PhotoDroid

William DeRaad, Russ Mehring, Derek Reiersen

Summary

PhotoDroid is android image processing app which can be used to take and manipulate images with various image processing techniques. PhotoDroid is designed to support all modern android platforms. PhotoDroid leverages the powerful image processing tools available in the OpenCV library.

Requirements

Business Requirements

ID	Requirement	Topic Area	User	Priority
BR-01	**App must be downloadable on android marketplace**	**Accessability**	**Clients**	**Medium**
BR-02	**App must have appropriate icon on marketplace**	**Marketing**	**System**	**Medium**

User Requirements

ID	Requirement	Topic Area	User	Priority
UR-01	**User can take a photograph with app**	**Basic usability**	**App User**	**Critical**
UR-02	**User can load image from gallery**	**Basic usability**	**App User**	**Critical**
UR-03	**User can perform image processing on an image**	**Basic usability**	**App User**	**Critical**
UR-04	**User can save image to gallery**	**Basic usability**	**App User**	**Critical**

Functional Requirements

ID	Requirement	Topic Area	User	Priority
FR-01	**App will display image on screen that is being edited**	**Basic usability**	**System**	**Critical**
FR-02	**App will display editing gui to user when image is being edited**	**Basic usability**	**System**	**Critical**
FR-03	**App will apply selected image processing features to image**	**Basic usability**	**System**	**Critical**
FR-04	**App will save image to gallery when requested**	**Storing images**	**System**	**Critical**
FR-05	**Camera Clicks when Photo Taken**	**User Feedback**	**App User**	**Medium**

Non-Functional Requirements

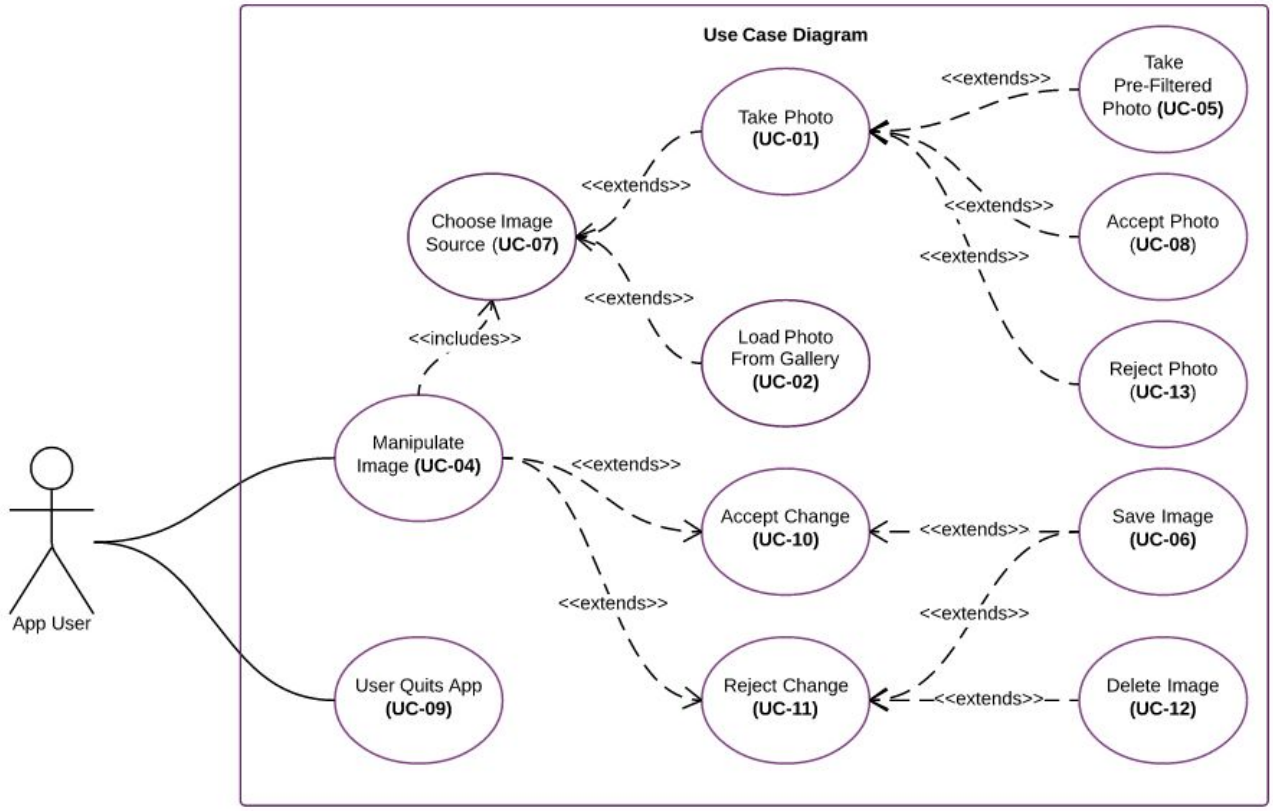
ID	Requirement	Topic Area	User	Priority
NFR-01	**App must work on all updated android devices**	**Accessability**	**System**	**Medium**
NFR-02	**App must not hang when performing image processing**	**Usability & Concurrency**	**System**	**High**

