Digikam/GSoC2012/CameraUserInterfaceRevamp

Contents

Digikam GSoC 2012 Camera User Interface Revamp

This page is meant to gather information about the GSoC 2012 Idea, Camera User Interface Revamp.

Requirements and Considerations

Summary

digiKam features a graphical interface to access and download pictures from digital cameras. Code is rather old, using Qt3Support classes for the icon view, the UI code intermangled deeply with backend code, and has not seen very much care and love for some years.

This project would involve taking the old code apart, rewriting a clean code base backend and front-end, but also adding user interface elements to make the most important everyday task as easy as possible.

Benefits to Community

In the next major Qt release before Augest the Qt3 support classes will be deprecated (source), and the current implementation of the camera interface is depending on q3support classes, so it is necessarily to port this interface to the new Qt 4 model/view.

After revamping the camera interface, users will be able to use it in a similar manner to the album gui. That will introduce some new UX features described below in the mockups.

In this way, 4 bugs marked as Qt model/view will be fixed: 121310 [0], 126149 [1], 158437 [2], 216491 [3].

Description & Related Work

Introduction

Camera interface is the name of the GUI that opens when user try to import some photos from different places like cameras, hard disk folders, USB storage devices, scanners, or remote computers. So this Interface is used in a generic manner to import photos, and I find it better to change its name from ***Camera Interface*** to ***Import Interface*** which apparently fits its usage better.

Deliverables

As mentioned before this project aims to port the (Camera Interface) which currently using Qt3support classes into Qt4 Model/View, the goal of model/view port is to be able to:

1. Factorize a lots of code shared with album GUI.
2. Remove Qt3 dependency.
3. Create more powerful icon-view for import interface, with advanced filtering option, group of items (RAW+JPEG), pre-tags options, etc...
4. To be able to use preview mode in camera interface (image + video), as in albumgui.

Functionality Listing

Here are the expected primary functionalities in the Import Interface.

1. Cancel importing
2. Hide imported photos
3. Upload photos
4. Toggle Lock
5. Mark as downloaded
6. Select/Filter type of files to be shown
7. Select (all, none, locked items, new Items)
8. Invert selection
9. Thumbnails
10. View Image
11. Delete (new, selected, all)
12. Show History
13. Full screen mode

14. Close

New Features

1. Download to (new, last, existing) album
2. Pause Importing see mockup number 3
3. Assign Tags see mockups number 5, 5.1
4. Remove Tags
5. Assign Labels
6. Group Selected photos (by time, etc) see mockup number 6
7. Back to icon view
8. Next Image (Back)
9. Previous image (Forward)
10. Fit image to window
11. Editing Icon view Settings, see mockup number 7.1
12. Sorting Images (flat list, by format)
13. Preview Videos see mockup number 4
14. Map view and improving the current map sidebar widget (Geolocation) see mockup number 10

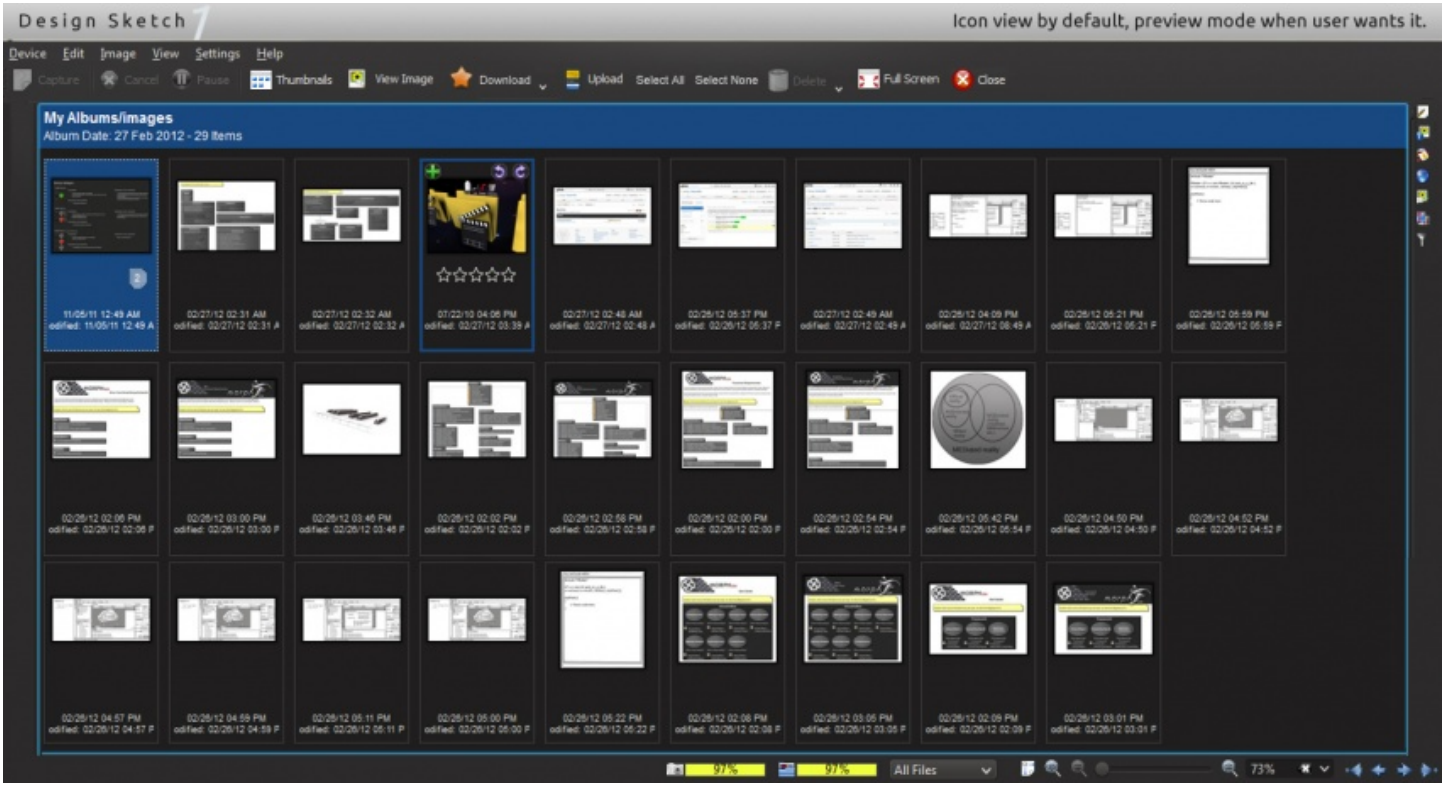
Design

Because one photo is better than thousand words, I made these mockups!

