of possible commands.
	4. Choose the ‘Edit...’ command.	5. The edit window with the desired photo's image appears.

- | | |
|---|---|
| 6. Choose the ‘Change brightness’ command. | 6.a. Prompt the user for a brightness value. |
| 6.b.Specifies a brightness value. | 6.c. Prompt disappears. |
| 7. The original photo’s image disappears and the edited photo’s image appears | |
| 8. Submit | 9. Include Use-case
7.1.5 ‘Select save method’ |

Post conditions: The brightness of the desired photo’s image is changed based on the specified brightness value.

Assumptions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 7.3

2.2.2.40 Use-case 7.1.4

Rotate photo’s image

Actor: End-user

Goal: To rotate the desired photo’s image as desired.

Related use-cases: Specialization of: Use-case 7.1 ‘Edit photo’s image’

Priority: High

Brief description: The end-user edits the desired photo’s image by rotating

it as desired.

Preconditions: The desired photo exists.

Flow of events: **User actions** **System responses**

1. **Include use-case 1.6.1**
'Browse for a photo'
2. Select the desired photo.
3. Display the list of possible commands.
4. Choose the 'Edit...' command.
5. The edit window with the desired photo's image appears.
6. Choose any of the 'Rotate direction' commands.
- 6.a. Prompt the user to specify rotation direction and angle.
- 6.c. Prompt disappears.
7. The original photo's image disappears and the edited photo's image appears
8. Submit
9. **Include Use-case 7.1.5 'Select save method'**

Post conditions: The rotation of the desired photo's image is changed based on the specified direction and angle.

Assumptions: N/A

Non-behavioral requirements: N/A

Issues: N/A

Source: Functional requirement 7.4

2.2.2.41 Use-case 7.1.5

Select save method

Actor: End-user

Goal: To prompt the user to choose a save method.

Related use-cases: N/A

Priority: High

Brief description: This use-case prompts the end-user to choose if the edited photo's image should overwrite the original photo's image or it should be saved with a different filename specified.

Preconditions:

- The desired photo exists.
- The desired photo was edited.
- An attempt was made to save the edited photo.

Flow of events:	User actions	System responses
		<ol style="list-style-type: none"> 1. Dialog appears prompting the user for a choice. 2. Chooses the 'do not overwrite...' command. 3. Prompt the user to specify the filename to save as. 4. Specifies filename 5. Submit 6. Prompt dialog disappears.

7. Display success message dialog.
8. Chooses the close command of the dialog.
9. Dialog disappears.

Post conditions: The edited photo's image is saved with the specified filename.

Assumptions:

Extensions: 2a. If the 'overwrite' command was choose instead: Use-case 7.1.6 'Overwrite existing photo's image'

Non-behavioral requirements: N/A

Issues: N/A

Source: N/A

2.2.2.42 Use-case 7.1.6

Overwrite existing photo's image

Actor: End-user

Goal: To overwrite the existing photo's image.

Related use-cases: Extension of: Use-case 7.1.5 'Select save method'

Priority: High

Brief description: The end-user chose to overwrite the original photo's image with the edited photo's image.

Preconditions:

- The original version of the desired photo's image exists.
- The desired photo's image was edited.
- An attempt was made to save the edited photo's

image.

- The choice was made to overwrite the original photo's image with the edited photo's image.

Flow of events:	User actions	System responses
	1. Chooses the 'overwrite...' command.	2. Prompt dialog disappears.
		3. Display success message dialog.
	4. Chooses the close command of the dialog.	5. Dialog disappears.
Post conditions:	The edited photo's image is saved and the original photo's image was overwritten.	
Assumptions:	N/A	
Extensions:	N/A	
Non-behavioral requirements:	N/A	
Issues:	N/A	
Source:	N/A	

2.3 Non-Functional Requirements

2.3.1 Compatibility

2.3.1.1 *Non-functional requirement 1.1*

The software shall be compatible with MAC OSX 10.9+ and Windows 8.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.1.2 *Non-functional requirement 1.2*

The software shall support JPEG and PNG file formats for photos.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.2 Reliability

2.3.2.1 *Non-functional requirement 2.1*

The software shall have a MTBF of 1000 hours.

2.3.2.2 *Non-functional requirement 2.2*

The software shall have 0.1% chance of failing every hour.

2.3.2.3 *Non-functional requirement 4.1*

Novice users shall be able to create a new photo album and add a photo to it within 2 minutes.

2.3.2.4 *Non-functional requirement 4.2*

Expert users shall be able to create a new photo album and add a photo to it within 1 minute.

2.3.2.5 *Non-functional requirement 4.3*

80% of the users surveyed after 3 months of usage shall rate their satisfaction with the system at 7 and more on a scale of 1 to 10.

2.3.2.6 *Non-functional requirement 5.4*

95% of the users shall be able to complete photo-organizing tasks without requiring assistance.

2.3.3 Correctness

2.3.3.1 *Non-functional requirement 5.1*

The software shall maintain the photographs and photo albums as how the user organizes them.

2.3.4 Implementation

2.3.4.1 *Non-functional requirement 6.1*

The software shall be standalone.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.4.2 *Non-functional requirement 6.2*

The software shall take complete ownership of the photos added.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.4.3 Non-functional requirement 6.3

The software shall send emails without a third-party application.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.4.4 Non-functional requirement 6.4

The software shall carryout black and white conversion, contrast and brightness modification, and photo rotation without a third-party application.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.5 Other Non-functional requirements

2.3.5.1 Non-functional requirement 7.1

The software shall allow any photo album to contain one or more photo albums.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.5.2 Non-functional requirement 7.2

The search feature shall be context sensitive.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.5.3 Non-functional requirement 7.3

The slideshow shall be context sensitive.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.5.4 Non-functional requirement 7.4

The user shall not be able to delete the root album.

Priority: High

Source: Dr. Mitropoulos

Date: August 27, 2014

2.3.5.5 Non-functional requirement 7.5

The user shall not be able to create photo albums of the same name in the same root album relative to it.

Priority: High

2.3.5.6 Non-functional requirement 7.6

The user shall be asked to create a root photo album when the software is started for the first time.

Priority: High

Source: Dr. Mitropoulos

Date: September 3, 2014

2.3.5.7 Non-functional requirement 7.7

Each photo album shall have a unique name within its relative root photo album.

Priority: High

Source: Dr. Mitropoulos

Date: September 3, 2014

2.3.5.8 Non-functional requirement 7.8

The slideshow shall have an effect that places each photo on top of the previous photo.

Priority: High

Source: Dr. Mitropoulos

Date: September 3, 2014

2.3.5.9 Non-functional requirement 7.9

The user shall be able to confirm or cancel destructive operations.

Priority: High

Source: Dr. Mitropoulos

Date: September 3, 2014

2.3.5.10 Non-functional requirement 7.10

The photos shall be ordered by date/time added in descending order in any display layout inclusive of lists and tables.

Priority: High

2.3.5.11 Non-functional requirement 7.11

The photo albums shall be ordered by name in ascending order in any display layout inclusive of lists and tables.

Priority: High

2.3.6 Reusability

2.3.6.1 Non-functional requirement 8.1

At least 80% of the source code in the software shall be reusable.