Users and Tasks

Users

Anyone who has an android capable device, but in particular an android user who is interested in image processing.

Tasks



(UC-01) User takes a new photo

Use Case ID:	UC-01
Date:	10-2-2015
Use Case Name:	Take Photo
Description:	User Takes a photo with the android camera
Actors:	Android App User
Preconditions:	Phone has working camera
Postconditions:	Photo will be displayed for accept / reject photo
Frequency of	High

<table>
<tr><td>Use:</td><td></td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>User presses new photo</td><td>Open android camera</td></tr>
<tr><td>2.</td><td>User uses android camera to frame desired image</td><td>Image is changed according to camera</td></tr>
<tr><td>3.</td><td>User presses take photo</td><td>Camera autofocuses, photo is taken</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>User adds a pre-filter to photo before it is taken</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>None</td></tr>
</table>

(UC-02) User uploads a photo from the gallery

Use Case ID:	UC-02
Date:	10-6-2015
Use Case Name:	Load photo from gallery
Description:	User uploads a photo from the gallery
Actors:	Android App User
Preconditions:	Photo exists in gallery User has allowed app access to the photo gallery User has chosen to load photo from gallery
Postconditions:	Photo will be copied into app local memory User decides to accept or reject image

<table>
<tr><td>Frequency of Use:</td><td>High</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>User opens gallery</td><td>Display photos saved in gallery</td></tr>
<tr><td>2.</td><td>User selects photo</td><td>Load image into editing mode</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>User leaves while in the process of selecting a photo, app resets to initial state.</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>There is probably android photo manager that handles selecting the photo, we should only have to call a function and expect a return image.</td></tr>
</table>

(UC-03) [Strech goal] User takes a video performs image processing and then saves the video to gallery

Use Case ID:	UC-03
Date:	9-28-2015
Use Case Name:	Perform Image Processing on a video.
Description:	User takes a video, performs image processing and then saves the video to gallery.
Actors:	Android App User
Preconditions:	None
Postconditions:	Edited video will be saved in gallery
Frequency of	N.A

<table>
<tr><td>Use:</td><td></td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>User takes video</td><td>Display camera UI</td></tr>
<tr><td>2.</td><td>User selects to edit video and not discard</td><td>Load video into editing mode</td></tr>
<tr><td>3.</td><td>User edits video</td><td>Video is edited on screen</td></tr>
<tr><td>4.</td><td>User saves video</td><td>Video is stored in android gallery</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>User leaves while editing video, partially edited video is saved in app memory (while app is running in the background)</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>None</td></tr>
</table>

(UC-04) User manipulates image

Use Case ID:	UC-04
Date:	10/04/2015
Use Case Name:	User manipulates image
Description:	The main functionality of the application, in which the user can choose which image processing filters they would like to apply to a photo.
Actors:	Android App User
Preconditions:	- User has app running on an Android device - User has selected photo to manipulate
Postconditions:	- User can save/view their manipulated image

<table>
<tr><td>Frequency of Use:</td><td>High</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Select photo</td><td>User takes a new photo or selects one from the gallery.</td></tr>
<tr><td>2.</td><td>User applies filter</td><td>Filtered image is displayed</td></tr>
<tr><td>3.</td><td>Save image</td><td>Save edited image to file system with feedback.</td></tr>
<tr><td>4.</td><td>Delete image</td><td>Restore and display original image.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>None</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>None</td></tr>
</table>

(UC-05) User takes filtered photo

Use Case ID:	UC-05
Date:	10-2-2015
Use Case Name:	Take filtered photo
Description:	User takes a photo with an image filter applied beforehand
Actors:	Android App User