UI/UX Design

1. **Icon view by default, preview mode when user wants it.**

In the new import interface, there will be an icon view by default, and when clicking on any photo the interface will be switched to preview mode as in the next image.

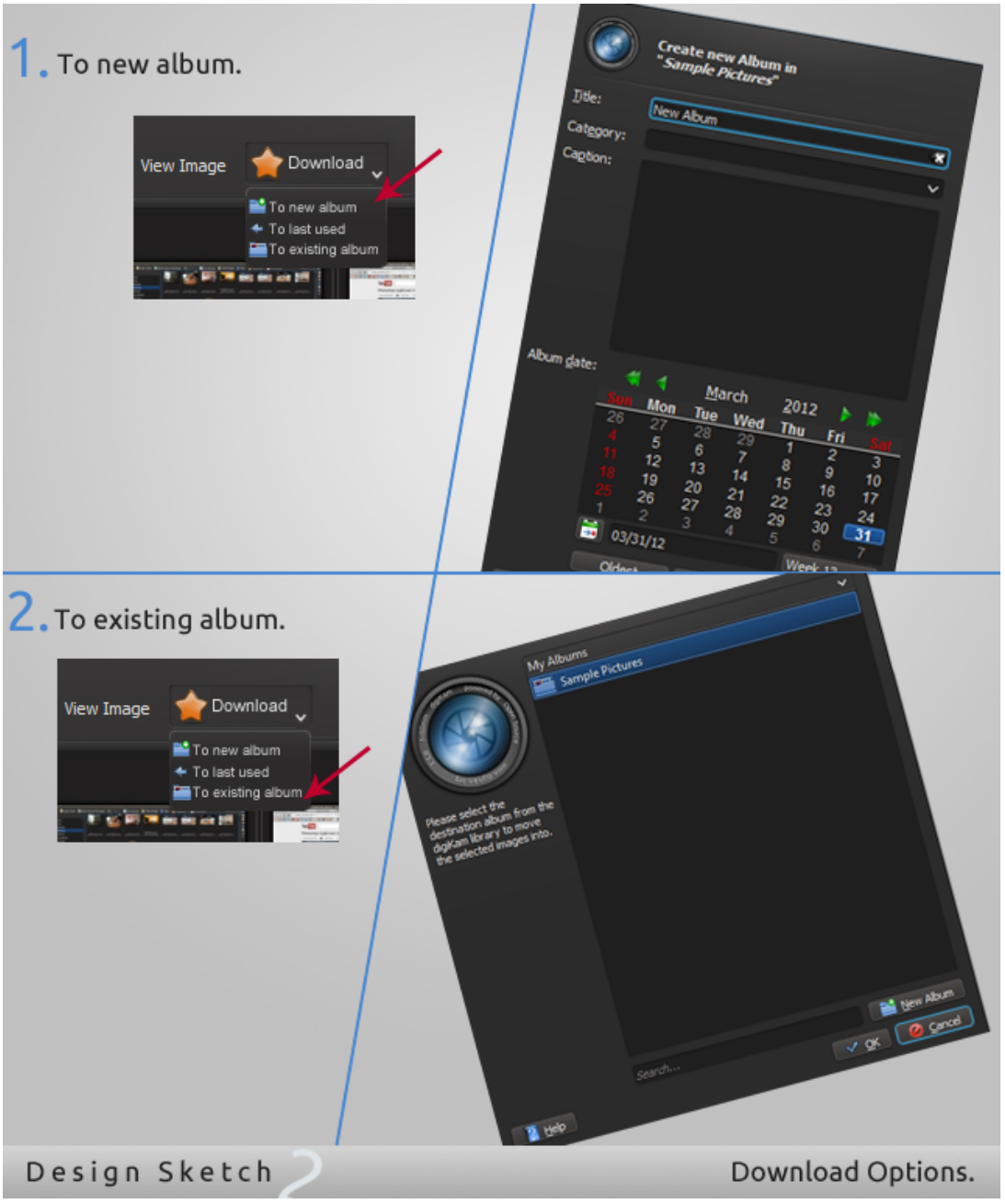


1.1. **Switched to preview mode.**

As mentioned, when clicking on any photo the import interface will be switched to preview mode.

2. **Download options.**

When user clicks on the download button, this means that he is using the default download path he already setup in the camera settings (mockup 7), also there will be 3 available download options. The first is download to a new album which opens a dialog to create a new album. The next option is to download to the last used album which doesn't need any dialog, and the last option is to download to an existing album which you can set it in the opened dialog.