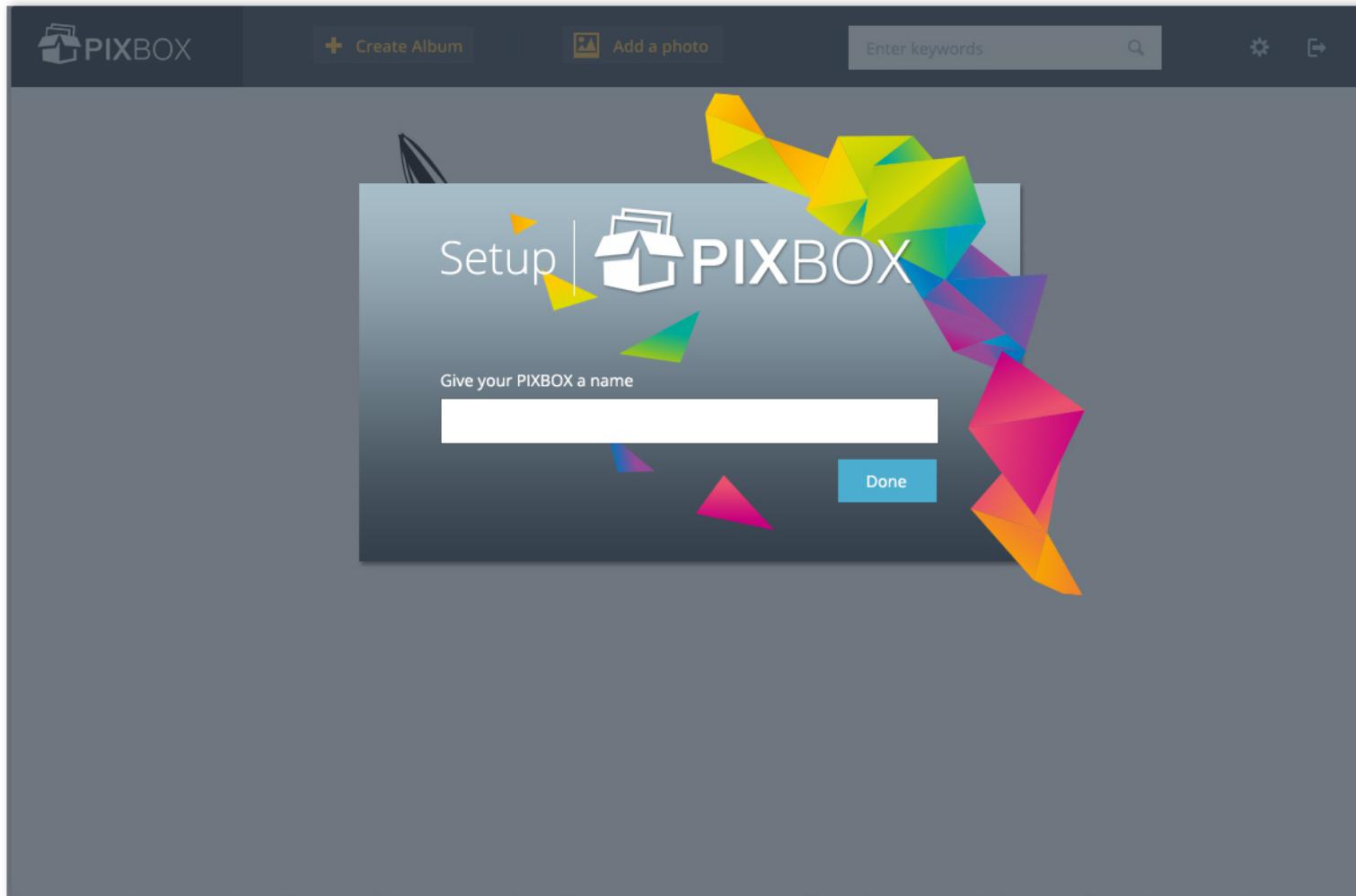
Priority: High

3 Requirements Prototyping

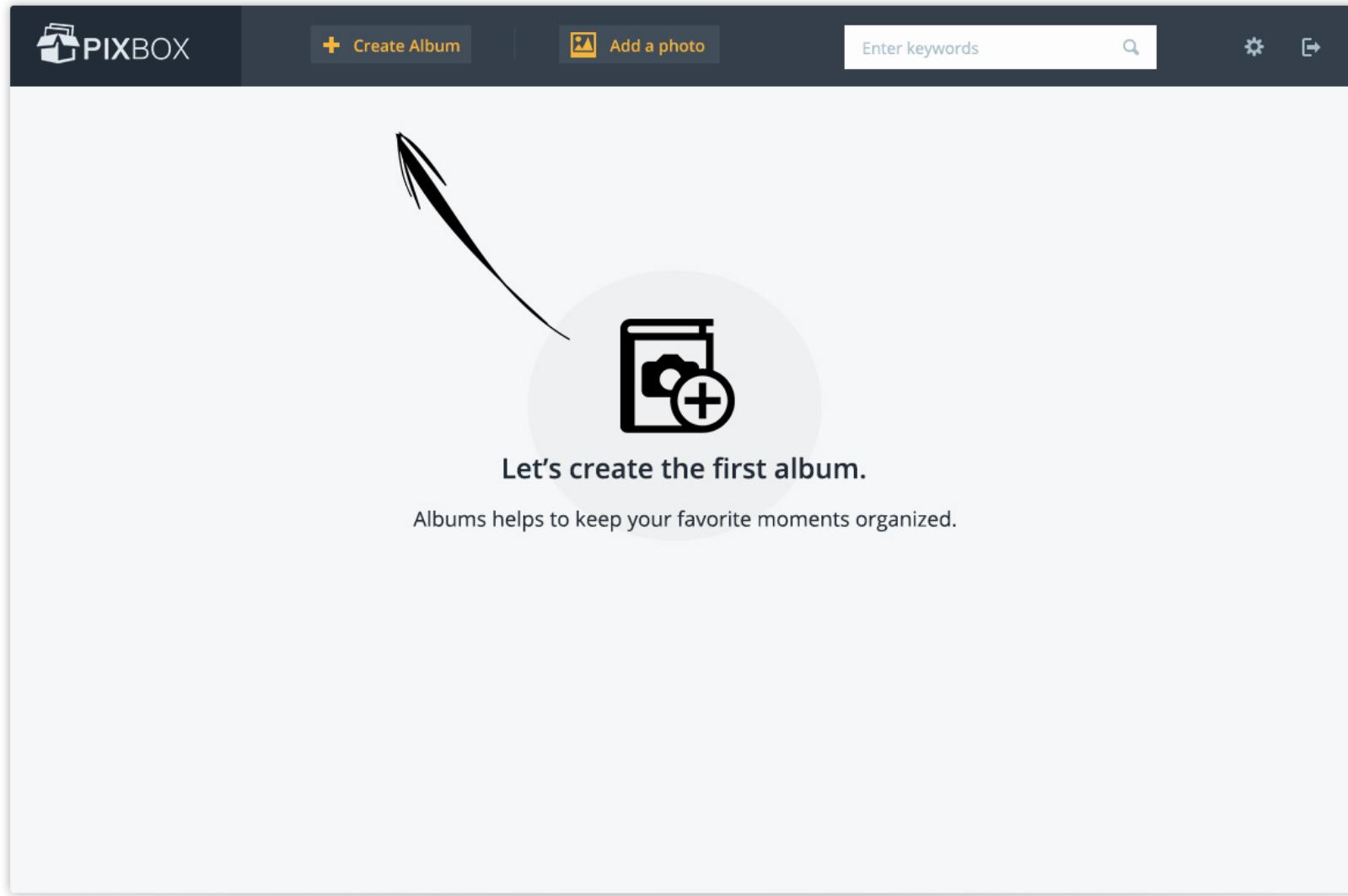
This section aids in the refinement and elaboration of the stated requirements and use cases through visualization.

3.1 Photo organization

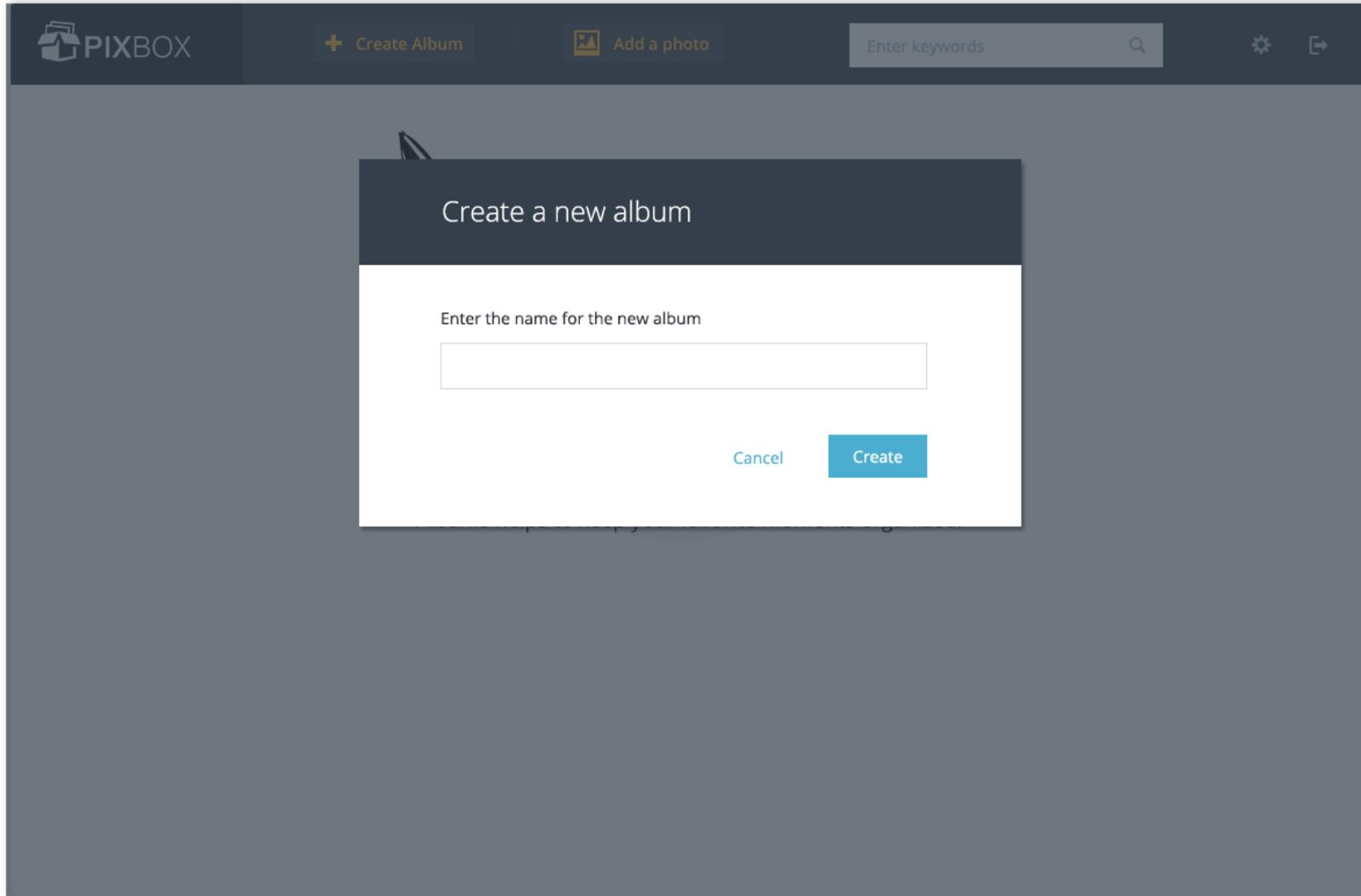
3.1.1 Prototype 1.1: Create root photo album