<table>
<tr><td>Preconditions:</td><td>Phone has working camera
User has OpenCV Manager installed
User has app installed on Android device</td></tr>
<tr><td>Postconditions:</td><td>User will have a new photo with at least one filter applied to the photo
User can save the photo to their gallery
Photo will be displayed for accept/reject photo</td></tr>
<tr><td>Frequency of Use:</td><td>N/A</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>User presses new photo</td><td>Open android camera</td></tr>
<tr><td>2.</td><td>User selects filter to apply to camera</td><td>Screen shows filtered camera feed</td></tr>
<tr><td>3.</td><td>User adjusts camera until satisfied with image</td><td>Image is changed according to camera</td></tr>
<tr><td>4.</td><td>User presses take photo</td><td>Camera autofocuses, photo is taken</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>Is a modified version of take photo</td></tr>
<tr><td>Notes and Issues:</td><td>See UC-01</td></tr>
<tr><td>Developer Notes:</td><td>None</td></tr>
</table>

(UC-06) User saves image

Use Case ID:	UC-06
Date:	10/6/2015

<table>
<tr><td><u>Use Case Name:</u></td><td>Save Image</td></tr>
<tr><td><u>Description:</u></td><td>User saves the image to the android gallery</td></tr>
<tr><td>Actors:</td><td>Android App User</td></tr>
<tr><td>Preconditions:</td><td>User is manipulating an image</td></tr>
<tr><td>Postconditions:</td><td>Image is saved in gallery, overwriting the previous manipulated image</td></tr>
<tr><td>Frequency of Use:</td><td>Whenever user wants to save their progress when manipulating an image</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>User presses save</td><td>Ask if user is sure</td></tr>
<tr><td>2.</td><td>User confirms</td><td>System saves image</td></tr>
<tr><td>3.</td><td>User Declines</td><td>System does nothing</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>None</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>None</td></tr>
</table>

(UC-07) User can select image source

Use Case ID:	UC-07
Date:	10-6-2015
Use Case Name:	Choose image source
Description:	User decides whether they would like to take a new photo or choose an

<table>
<tr><td></td><td>existing photo from the gallery.</td></tr>
<tr><td>Actors:</td><td>Android App User</td></tr>
<tr><td>Preconditions:</td><td>- User has allowed app access to the camera
- User has allowed app access to the photo gallery</td></tr>
<tr><td>Postconditions:</td><td>- User will be moved</td></tr>
<tr><td>Frequency of Use:</td><td>High</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1a.</td><td>User selects to take a new photo.</td><td>Camera is opened.</td></tr>
<tr><td>1b.</td><td>User select to choose an existing photo</td><td>Load Photo functionality is triggered.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>None</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>Note (Probably too early in the design process):

- A lot of the photo sharing/processing apps with strong UI design (at least for iOS), have the functionality where the initial state (which is this UC) is in the camera mode with a square in the bottom corner to go to the gallery.

- We might be able to extend this idea by having various other radio buttons (for lack of a better term) that users can toggle for pre-filtering types. This reduces UI clicks and then would reduce three of four UC’s into one.</td></tr>
</table>

(UC-08) User can accept the photo taken

Use Case ID:	UC-08

<table>
<tr><td><u>Date:</u></td><td>10/6/2015</td></tr>
<tr><td><u>Use Case Name:</u></td><td>Accept Photo</td></tr>
<tr><td><u>Description:</u></td><td>User Accepts the photo they have taken</td></tr>
<tr><td>Actors:</td><td>Android App User</td></tr>
<tr><td>Preconditions:</td><td>User has taken a photo</td></tr>
<tr><td>Postconditions:</td><td>Photo is loaded and ready to be manipulated</td></tr>
<tr><td>Frequency of Use:</td><td>High</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>User accepts photo</td><td>System loads photo into the manipulate image interface.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>None</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>None</td></tr>
</table>

(UC-09) User can close the application at any time

Use Case ID:	UC-09
Date:	10/07/15
Use Case Name:	User Quits App
Description:	User closes the application
Actors:	Android App User

<table>
<tr><td>Preconditions:</td><td>App is running</td></tr>
<tr><td>Postconditions:</td><td>App closes</td></tr>
<tr><td>Frequency of Use:</td><td>High</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Selects ‘Exit’ button</td><td>If work in progress, app attempts to maintain state.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>App crashes (i.e. closes against users will), user forces their phone to restart (effectively crashing the app also)</td></tr>
<tr><td>Notes and Issues:</td><td>None</td></tr>
<tr><td>Developer Notes:</td><td>None</td></tr>
</table>

(UC-10) User can accept changes to their image

Use Case ID:	UC-10
Date:	10-8-2015
Use Case Name:	Accept Change
Description:	User accepts the changes they have made by manipulating an image.
Actors:	Android App User
Preconditions:	- User has manipulation an image
Postconditions:	- Image remains in its current state
Frequency of Use:	High