3. **While importing the pause option will be available.**

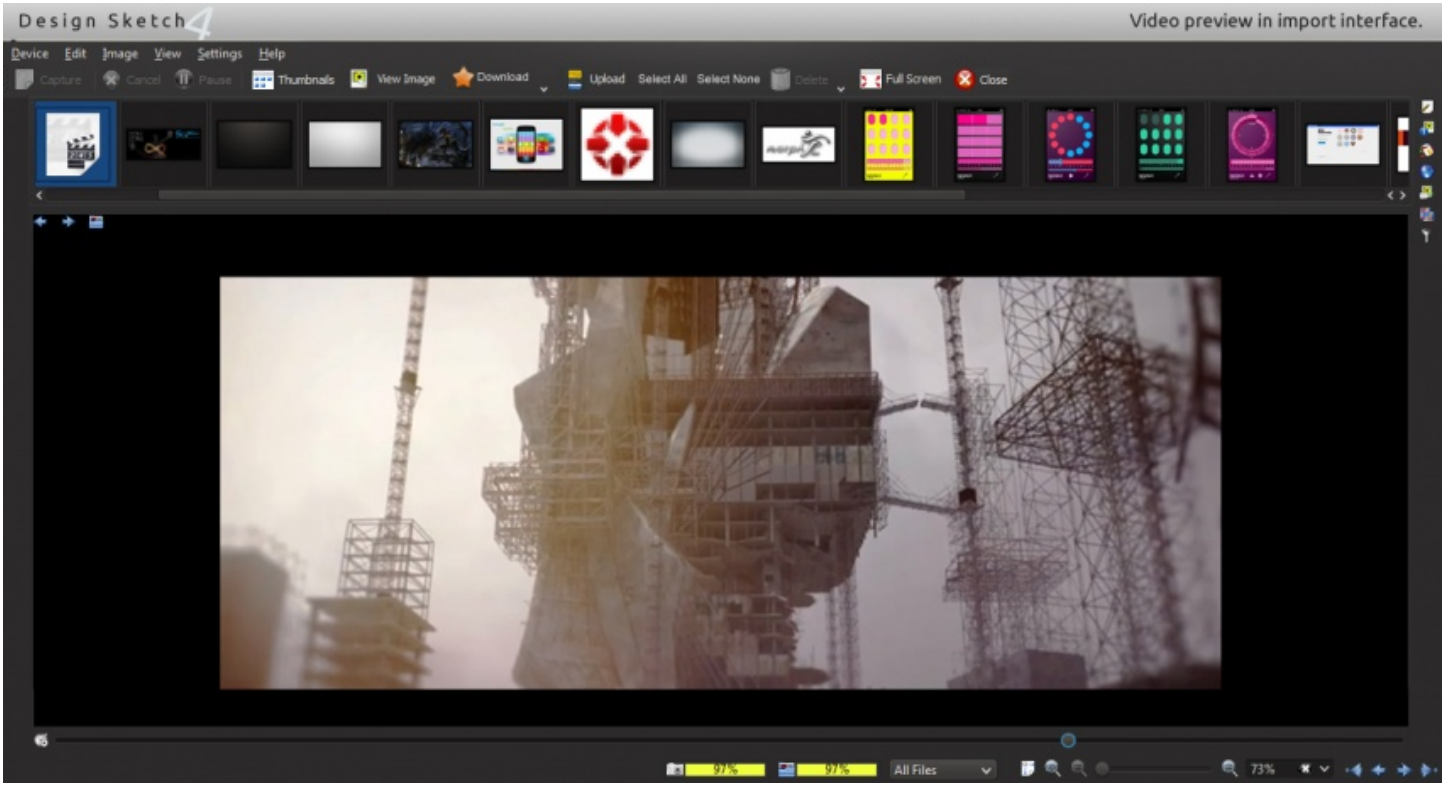
When you download photos from the backend, you can pause importing with the new pause option.

3.1. **Pause option clicked, so resume option will be available.**

When pausing the import process, you can resume it with the new resume option.

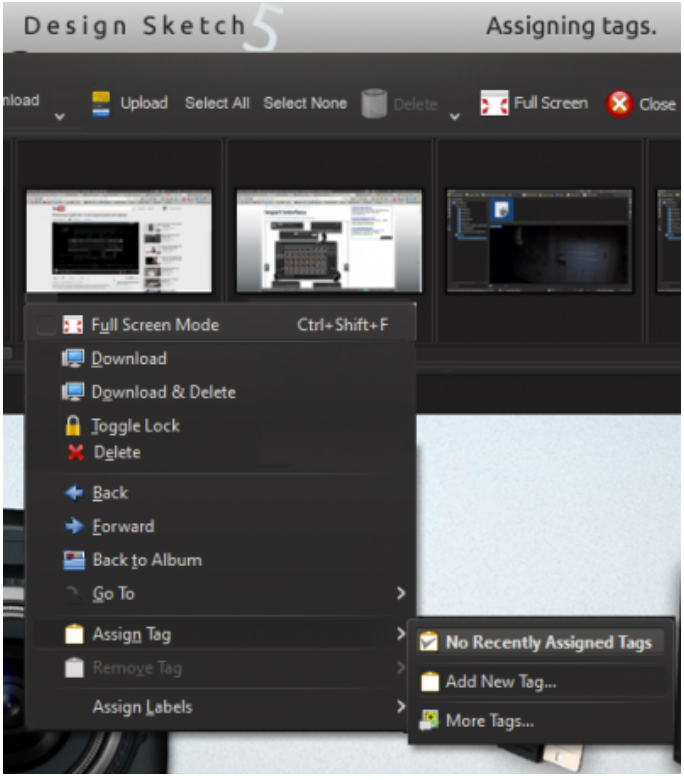
4. **Video preview in import interface which fixes bug 216491.**

The new import interface will be able to preview videos from the backends.