3.1.2 Prototype 1.2: Empty slate – no photo album exists



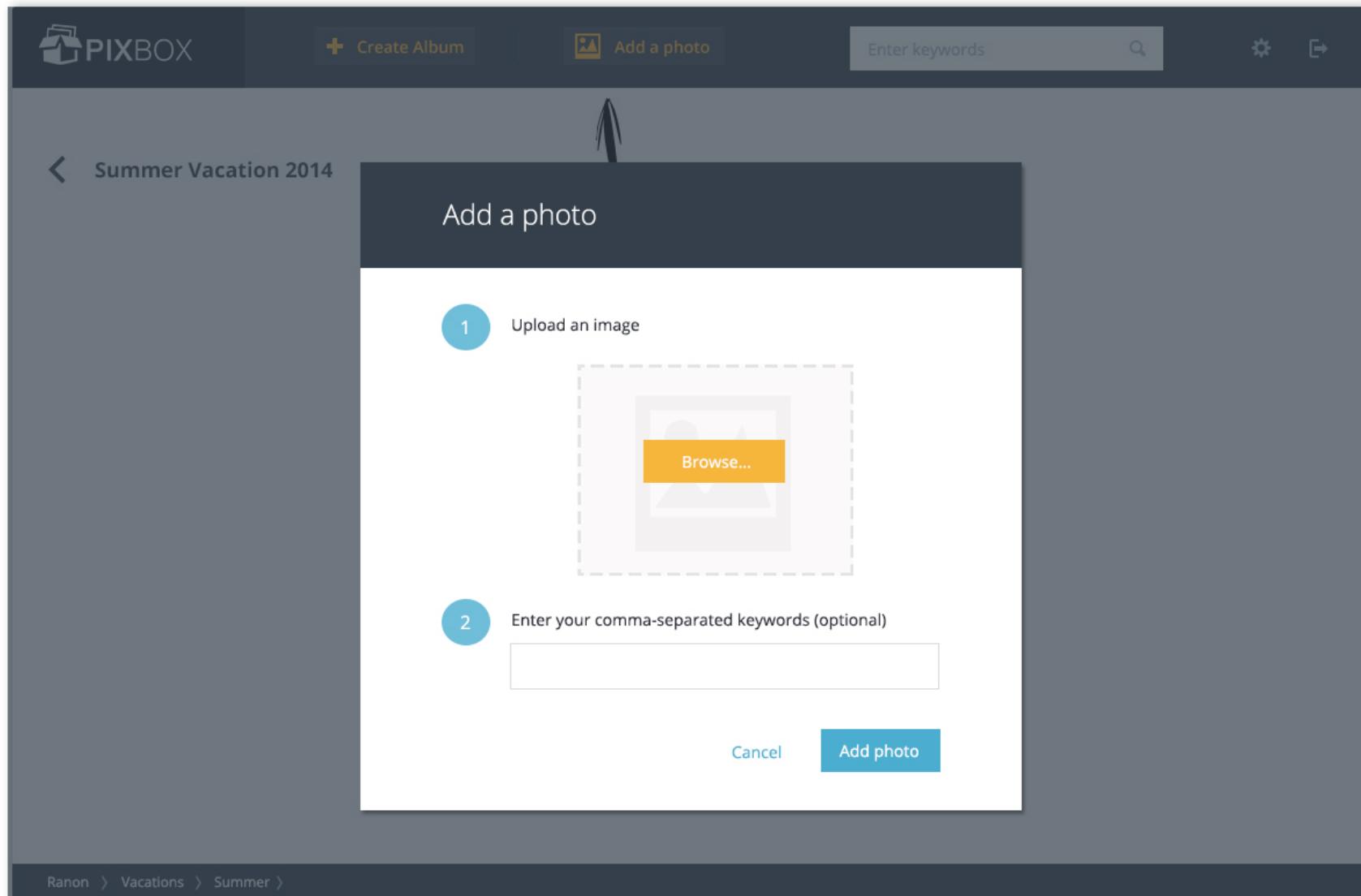
3.1.3 Prototype 1.3: Create new photo album



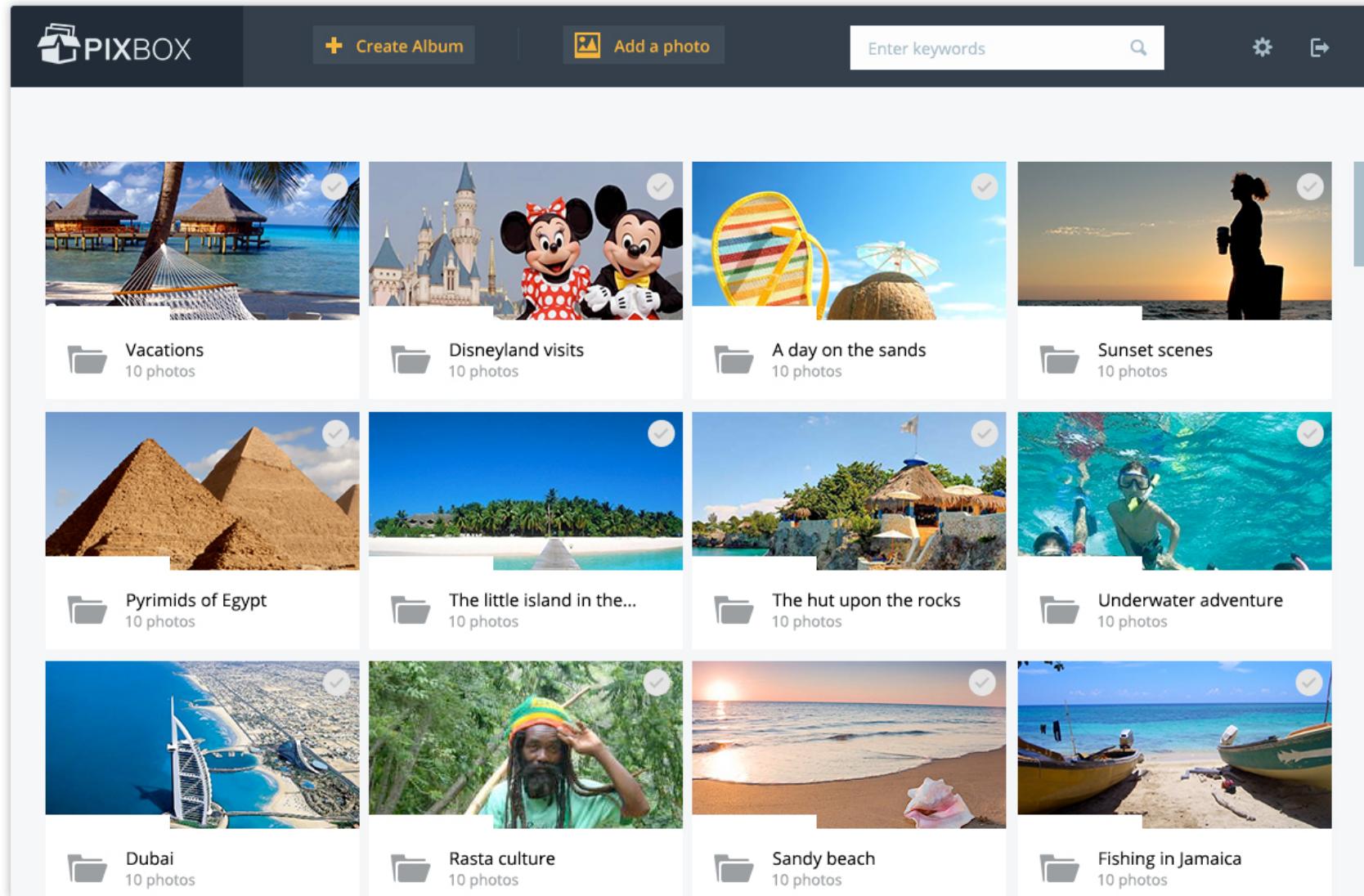
3.1.4 Prototype 1.4: Empty slate – no photo exists

The screenshot shows a user interface for a photo album application named PIXBOX. At the top, there is a dark header bar with the PIXBOX logo, a 'Create Album' button, an 'Add a photo' button, a search bar labeled 'Enter keywords', and a settings gear icon. Below the header, the title 'Summer Vacation 2014' is displayed with a back arrow. In the center, there is a large, semi-transparent circular placeholder with a camera icon and a plus sign, indicating where a photo can be added. Below this placeholder, the text 'Let's add your first photo to this album.' is shown, followed by the subtext 'Albums helps to keep your favorite moments organized.' At the bottom of the screen, a navigation bar shows the path: Ranon > Vacations > Summer >.

3.1.5 Prototype 1.5: Add a photo to an existing photo album



3.1.6 Prototype 1.6: Root photo album with content



3.1.7 Prototype 1.7: Photo album with content and multiple photo selection

The screenshot shows the PIXBOX photo album interface. At the top, there is a dark header bar with the PIXBOX logo, a 'Create Album' button, an 'Add a photo' button, a search bar labeled 'Enter keywords', and settings/gear icons. Below the header is a yellow navigation bar with the text 'With the 3 selected photos:' and three buttons: 'Copy', 'Move', and 'Delete'. A back arrow and the title 'Summer Vacation 2014' are also present. The main content area displays a grid of 12 vacation photos. Each photo has a small yellow circular checkbox in the top right corner. Some photos have additional information below them: the second photo is part of a 'Disneyland' folder containing 10 photos; the fifth photo is part of an 'Egypt' folder containing 10 photos; and the bottom-left photo shows an aerial view of a city with a prominent skyscraper. The bottom navigation bar shows the path: Ranon > Vacations > Summer >.

3.1.8 Prototype 1.8: Photo album with content and single photo selection

The screenshot shows a user interface for a photo album application named PIXBOX. At the top, there is a navigation bar with the PIXBOX logo, a 'Create Album' button, an 'Add a photo' button, a search bar containing the text 'Sunshine', and various settings icons. Below the navigation bar is a yellow toolbar with buttons for 'Copy', 'Move', 'Delete', 'View', 'Send as email', and 'Edit image'. The main content area displays a grid of 15 photos. The first four photos are in the top row: a beach scene with a hammock, Mickey and Minnie Mouse at Disneyland, a pair of flip-flops and sunglasses on a beach, and a person silhouetted against a sunset. The next four photos are in the second row: pyramids in Egypt, a wooden pier leading to a tropical island, a rock bar on a cliff, and two people snorkeling in clear blue water. The last seven photos are in the third row: an aerial view of a coastal city, a man with dreadlocks wearing a colorful hat, a sunrise over a beach, and a small boat on a beach.