Flow of Events:	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Chooses to accept changes</td><td>Image remains in its current modified state</td></tr></table>
Variations:	None
Notes and Issues:	None
Developer Notes:	None

(UC-11) User can reject changes

Use Case ID:	UC-11
Date:	10-9-2015
Use Case Name:	Reject Change
Description:	User rejects the changes they have made by manipulating an image.
Actors:	Android App User
Preconditions:	- User has manipulation an image
Postconditions:	- Image is reset to its previous state
Frequency of Use:	High
Flow of Events:	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Chooses to reject changes they have made.</td><td>Image is restored to its previous state</td></tr></table>

Variations:	None
Notes and Issues:	None
Developer Notes:	None

(UC-12) User can delete an image

Use Case ID:	UC-12
Date:	10-8-2015
Use Case Name:	Delete Image.
Description:	User deletes an image.
Actors:	Android App User
Preconditions:	- User has manipulation an image.
Postconditions:	- App will reset to initial state.
Frequency of Use:	High
Flow of Events:	

	Actor Action	**System Response**
1.	User chooses to delete an image.	The app's local copy of the image is deleted and the app resets to its initial state.

Variations:	None
Notes and	None

Issues:	
Developer Notes:	None

(UC-13) User can reject the photo taken

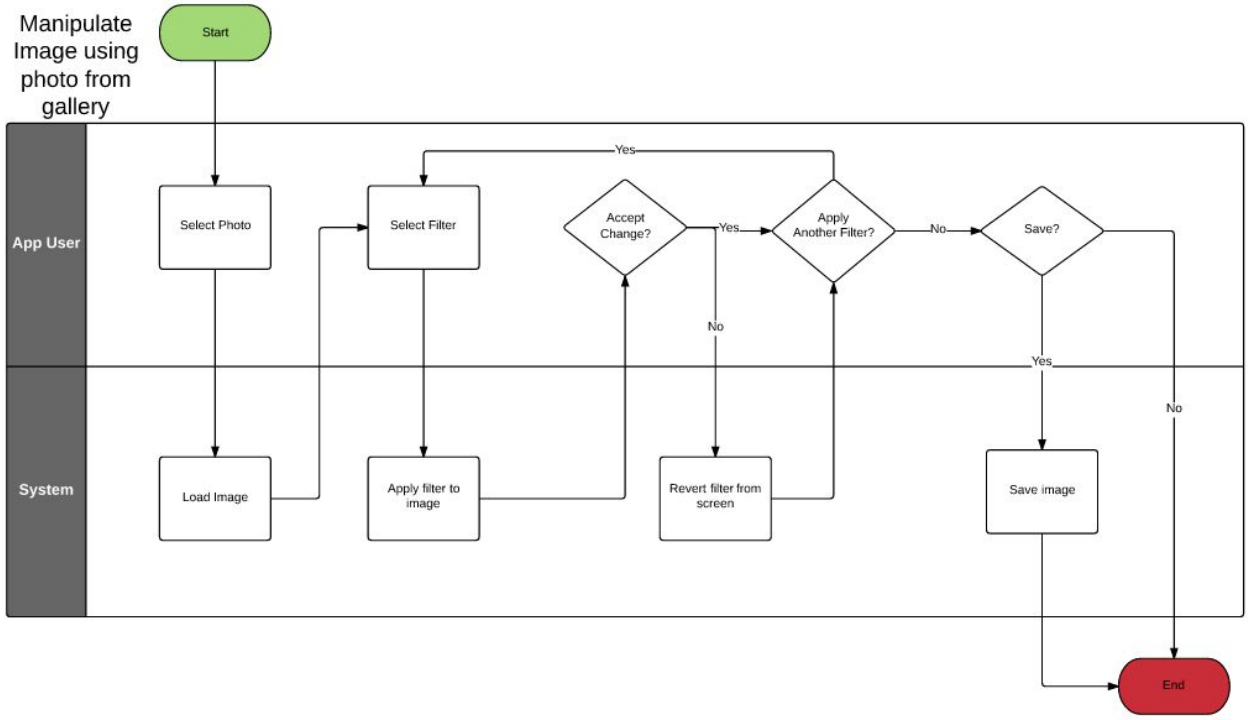
Use Case ID:	UC-13
Date:	10/9/2015
Use Case Name:	Reject Photo
Description:	User rejects the photo they have taken
Actors:	Android App User
Preconditions:	User has taken a photo
Postconditions:	System is reset to previous state (i.e. user can take a new photo)
Frequency of Use:	High
Flow of Events:	

	Actor Action	**System Response**
1.	User rejects photo	System removes taken photo; system is reset to previous state.