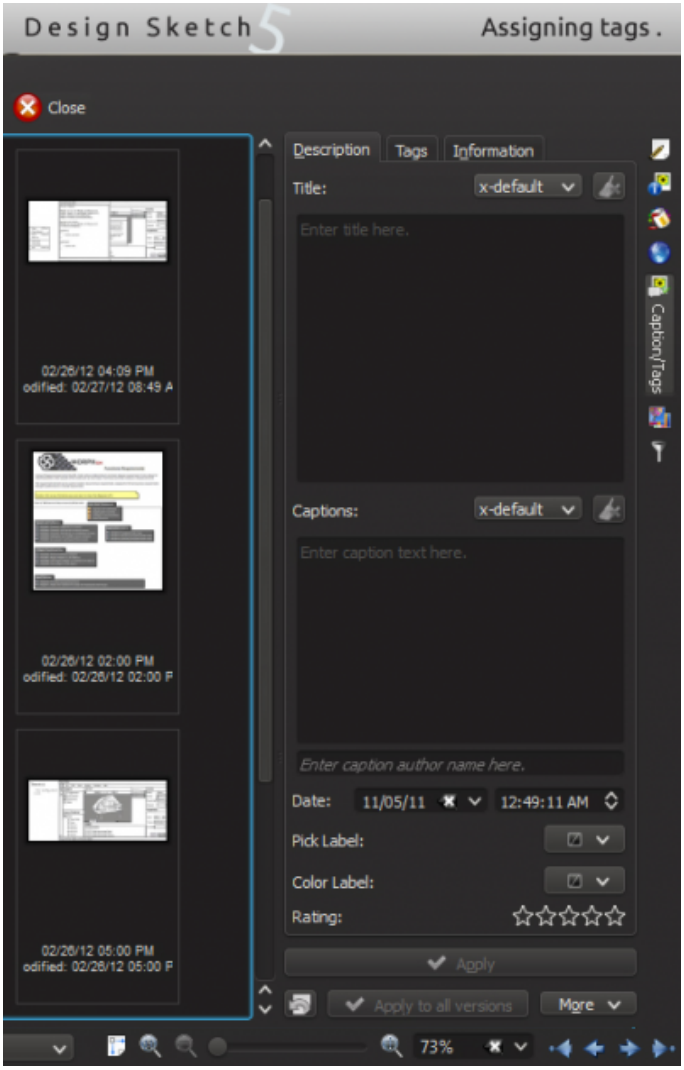
5. **Assigning tags while importing which fixes bug 158437.**

While opening the import interface and viewing photos from the backends, you can add tags to their metadata.



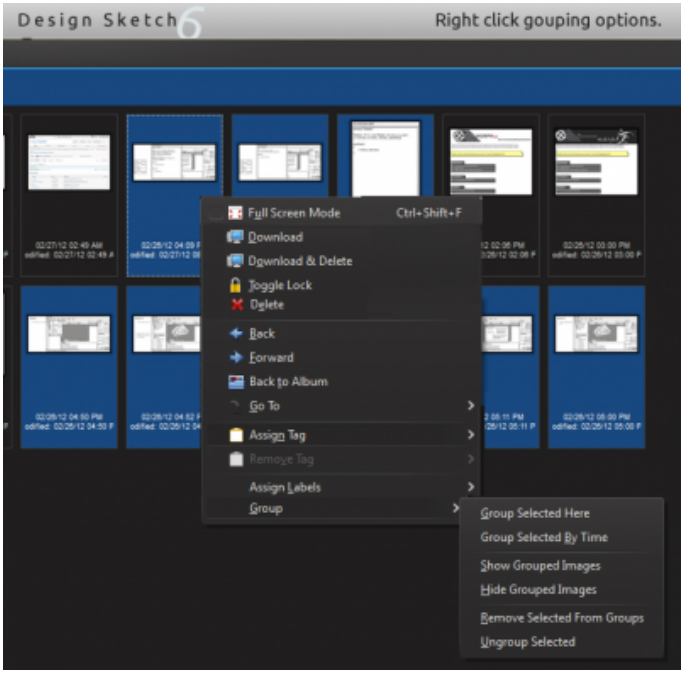
5.1. **Assigning tags from right sidebar.**

There will be a new sidebar for assigning/removing tags as well as to a right click menu as in the next mockup.



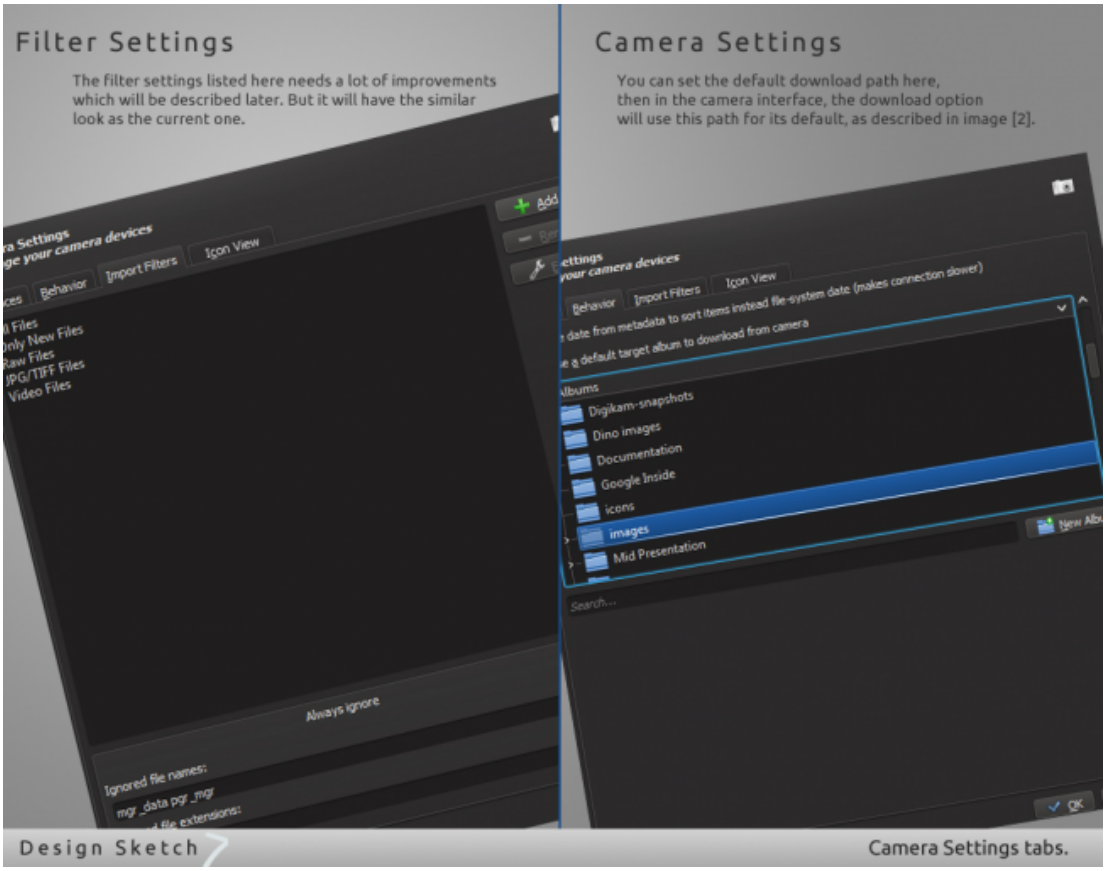
6. Photo grouping which fixes bug 121310.