With the selected photo: Copy Move Delete View Send as email Edit image

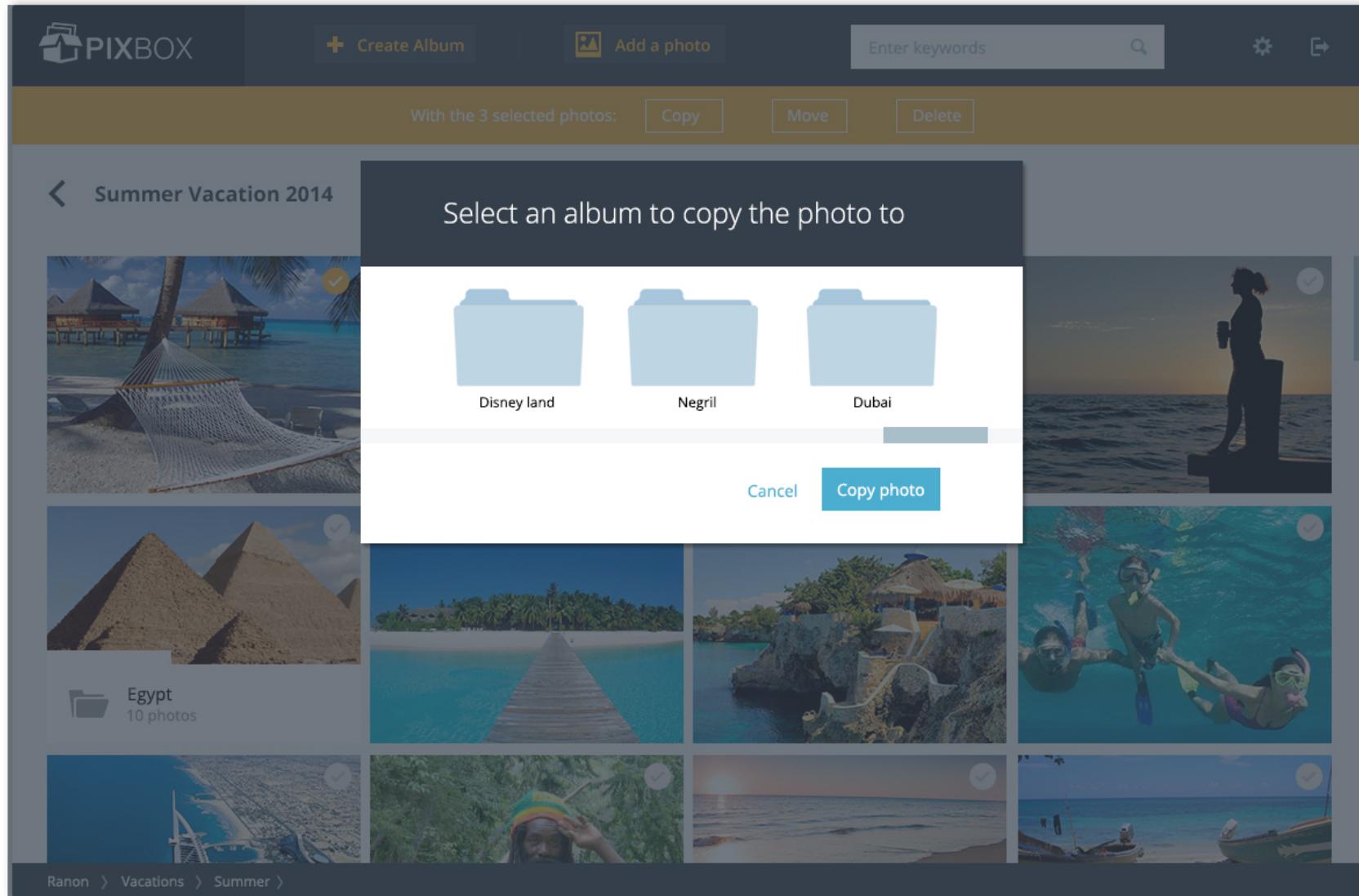
Summer Vacation 2014

Disneyland 10 photos

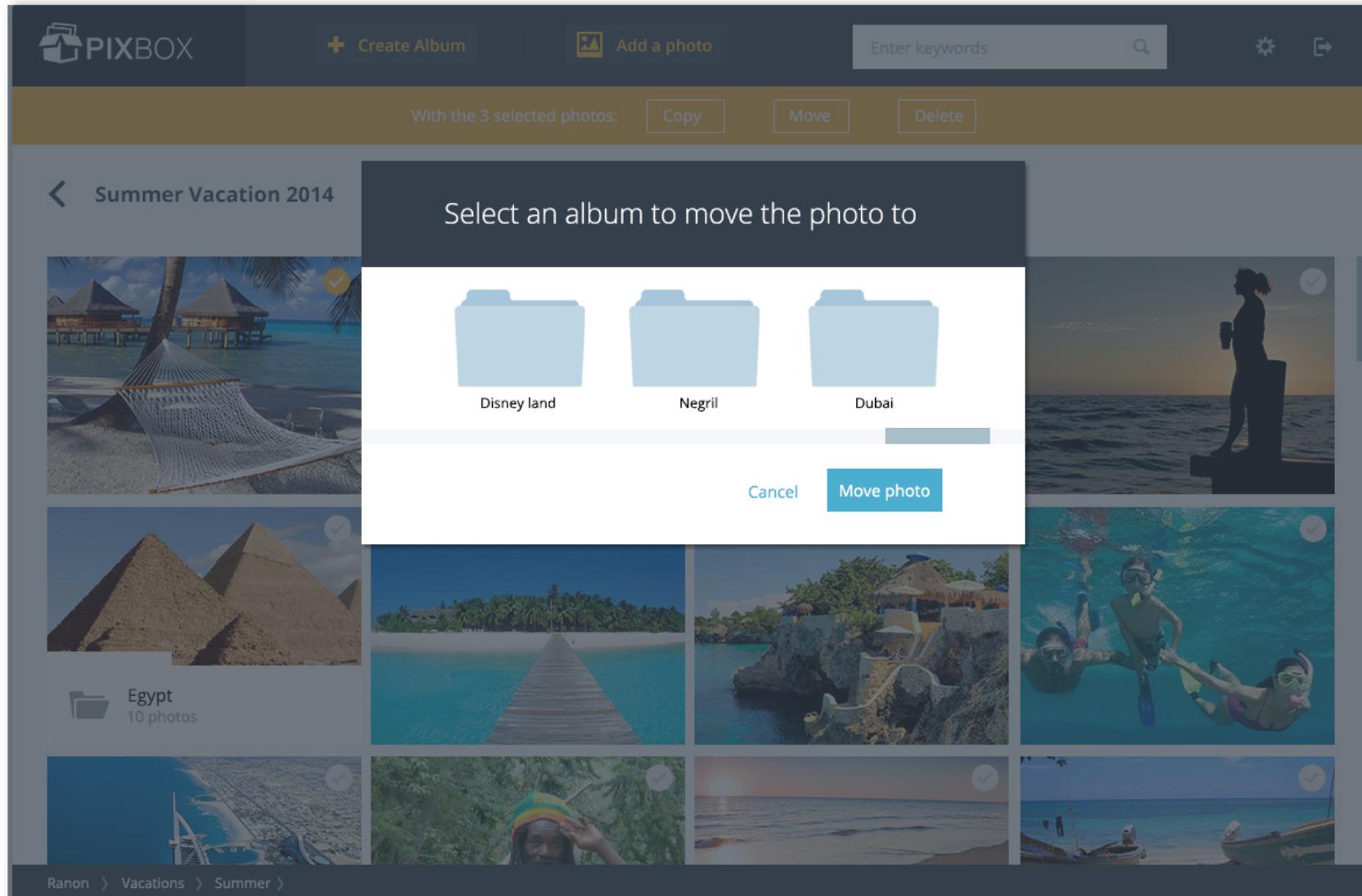
Egypt 10 photos

Ranon > Vacations > Summer >

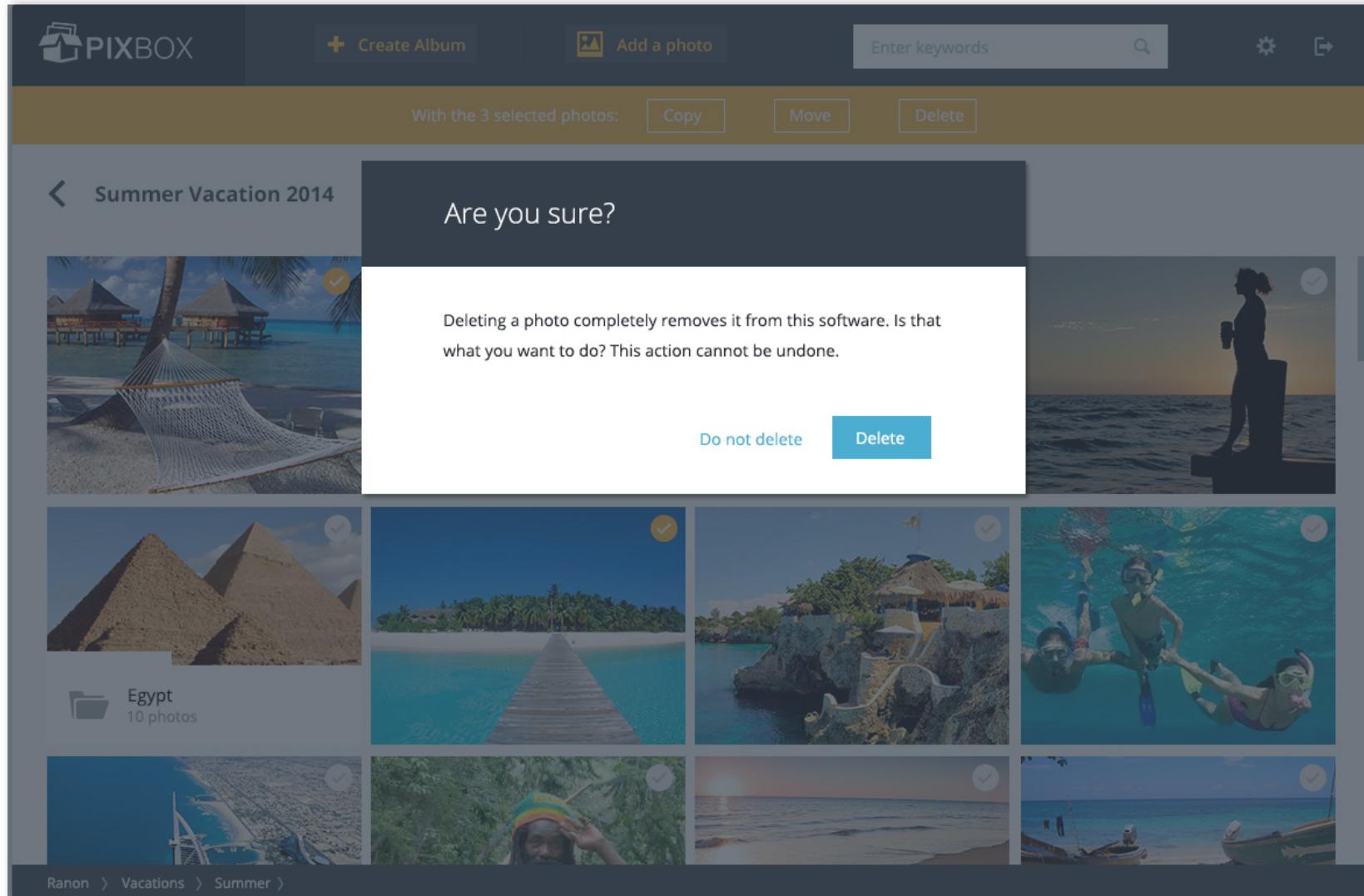
3.1.9 Prototype 1.9: Copy photo



3.1.10 Prototype 1.10: Move photo



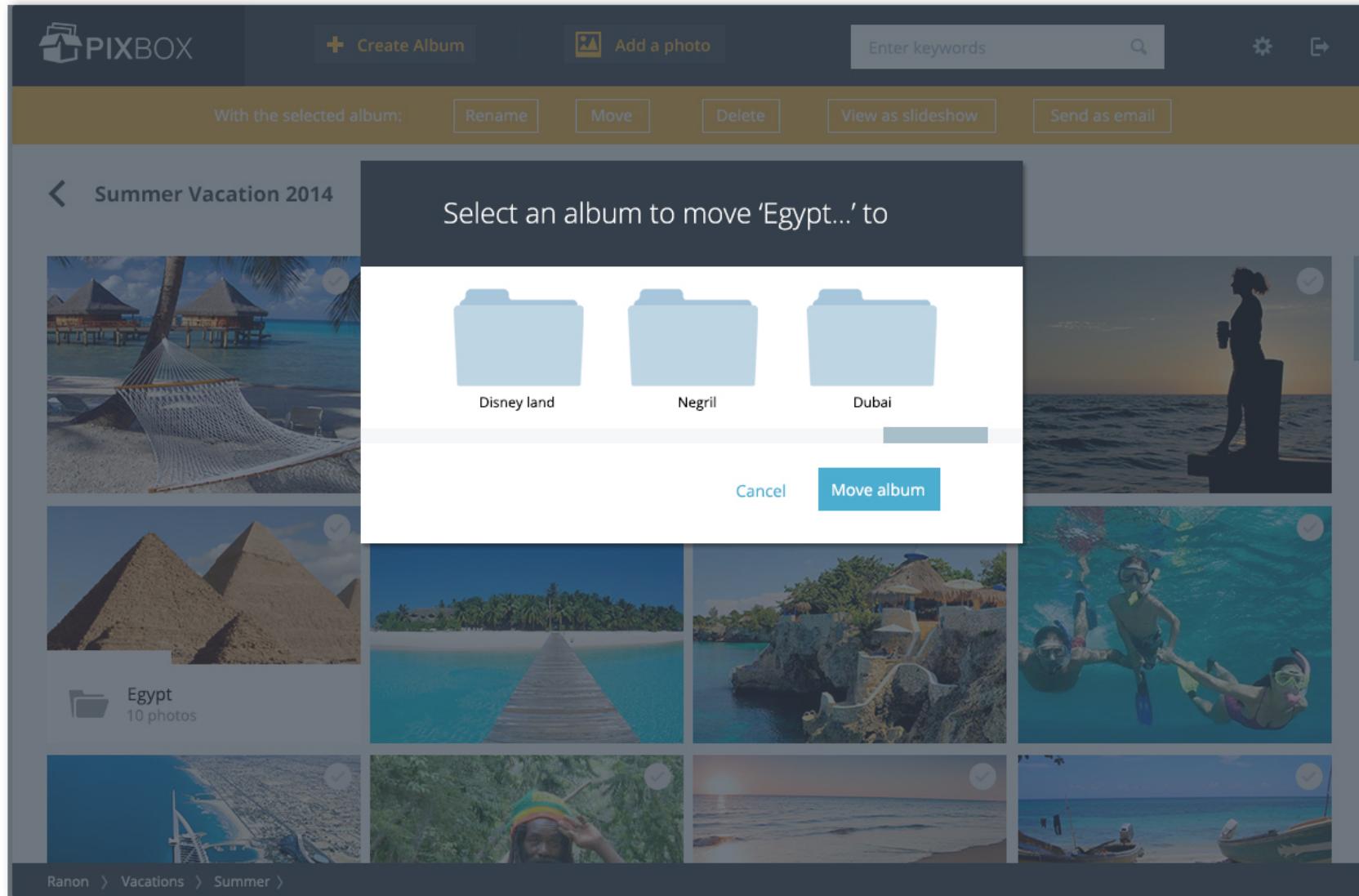
3.1.11 Prototype 1.11: Delete photo



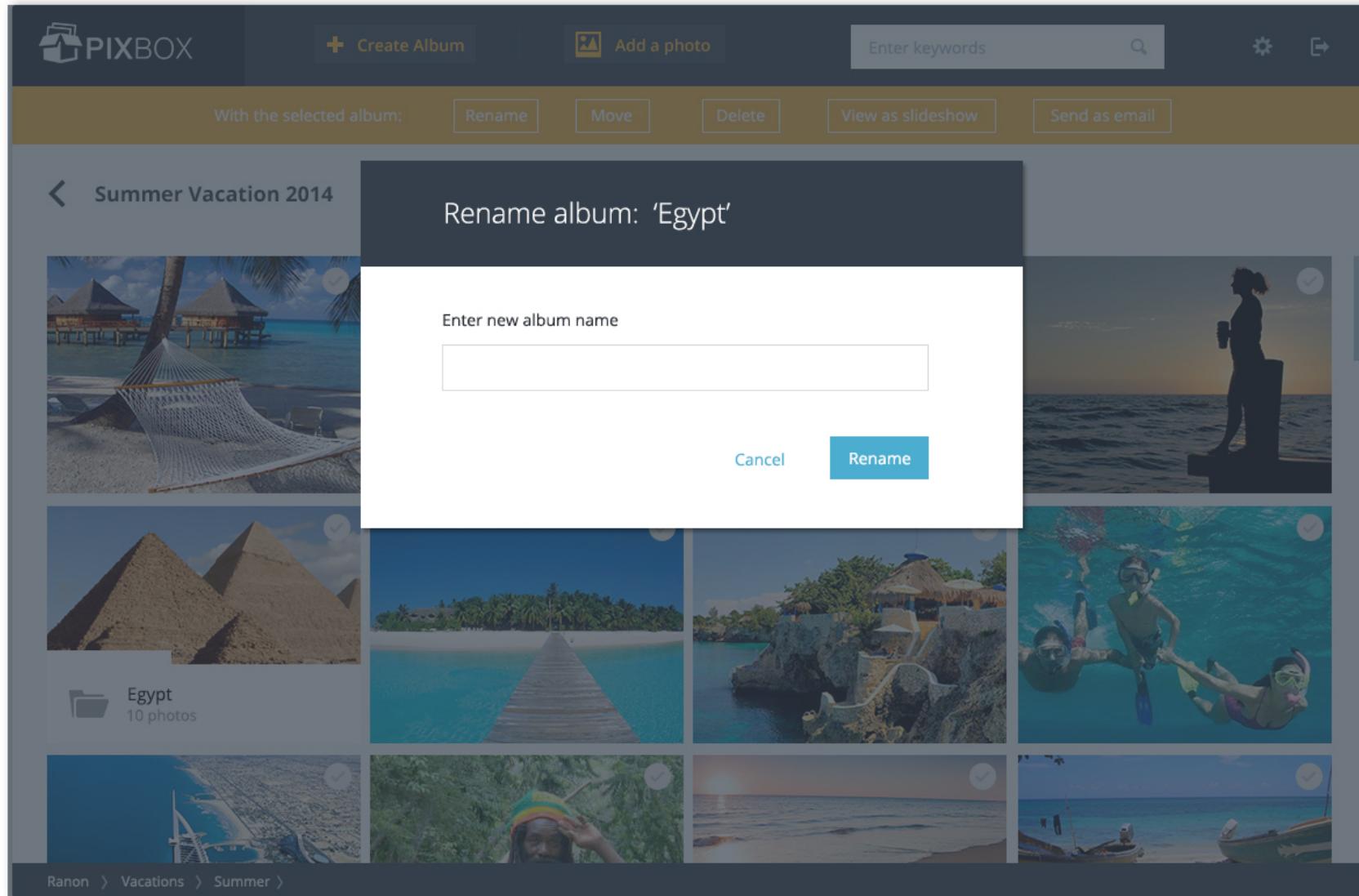
3.1.12 Prototype 1.12: Photo album with content and single photo album selection

The screenshot shows a user interface for managing a photo album. At the top, there's a navigation bar with the PIXBOX logo, a 'Create Album' button, an 'Add a photo' button, a search bar labeled 'Enter keywords', and settings/sync icons. Below the bar is a yellow header with buttons for 'Rename', 'Move', 'Delete', 'View as slideshow', and 'Send as email'. The main content area is titled 'Summer Vacation 2014' with a back arrow. It displays a grid of 15 photo thumbnails. Some thumbnails have checkmarks in their top right corners. Two thumbnails are highlighted with a white overlay: one for 'Disneyland' (10 photos) showing Mickey and Minnie Mouse, and another for 'Egypt' (10 photos) showing pyramids. The bottom navigation bar shows the path: Ranon > Vacations > Summer.