Variations:	None
Notes and Issues:	None
Developer Notes:	None

Activity Diagram

User takes a photo, performs image processing and then saves the photo to gallery.



Data Storage

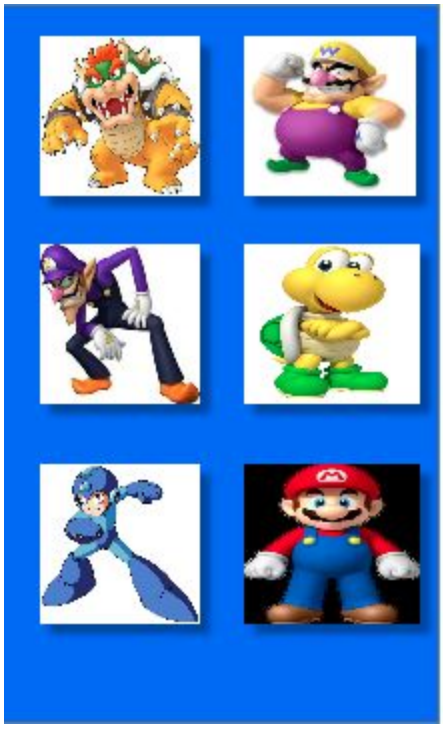
Android file system both internal (always available, accessible by only your app) and external (not always available, world-readable).