A new right click menu to group photos with different grouping options.



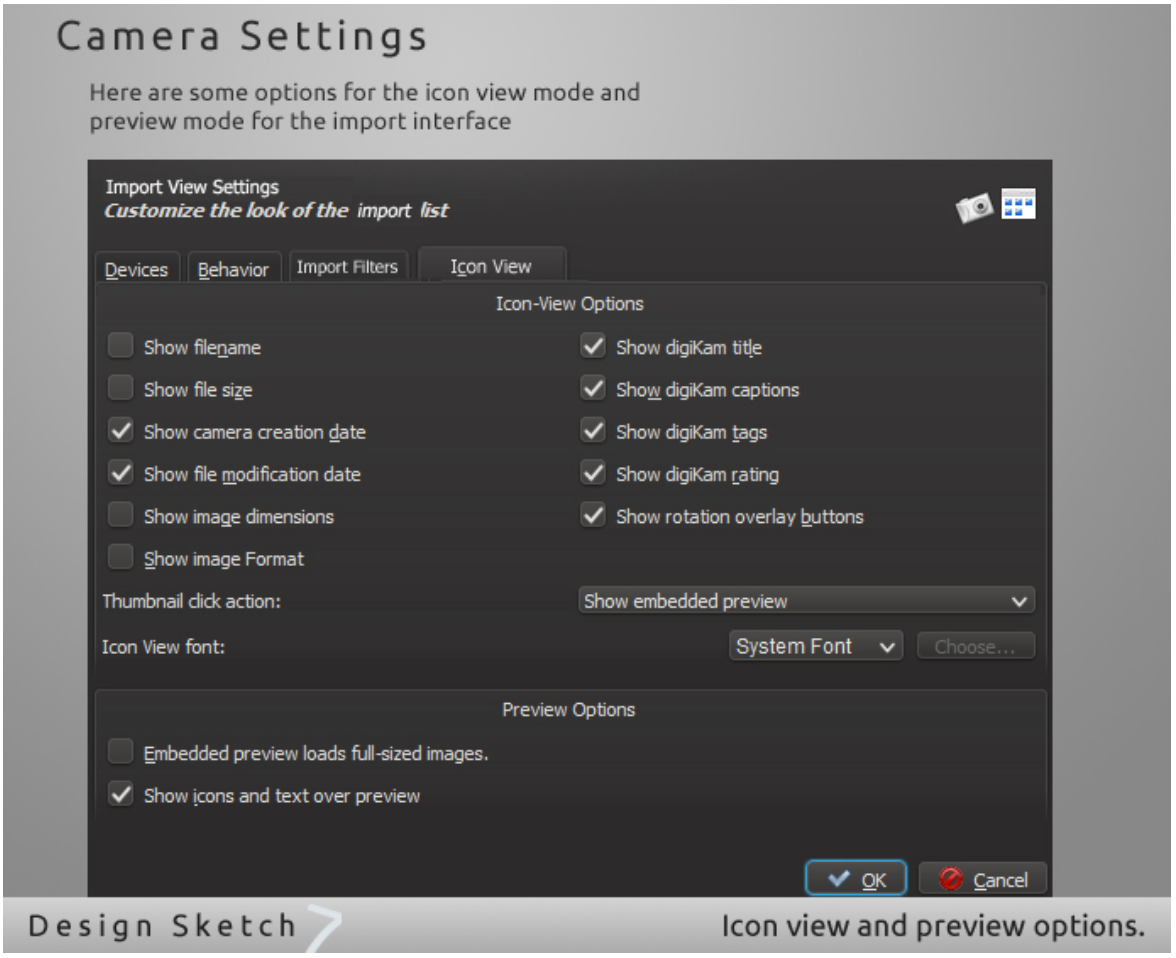
7. **Camera settings panel**

The default download option will download behave the same as the current one, it will download photos to the selected path in the behavior tab in camera settings panel. Also the filter settings will have some improvements.