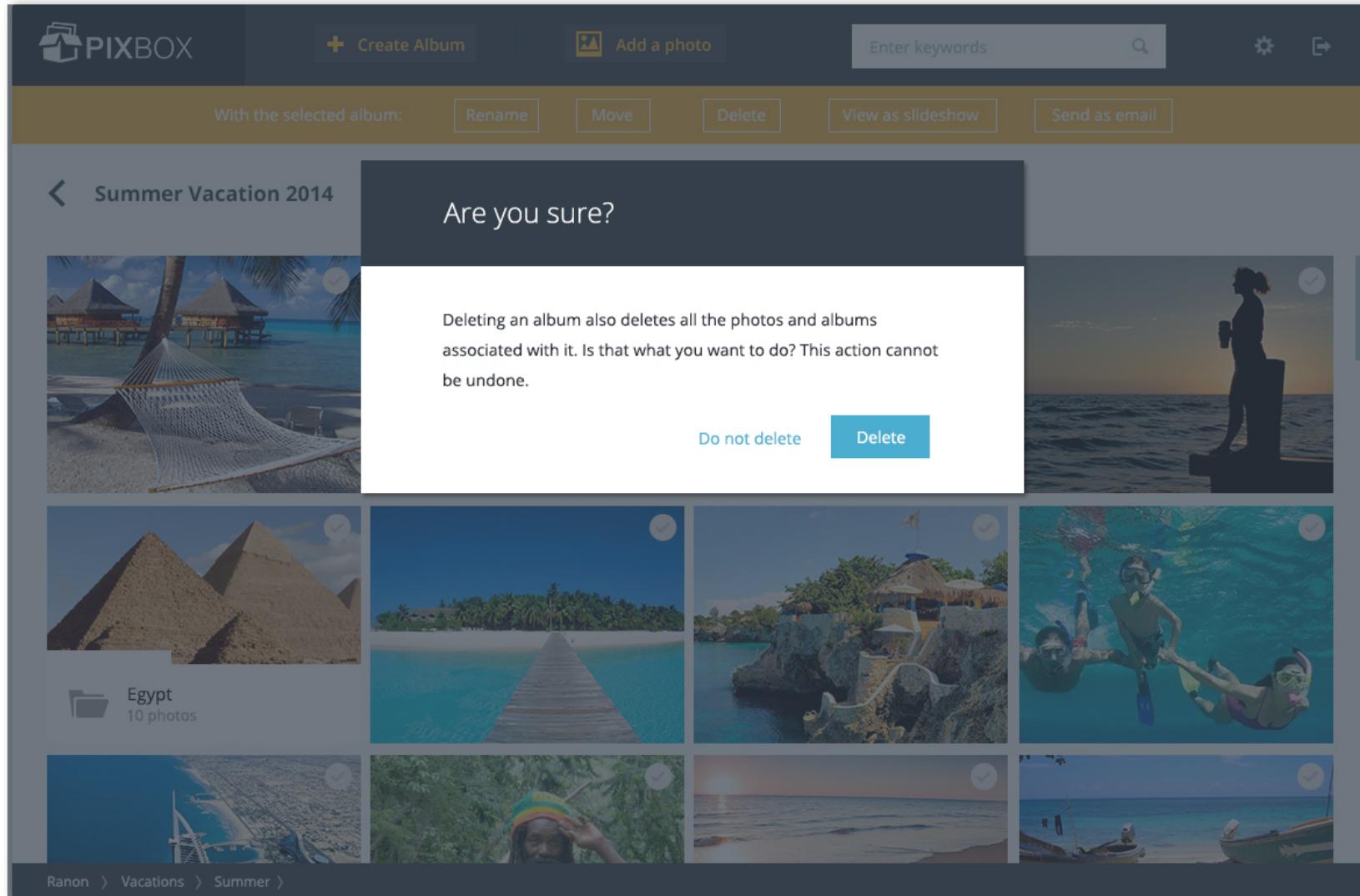
3.1.13 Prototype 1.13: Move photo album



3.1.14 Prototype 1.14: Rename photo album

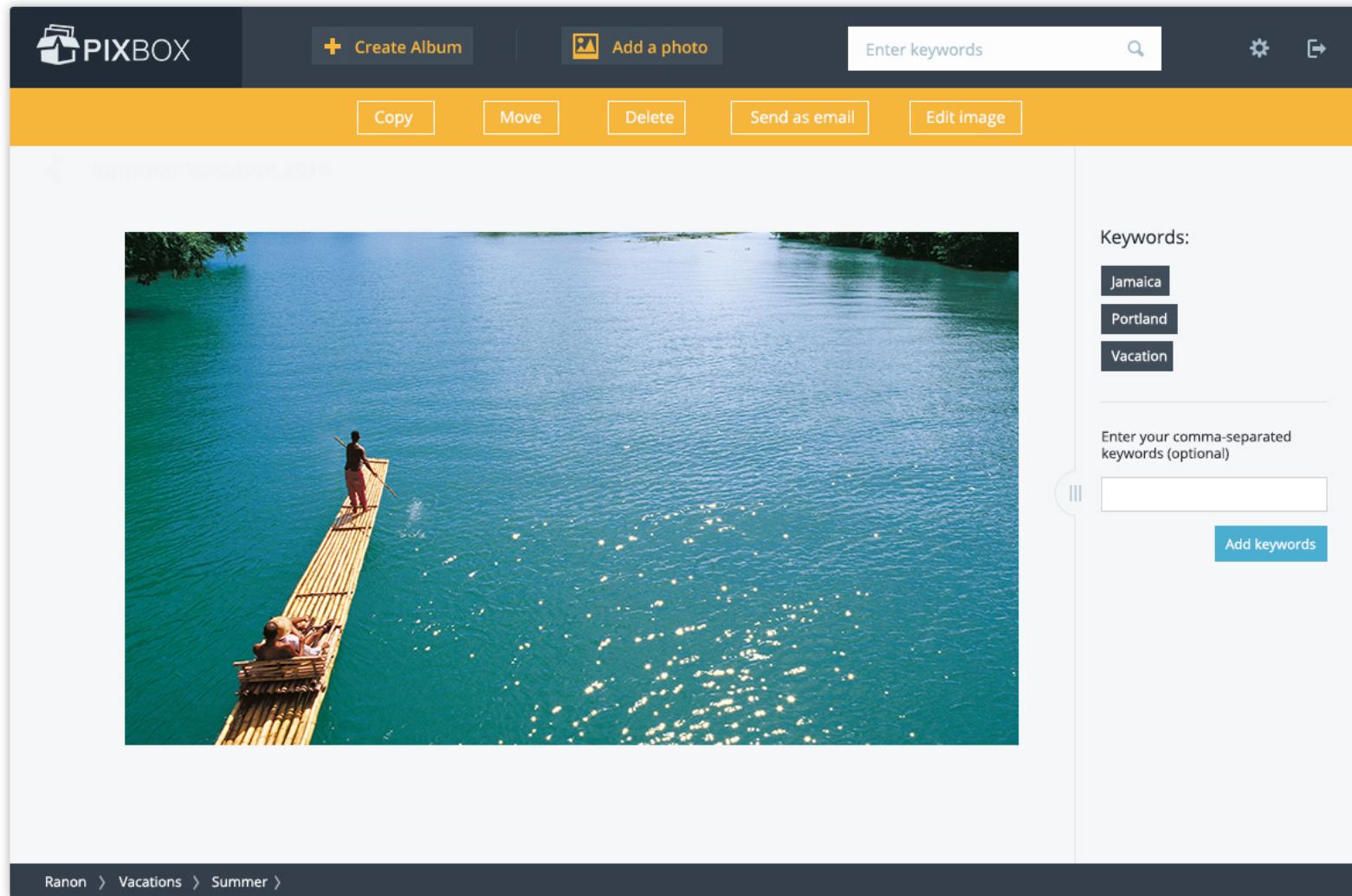


3.1.15 Prototype 1.15: Delete photo album



3.2 Photo viewing

3.2.1 Prototype 2.1: View a single photo



The screenshot shows a user interface for a photo viewing application named PIXBOX. At the top, there is a dark header bar with the PIXBOX logo on the left, followed by buttons for 'Create Album' (with a plus icon), 'Add a photo' (with a camera icon), a search bar labeled 'Enter keywords' with a magnifying glass icon, and settings/exit icons on the right.

Below the header is a yellow navigation bar containing five buttons: 'Copy', 'Move', 'Delete', 'Send as email', and 'Edit image'. The main content area displays a large, vibrant photograph of two people on a long, narrow bamboo raft in a bright blue body of water. One person is seated at the front, and another is standing at the rear, holding a long wooden pole.

To the right of the photo, there is a sidebar with the heading 'Keywords:' followed by three solid black rectangular buttons with white text: 'Jamaica', 'Portland', and 'Vacation'. Below this is a text input field with placeholder text 'Enter your comma-separated keywords (optional)' and a blue 'Add keywords' button.

At the bottom of the page, a dark footer bar contains the text 'Ranon > Vacations > Summer >'.