UI Mockups

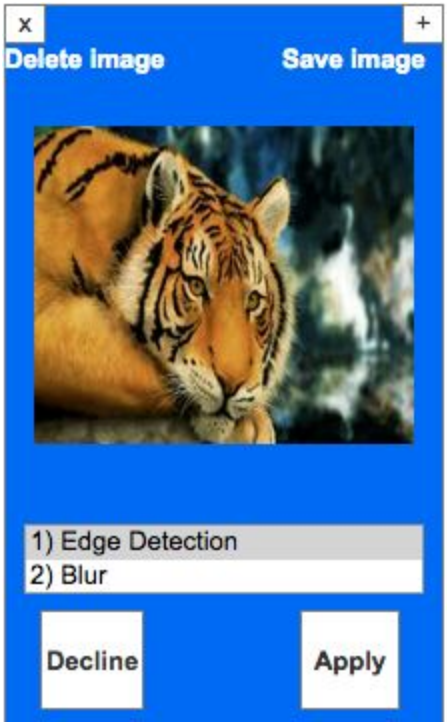

PhotoDroid
Take New Photo
Load Photo
Quit


Take Photo


Reject
Accept



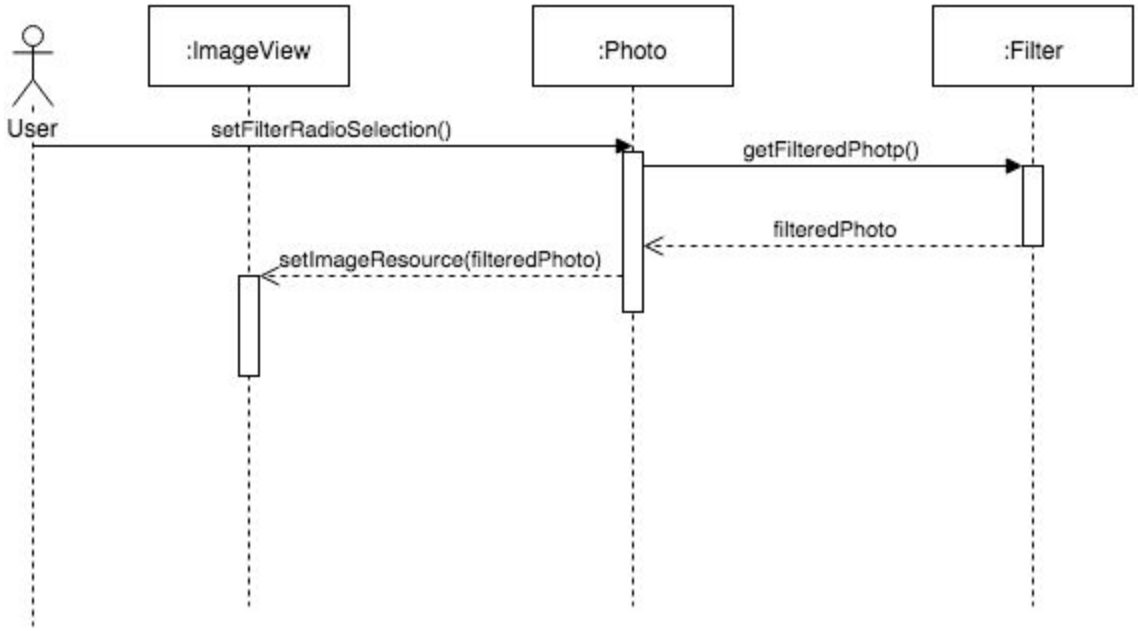

Reject
Accept


x
+
Delete Image
Save Image
1) Edge Detection
2) Blur
Decline
Apply

User Interactions

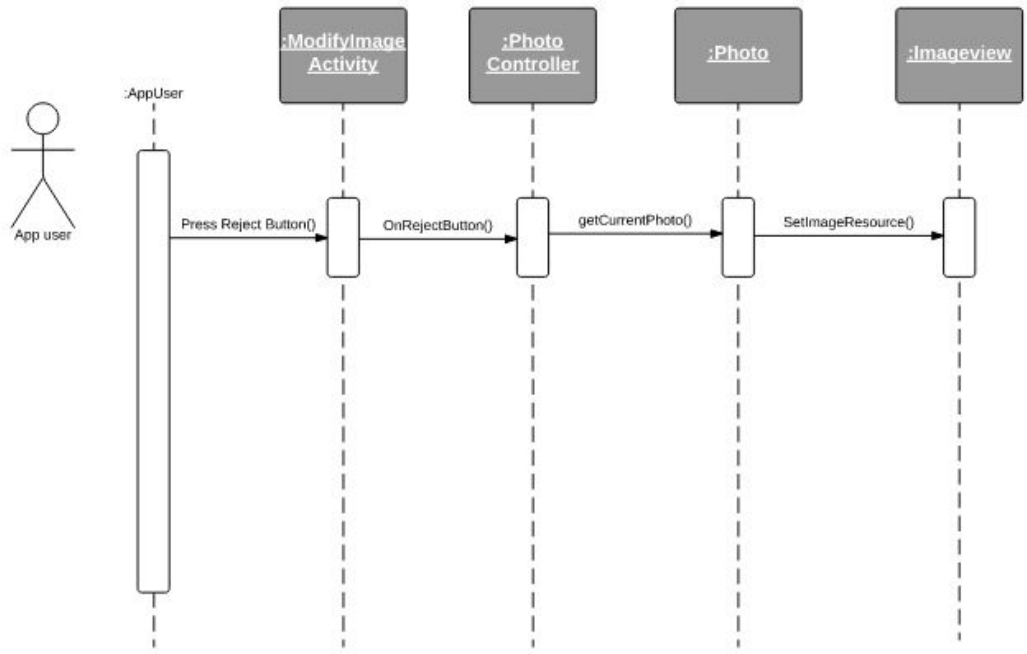
User Interaction for (UC-04)

The user can manipulate an image by making a radio selection of which filter they would like to apply. We will support this by applying the selected filter to the current image, and then updating the app's view interface.



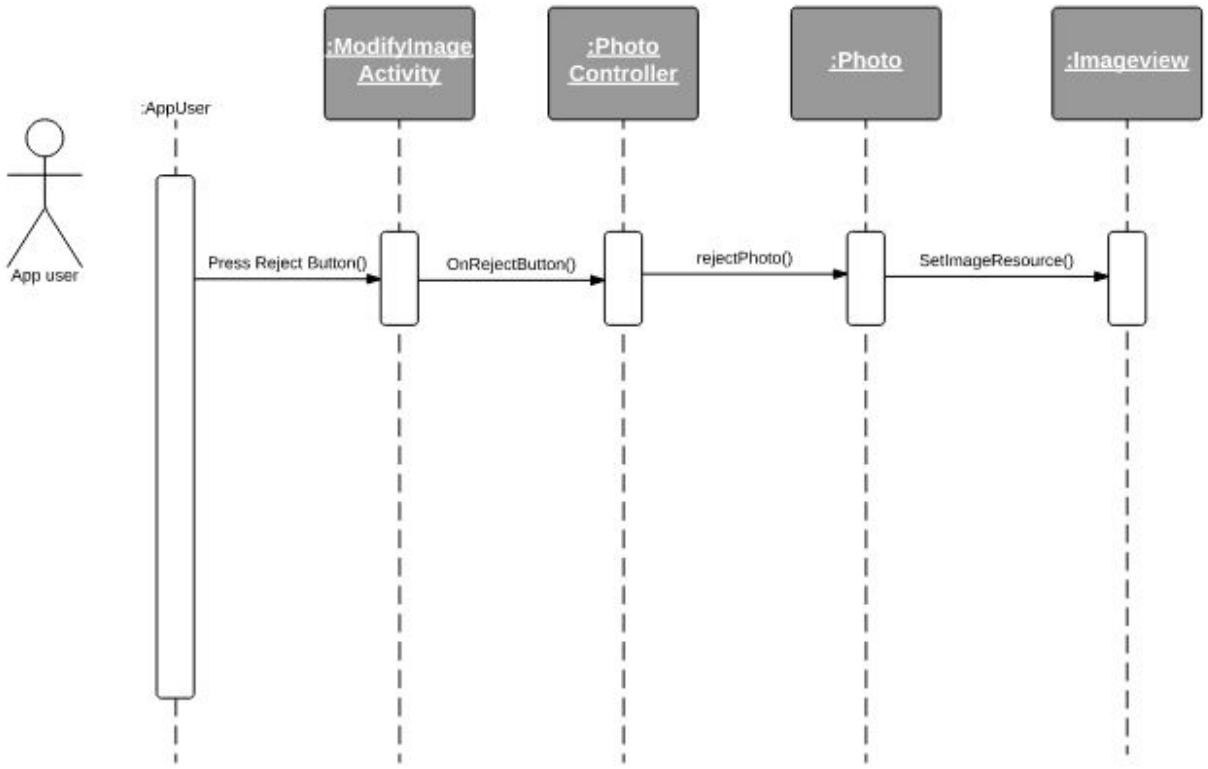
User Interaction for (UC-11) Variation 1