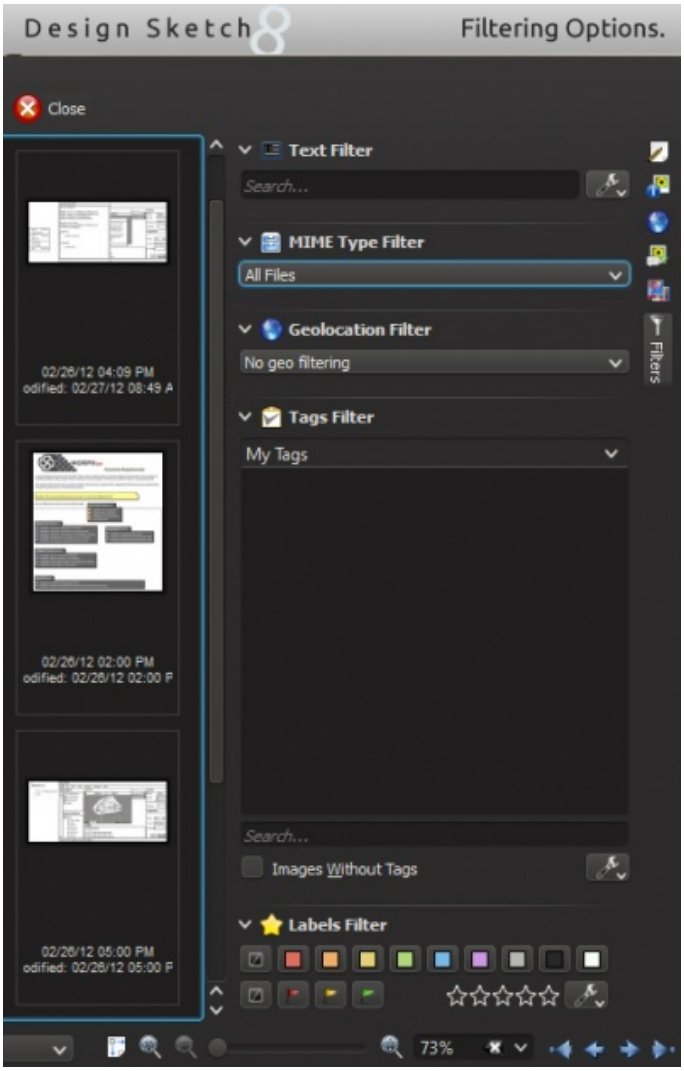
7.1. **Icon view settings for import interface**

The settings of the new icon view and preview mode can be adjusted with the new Icon View settings panel.



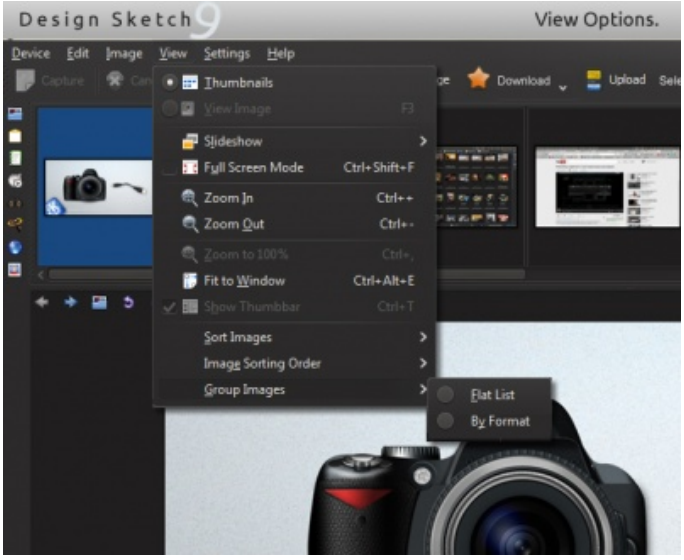
8. **Filters panel**

A new filters panel with some filtering options like filtering by labels and some other options as in the image.