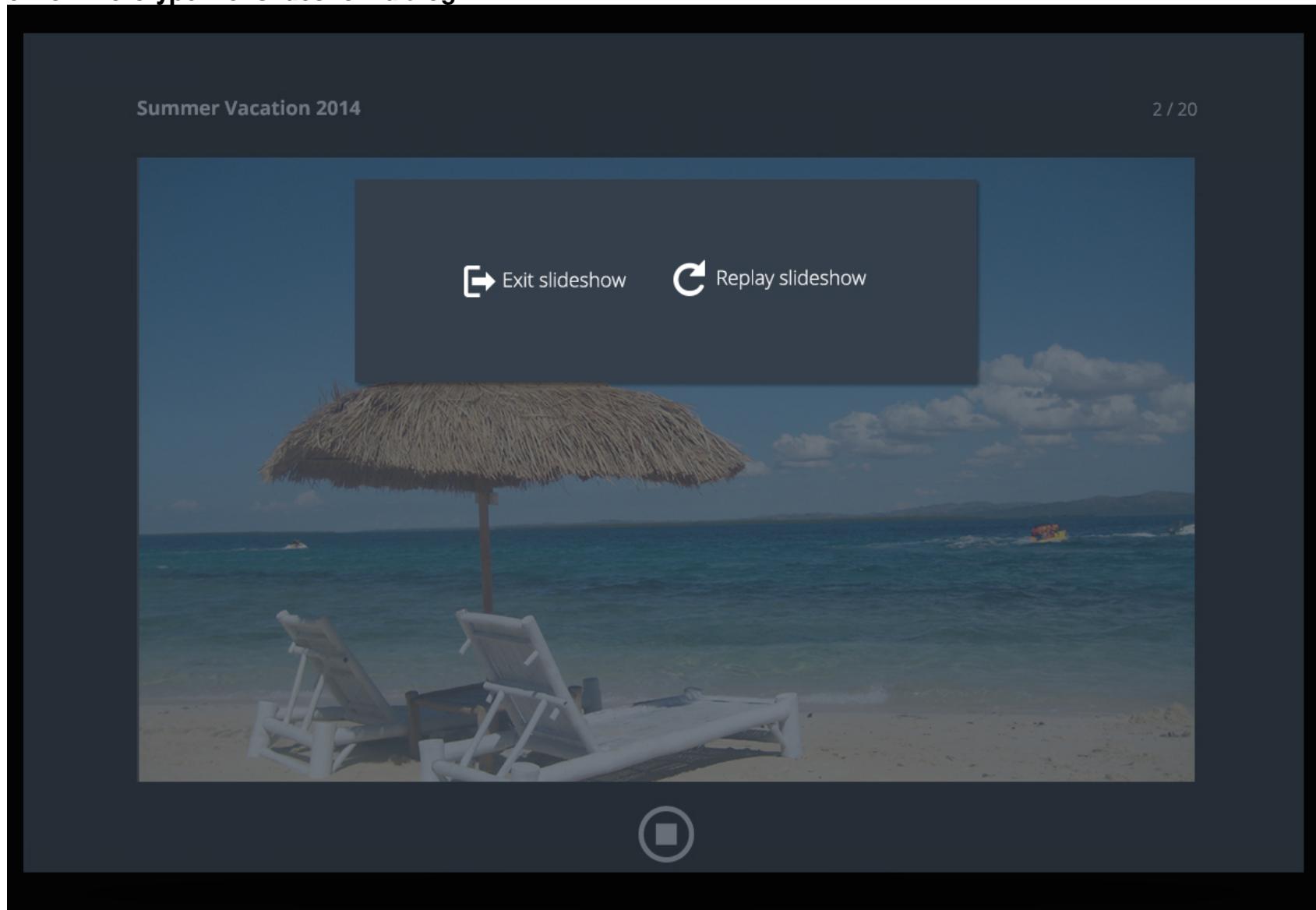
3.2.2 Prototype 2.2: View an entire photo album of photos as a slideshow