The user can reject a change they made in the manipulate image activity. This is supported by the filter being removed and the current image being displayed.



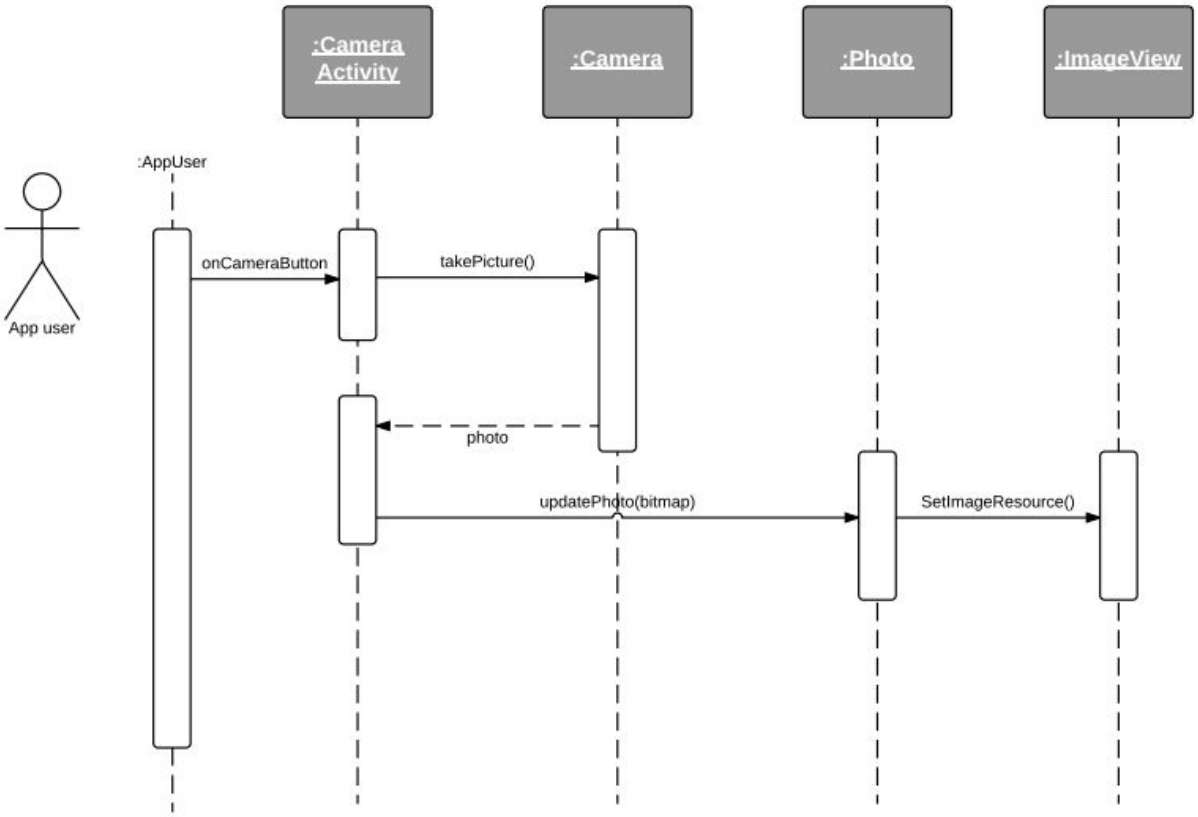
User Interaction for (UC-11) Variation 2

The user can reject a change they made in the manipulate image activity even when a filter hasn’t been selected, this will revert to the previous image.