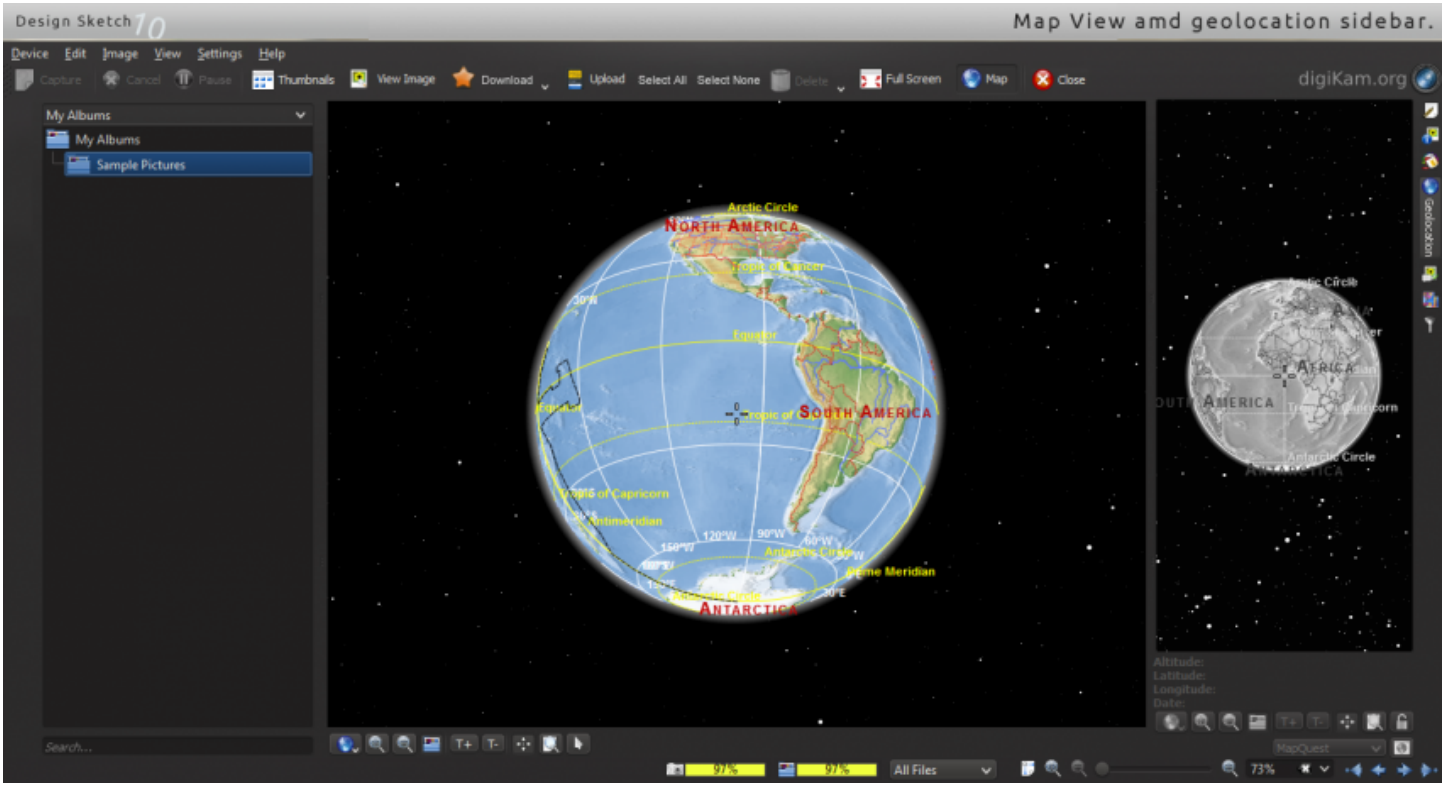


9. **View Options**

There will be some new viewing options like sorting images, grouping the viewed images by a flat list or by format.



10 **10. Map view.** In the revamped interface there will be a new map view, here is a mockup showing it with the Geolaction tab (which is not working in the current implementation).



Implementation

Project repository: [1]

Implementation plan

The first step is to write models in a separate feature branch to list images for this Import interface which appears from its name that it should be generic user interface which means that it must have the ability to import photos from different places and devices, so these models need to be able to list images from two main backends, first one is using the library gPhoto2 which supports around 1400 cameras also in its latest release it added support for some useful extensions like ogg and mp4 which will give us the ability to download them and preview them in the Import interface. Many cameras are not supported by gPhoto2 yet, but have support for the USB mass-storage device class (USB MSC or UMS), which is a protocol that allows USB devices to be accessible, it is well-supported under Linux. So the second backend is UMS.

There are some models for the album gui which are depending on database and specific to albums, we cannot use them, I need to write some new models. Please have a look on the new proposed structure for more details about these new models. Also I need to write test cases for the models.

Next is the views, digiKam already has classes of icon view which are prepared for code reuse, I can reuse them for the Import interface with their delegate classes. That will give us some benefits including a lot of features already implemented in the icon view like tagging/grouping photos, and the preview mode as mentioned in the mockups, also the Import interface code will be cleaned up and the UI will be separate from the model which will handle all the underlying work.

The current implementation has a main view (CameraIconView) which needs to be ported, also there will be a lot of new views to be implemented see a list of them.

The revamped import interface will have tagging, labels and rating facilities, I will add the needed widgets for setting them.

Some UI/UX requirements that are fulfilled in the above mockups:

The current user interface design is powerful, exposing many options. We want to preserve that. But at the same time, there are three very common actions:

a) Download all new files to the last used album

b) Download all new files to a new album

c) Download all new files to an existing album.

It should be possible to carry out task (a) with one click, task (b) and (c) with two or three clicks, without opening a dialog. Friendly to the new user, preserving access to all options for the power user.

Affected Modules

Name of module	Description of changes
cameraiconitem.h	CameraIconItem now is only used by CameraIconView. CameraIconItem is depending on **iconitem** class which will be deprecated and reuse the existing icon-view classes. As a result of that IconGroupItem will also be deprecated.
cameraiconview.h	This module is currently depending on some parent classes (**iconview**) which is based on Q3ScrollView. IconView class will be ported to the existing view (**DCategorizedView**) and CameraIconView will be revamped and adding tagging/lables/rating.
setupcamera.cpp	Camera setup will have a new tab for the icon view and preview modes settings, see mockup no.7.1
devices	This folder includes all the backend code for the current implementation, it will be revamped for the new models.
cameraui	Will be moved to a new name importui and updated with the new interface.
widgets	I will add some new widgets for the import interface, find them in the proposed structure.

Proposed structure for importgui

Name of new model	Description
ImportItemFilterModel	This model is based on QSortFilterProxyModel to be used for sorting items, and filtering them out.
ImportItemModel	This is the main model for listing files from the backends, it might needs some base classes which I will design later.

Name of new view	Description
ImportStackedWidget	Widget stack to contain any of the Image Preview mode, Image Icon View, Media Player view, or even the Map widget mode for the import interface, it will have a similar implementation to the stackedview.h of the album gui.
ImportCategorizedView	CameraIconView will be revamped to based on the DCategorizedView class, this new view is responsible for view the images in the camera/device in a way similar to that used in the album view (PreviewAlbumMode). You can see mockup number 1.
ImportPreviewView	Preview mode for the import interface, it will have a similar implementation to the ImagePreviewView of the album gui. You can see mockup number 1.1.
ImportMediaPlayerView	Preview mode for video and other media files based on Phonon media player to be embded in the stacked widget of the import interface, it will have a similar implementation to the MediaPlayerView of the album gui.
ThumbBarDock	A dock widget to view image thumbnails when the Preview mode is active, it has the same look as a toolbar ThumbBarDock.
ImageThumbnailBar	Preview mode for video and other media files based on Phonon media player to be embded in the stacked widget of the import interface, it will have a similar implementation to the MediaPlayerView of the album gui.

Name of new delegate	Description
ItemViewImageDelegate	The delegate for item view for images.
ImageThumbnailDelegate	The delegate for the Thumbnail bar for images.

Name of added widget to importgui	Description
Captions, rating and tags	There is imagedescedittab.h which can be used in the import interface as in the mockup no 5.1.

Pick label widget	As in the mockup no.8 we need a picklabel widget.
Rating widget	Also we need the rating widget
Color label widget	Also color label.