Summer Vacation 2014

2 / 20

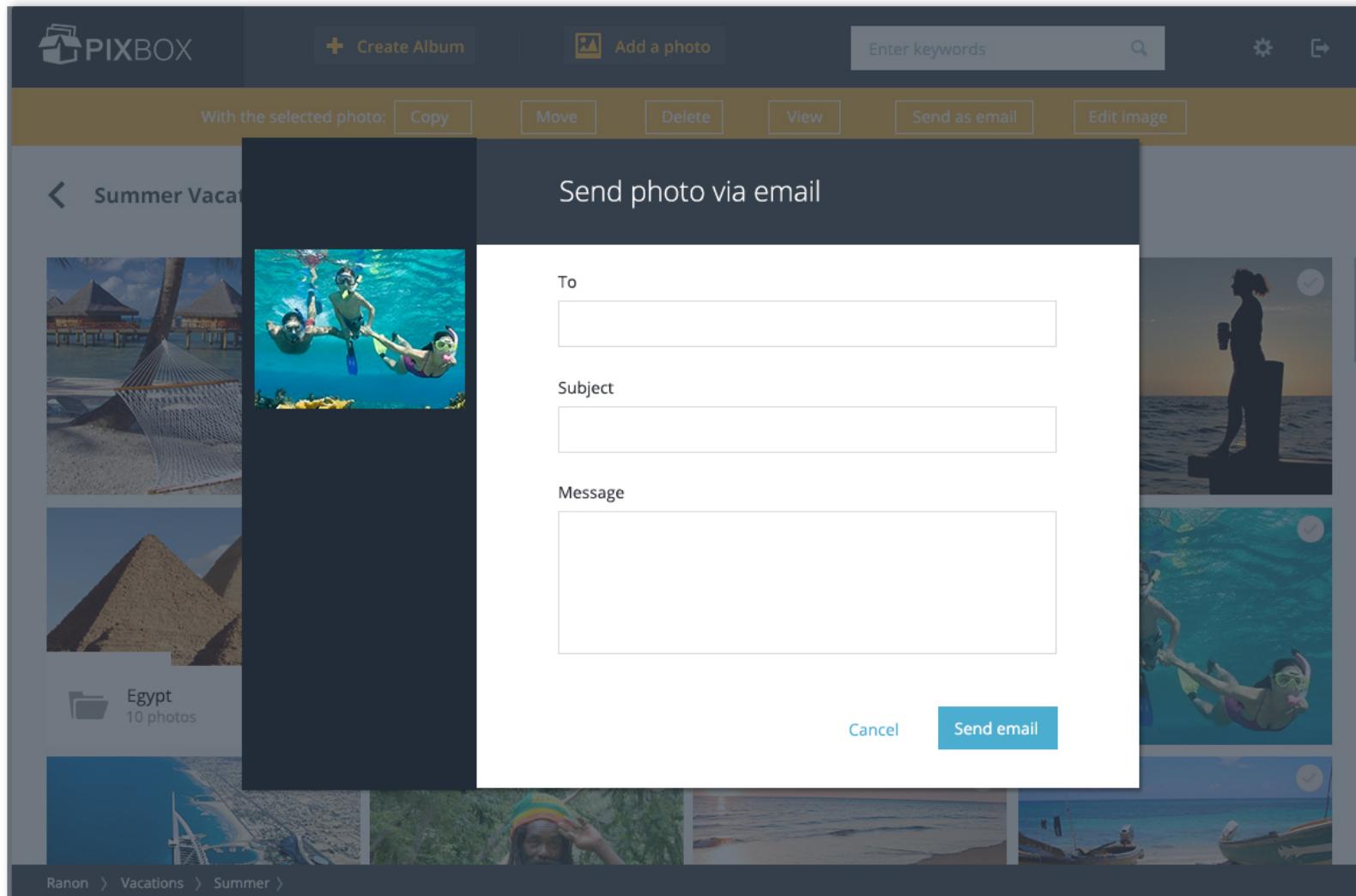


3.2.3 Prototype 2.3: Slideshow dialog

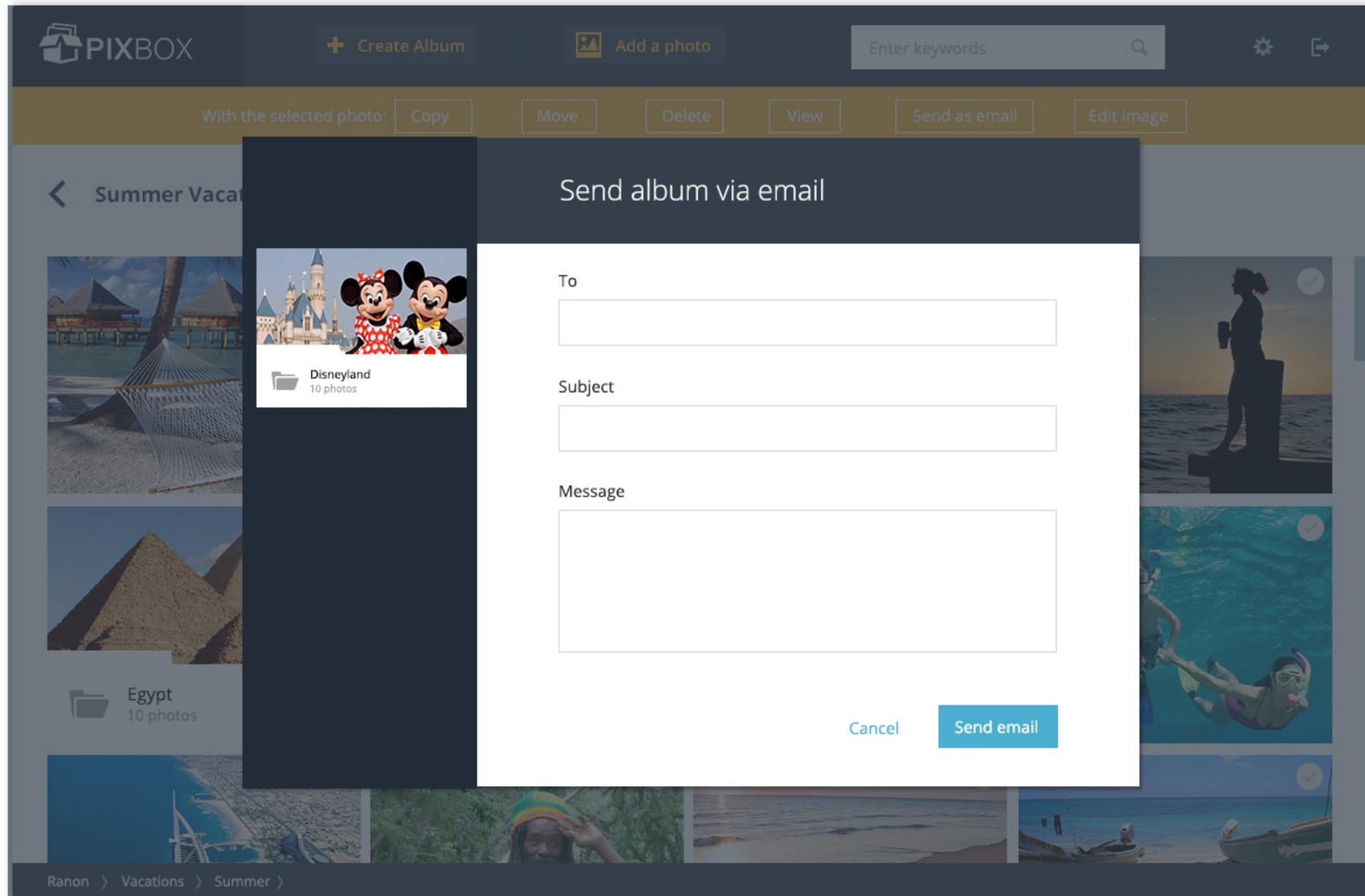


3.3 Photo sharing

3.3.1 Prototype 3.1: Send a single photo as an email

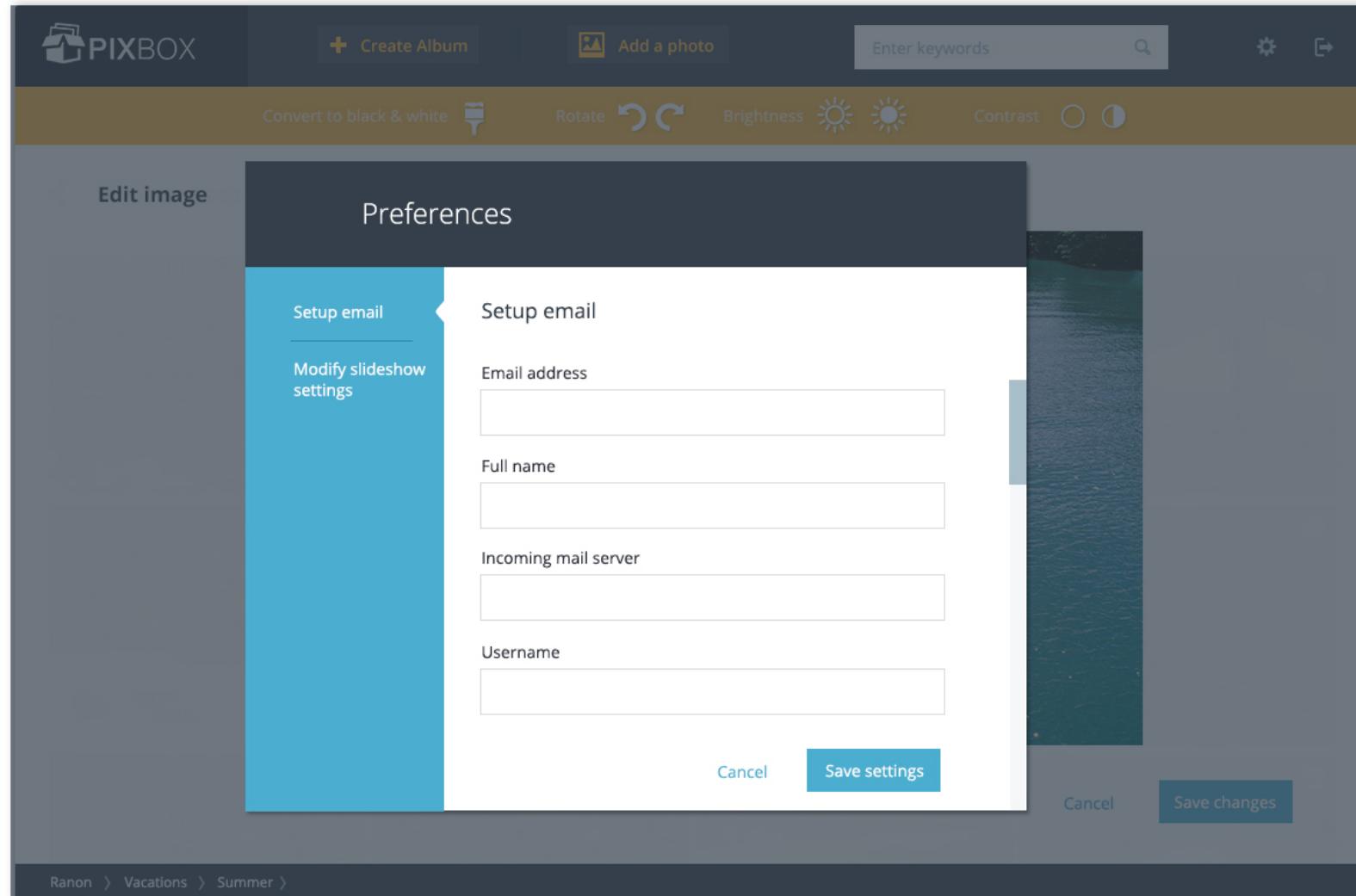


3.3.2 Prototype 3.2: Send an entire photo album of photos as an email

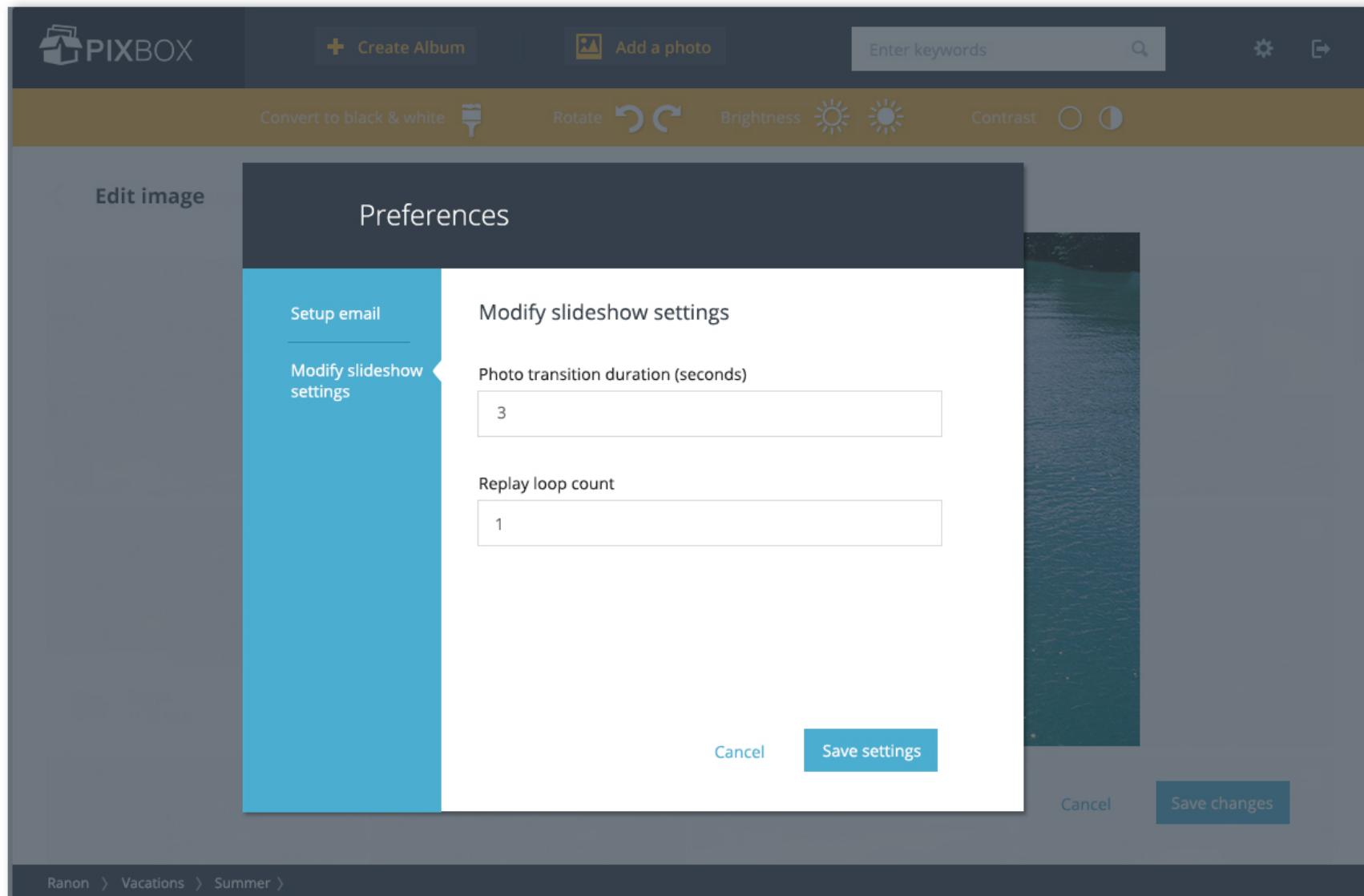


3.4 Preference configuration

3.4.1 Prototype 4.1: Email setup



3.4.2 Prototype 4.2: Modify slideshow settings



3.5 Search

3.5.1 Prototype 5.1 Search results

The screenshot shows a user interface for a photo sharing platform. At the top, there's a dark header bar with the logo 'PIXBOX', a 'Create Album' button, an 'Add a photo' button, a search bar containing the text 'Sunshine', and various navigation icons. Below the header, the title 'Summer Vacation 2014' is displayed, followed by the subtitle 'Search results for 'Sunshine''. The main area contains a grid of 15 thumbnail images. The first row includes: a hammock on a beach, a large swimming pool with yellow umbrellas, a pair of colorful flip-flops next to a coconut, and an aerial view of a resort with pools and palm trees. The second row includes: the Great Pyramids of Giza, a wooden pier extending into clear blue water, a thatched-roof hut built into a rocky cliffside, and three people snorkeling in the ocean. The third row includes: an aerial view of a coastal city with a prominent skyscraper, a person wearing a colorful rainbow hat, a sunset over a beach, and a small boat on the water. Each thumbnail has a circular checkmark icon in the top right corner.

3.6 Basic photo editing

3.6.1 Prototype 6.1 Edit photo image

PIXBOX

+ Create Album

Add a photo

Enter keywords

Convert to black & white

Rotate

Brightness

Contrast

Edit image

4 Risk Analysis

This section outlines the risks associated with this software project.

Risk	Dimension
Continually changing requirements	Requirements
System requirement not adequately identified.	Requirements
Unclear system requirements	Requirements
Incorrect system requirements	Requirements
Project progress not monitored closely enough.	Planning and control
Poor project planning	Planning and control
Ineffective communications	Planning and control
Personnel shortfalls	Team
Absent team member	Team
Unrealistic time and cost estimates	Planning and control
Developing the wrong software	Requirements
Developing the wrong user interface	Requirements
Not managing change properly	Planning and control
Failure to manage end-user expectations	User
Lack of required knowledge or skill	Team
Changes to team	Team