:ModifyImage Activity
:Photo Controller
:Photo
:Imageview
:AppUser
App user
Press Reject Button()
OnRejectButton()
rejectPhoto()
SetImageResource()

User Interaction for (UC-01)

The user can press the take photo button. This will capture the current image the camera is capturing.



Class Diagrams

Camera Activity and Take Photo Class Diagram

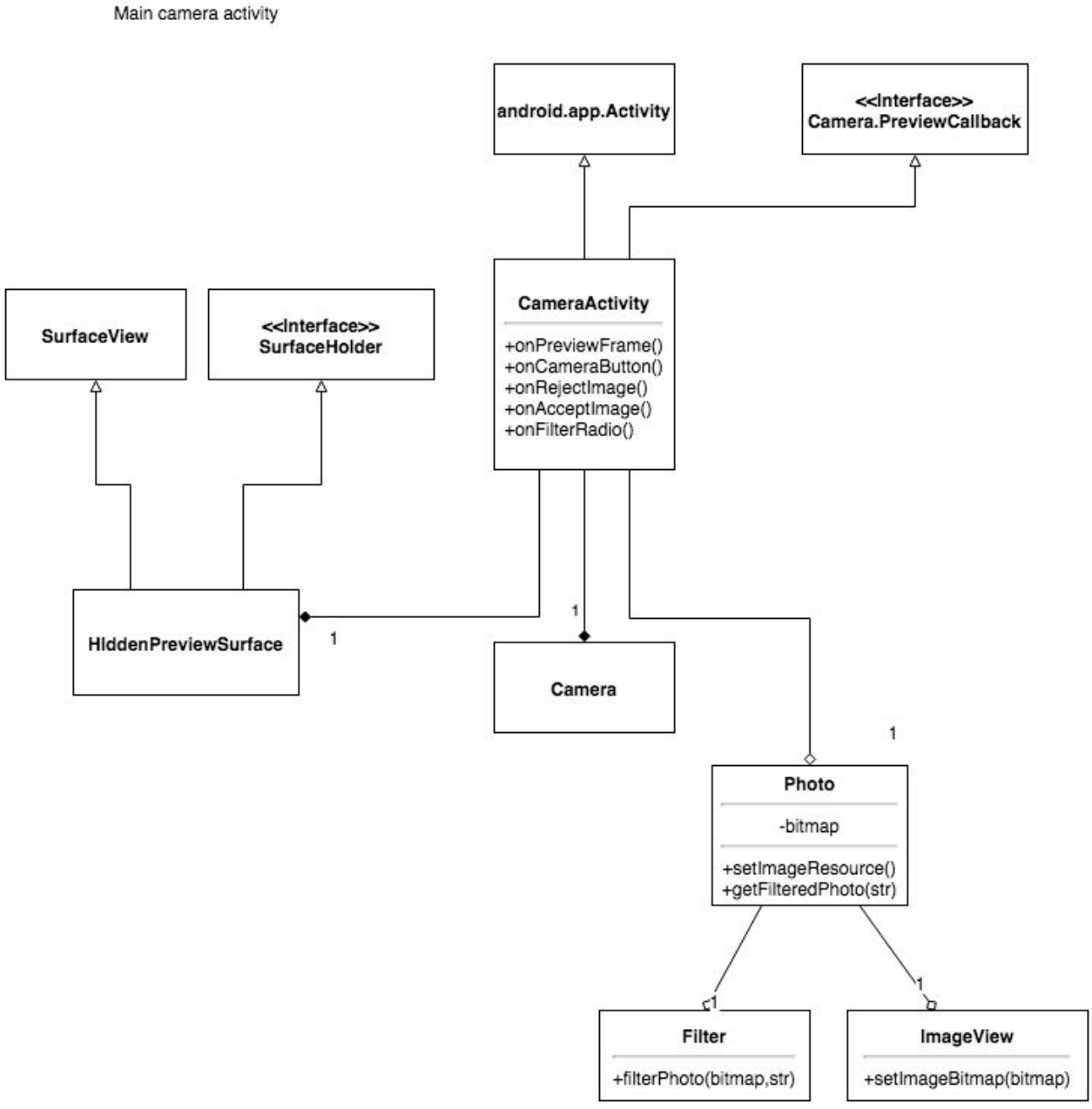


Image Manipulation Model-View-Controller Classes

Manipulate Image Activity Class Diagram