Photo Album

When start, use a username to login

If username == admin

different screen:

he can do List users, Create a new user, and Delete an existing user.

for non-admin users:

Once login successfully, show the photo and album information.

Initially, all the albums should be displayed.

For each album, show its name, number of photos, the date of the oldest photo, and the range of dates (earliest to latest).

The user also can do:

Create albums;

Delete albums;

Rename albums;

Open an album, it will display all the photos with their thumbnail images and captions.

once open can do:

Add a photo;

Remove a photo;

Caption/recaption a photo;

Display a photo; (fit the image in its display area, should also show the caption, date-time of capture, and all the tags)

Add a tag to a photo;

Delete a tag from a photo;

Move a photo from one album to another;

Go through photos in album in sequence forward or backward (manual slide-show)

Search for photos (Photos that match the search criteria should be displayed in a similar way to how photos in an album are displayed)

Search by a date range;

Search by tag type-value pairs;

There should be functionality to create an album containing the search results.

The user log out at the end of the session, all the albums and photos should be stored in the disk.

After the user log out, the app is stilling running, allowing another user to log in.

Build an exit button, to quit the app safely.

New window for any errors.

Date of photo:

Use a java.util.Calendar instance for the date and time the photo was taken.

When create a time, set the ms to 0

cal.set(Calendar.MILLISECOND, 0);

Tags:

Photos can be tagged with many attributes you think is useful group by.

e.g. locations, names of people in the photo

tag type & tag value

e.g. “location” & “New Brunswick”

Each pair for a photo is unique.

Model package:

include all data objects (e.g. album, photo), and code to store and retrieve photos for a user.

Must use java.io.Serializable interface and the

java.io.ObjectOutputStream/java.io.ObjectInputStream classes to store and retrieve data.

The photos cannot be stored in the app’s project space.

One user cannot have duplicate album names, but different users can.

GUI Storyboard

Have one or more overview diagrams that show all screens and all transitions between them.

The rest of the storyboards will show the details.

Each screen must be drawn using some drawing package. Or a screenshot for scenebuilder.

Each screen will represent one window of your GUI and will contain all the widgets that go into that screen - text fields, buttons, etc.

Each screen will show transitions to other screens and the events that trigger these transitions.

Every screen needs an appropriate title name.

Work Split Document

Tell what to do in each screen.

OO design - UML representation