Relevant Bugs

URL	Name of Bug
121310	Allow to have a group of pictures.
126149	Camera stores both jpeg and raw (nef), handle both as one.
158437	Add ability to create/assign tags to pictures while importing.
216491	Camera interface view does not work to preview movies.
161783	Exif autorotation on import should be backgrounded / pipelined

Tentative Timeline

Date	Work Description
April 24 - May 20	Familiarize with the current model/view code. Discuss with the mentor all the details needed before begining writing code. Establish regular communication with mentor and setup meeting schedules. Setup project blog to monitor progress. Speeding up to code.
21 May – 27 May	Start coding the listed models.
28 May – 3 June	Finishing the models, bug fixing.
4 June – 10 June	Make test cases for the various import models. Start coding the initial ImportStackedWidget.
11 June – 17 June	Start porting the CameraIconView module to model view concept and to be ImportCategorizedView, make use of it in the ImportStackedWidget, refine the ImportStackedWidget.
18 June – 24 June	Deleting old code, make sure that the ported module behaves fine with the models, fixing bugs that will arise.
25 June – 1 July	Coding the ImportPreviewView which is responsible for preview mode in import gui, and add it to the ImportStackedWidget.
2 July – 8 July	Coding the ImportMediaPlayerView and ThumBarDock/ImageThumbnailBar classes, adding them to the stacked widget, fixing bugs, and code review.
9 July -12 July	Removing useless parts, refactoring code, asking community for testing, documentation, getting ready for mid-term evaluations.
July 13	Mid-term evaluations deadline.
16 July – 22 July	Start coding the necessary widgets.
23 July – 29 July	Finishing all the necessary widgets for the import interface, make sure things are working fine.
30 July – 5 August	Finishing all the necessary widgets for the import interface, make sure things are working fine.Fixing any other bugs or troubles that may occur.
6 August – 12 August	Improving the filtering options for the import interface.
13 August – 23 August	Improve documentation, Polishing the code in the feature branch and getting ready for the final evaluation.
August 24	Final evaluation deadline.
After final evaluations	Translation, asking community for feedback, testing, and make sure all things just working fine. I will keep developing for digiKam and will move to some other tasks

---

Do you have other obligations from late May to early August?

Yes, I will have exams which will start in 26/5 and ends in 4/6. During this dates I will work on the project for 15 - 20 hours.

Minimum time involvement estimation:

April 24 – May 20: 35 - 40 hours per week

May 21 – June 4: 15 - 20 hours per week (my exams period)

June 4 – June 25: 35 - 40 hours per week

June 26 – August 15: 45 - 50 hours per week

Information about me

My name is Islam Wazery, I am a 4th grade Computer Science student. I have experience with a wide range of DCC S/Ws especially CG ones like Maya and Blender. Also I worked in low level programming topics like Linux application programming in C. I have a good knowledge of Data Structures and Algorithms. I am feeling good with working in Object Oriented applications in C++, Python, C# or any other OOP language. for the last few months I worked on a self Open Source project which is an SDK for developing 3D Virtual Reality applications using Qt for its visual editor and OGRE for its graphics engine, here is its Github page https://github.com/Zeology/Morph-SDK. I am also an Ubuntu member, I try my best to be involved in Open Source projects to gain more experience. I have a good knowledge in Qt and a bit in GTK+. About source controlling I am using Git and SVN regularly.

I am using DigiKam to organize my pictures. So I'm very familiar with this awesome software.

I have made several patches to Digikam, and I am really interested in participating in its development and to be a part of its great team. I hope I can spend this summer hacking for Digikam, and continue hacking for it after this summer. I have to say that I really feel a great pleasure and a lot of fun when working in the beloved Open Source world, especially in Digikam.

I have a KDE identity with the developer access.

I am planning to spend in average 35 hours every week to work on the project. For the mentor to track my work, I will submit a progress record once a week. Also, we can have meetings on IRC. As long as I can coordinate with the mentor, it really doesn't matter where he/she lives.

Contact information:

IRC nick: Wazery

Email address: Wazery [at] ubuntu [dot] com

Country, City: Egypt, Cairo

IM Service and Username: al.wazery [at] gmail [dot] com

Telephone number: +202 0109-255-9132

Ubuntu Wiki Page: wiki.ubuntu.com/Wazery

TODO

I will use this wiki page to document my progress, so here is a TODO list for the implementation.

Name	Description	Assigned to	Status
Solving bug #161783	Exif autorotation on import should be backgrounded / pipelined	Islam Wazery	In Progress
The last model	Designing and coding the last model for the camera items	Islam Wazery	In Progress
Test the three models	Make some tests for the models	Islam Wazery	In Progress
Delete the CameraThumbsCtrl	The ImportThumbnailModel has replaced it, when using the new model I need to delete the CameraThumbsCtrl	Islam Wazery	Not now!
Implement ImportSettings, like in the mockup number#7.1	The icon-view settings for the Import Tool	Islam Wazery	In Progress
Implement the MediaPlayerView	Like is in the mockup number#4	Islam Wazery	In Progress
Implement the ImportPreviewView	Like is in the mockup number#1.1	Islam Wazery	In Progress
Implement the ImportStackedView	Which will handle (contain) the different types of views	Islam Wazery	In Progress

Completed

Name	Description	Assigned to	Status
Git branch creation	Creating a git branch with the name development/3.0.0.	Islam Wazery	Done
Option to turn on/off high quality thumbnails.	Add a new option in Setup/Camera/Behavior to turn on/off high quality thumbnails.	Islam Wazery	Done
Designing and coding the base ImportImageModel	Detailed design of the base ImportImageModel for the interface and coding it	Islam Wazery	Done
Designing and Coding the thumbnail model	Detailed design of the thumbnail model for the interface and coding it	Islam Wazery	Done
Implement the filter model	Desinging and implementing the filter model, NOTE: filtering options not fully implemented, postponed for now.	Islam Wazery	Done
Design and implement the base views	Design and implement the base views, ImportCategorizedView and its base classes	Islam Wazery	Done
Implement the ImportThumbnailBar	Which is in the mockup number#1.1	Islam Wazery	Done

Note

Please use the talk page to discuss this proposal.

- KDE Links
  - Homepage
  - News
  - Community Forums
  - UserBase Wiki
  - TechBase Wiki
  - Documentation
  - Planet
  - Bugzilla
  - Developer Blogs