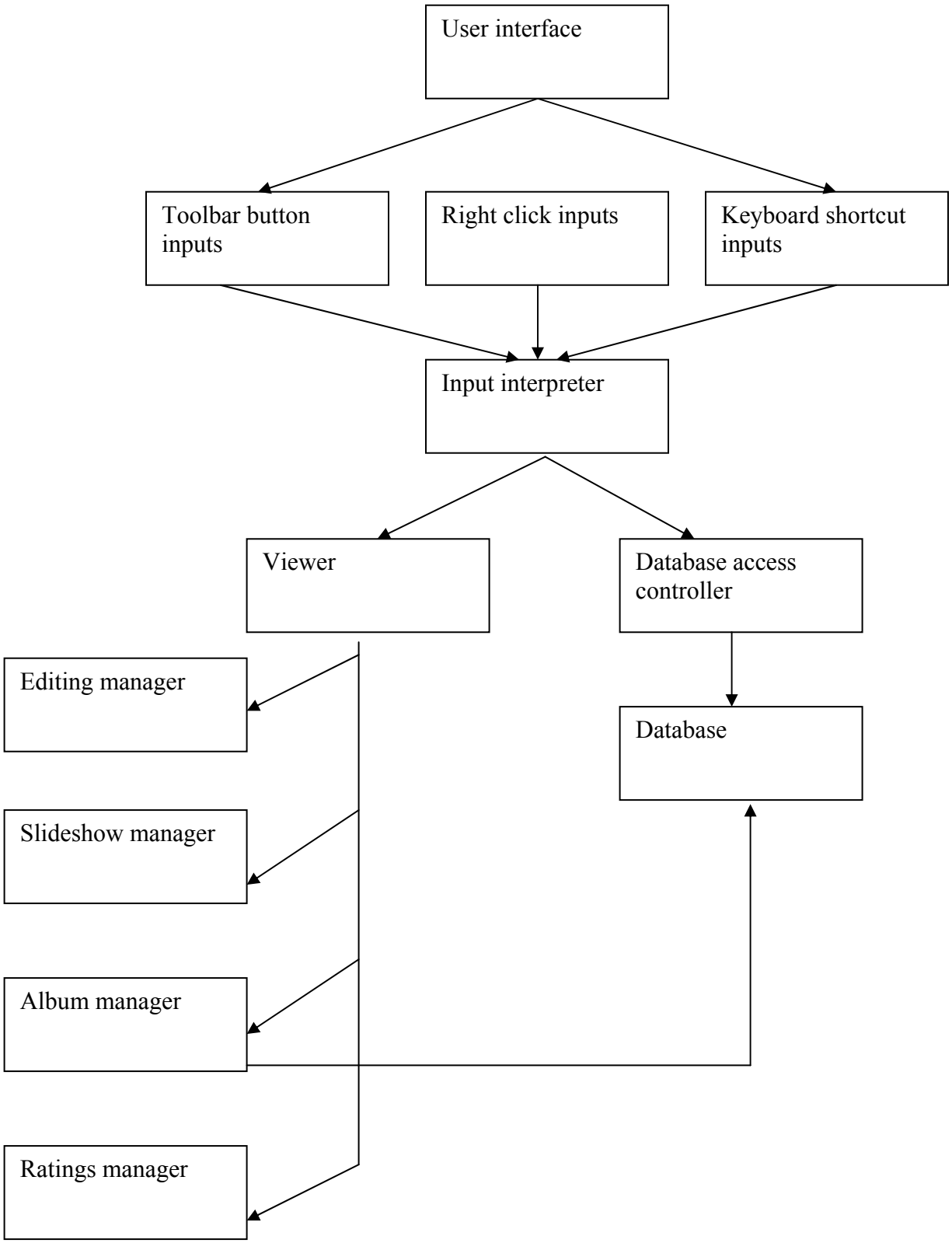
Jason Kim
Jkim21 – 2/21/03

Top Level Design: Photo Wiz

Jason Kim
Jkim21 – 2/21/03

1) Levelized high-level component diagram(s) and description of components



user interface:
This is the GUI where the user sees everything. The pictures are shown and the menus and other interface items appear on it. The user controls the program and all the pictures from here. From here whatever actions the user does will be sent next to the three different types of input components, depending on which action the user does.

toolbar button inputs:
These are inputs that are from clicking on the toolbar buttons or accessing the menu in the interface. From here many of the actions can be performed.

right click inputs:
All the actions that are available on the right click inputs are available as well on the toolbar and menus. The toolbar and menu has more functionality because only the most common actions and functions will be available through right clicking. The way this works is that the user right clicks on a picture and will have a right click menu of functions he can perform.

keyboard shortcut inputs:
If the user does not want to use the toolbar nor the right click option, he has the option of using keyboard shortcuts for functions in the program. Again, not every function available in the toolbar will be able to be hotkeyed. Similar to the right click module, only common functions which are decided to be used often will be hotkeyed.

input interpreter:
This is where all the inputs will go first before moving on to the program to actually do something. This will reduce repeated code. For example, if there was a common function that was available in all three input modes, it would be repeated code to have the same action performed to each input type. But with the input interpreter, from whichever input, the request for the function will be sent to whichever component it needs to go to and make the request.

viewer:
The viewer is what will be used for all viewing of the images. This will control how codecs are processed and how images are displayed within the user interface. The viewer will also control the expandable tree view of albums. This will be the component that then accesses other components for different viewing options and editing.

editing manager:
This is the component that will be accessed if the user wants to change orientation, contrast, brightness, colors, or size. This component will be integrated within the program so that a separate program won't have to open up.

slideshow manager:
If the user wishes to watch a slideshow of the images he has, he will be able to select the slideshow option within the user interface. This will have the viewer access the

slideshow manager which will then go to full screen and display the slideshow of images. There will be options to set random order, delay between pictures, and what kinds of transitions (fade, instant, slide left or right, etc).

album manager:
This is how the user can organize his images. He can create images and then put them into albums that he creates. The albums will be able to have names, dates, and comments so that the albums can carry descriptions. Within the album, the user can set number of pictures per page and have descriptions and other information under each picture.

ratings manager:
This is what the viewer will access if the user wishes to give the images ratings. Using these ratings, the user can then sort the pictures according to the ratings or do searches based on ratings so that there can be order in the madness of so many pictures.

database access controller:
This controller will be called whenever pictures are accessed, added, deleted from the database. Calls to sort will also use this controller.

database:
This is the database that will store all the images along with their information.

2) External dependencies

There are a few external dependencies. One is imagery codecs. Although .bmp, .jpg, .gif, and such are well defined, when new ones come out and if they become standard, we will have to accommodate by editing the viewer so that it can handle them. For the database, if we use another database since creating our own would be a lot of work and probably not as efficient nor powerful as ones already out there that we could use like SQL, then we will have to update whenever they update and when vulnerabilities are found in them, then they will be our vulnerabilities.

3) Task breakdown/group organization

Program manager:
This person will head up the project and be the most responsible for the overview of all the work. He will be the one in charge of motivating others, keeping them on task and on schedule, and have the most authority and overall control. He will be in charge of making sure all the separate teams are talking to each other and that they are on track and getting work done. He will coordinate the meetings. If any major problems should arise, while not expected to solve them, he will be the one the team looks to for suggestions and for leadership in resolving them. As for progress, he will be in charge of making sure that everyone is doing their fair share of work. This is our team leader.

Jason Kim
Jkim21 – 2/21/03

Program Coordinator:
This person is the one who understands this program inside out and knows every detail of the components and how they all go together. This is basically the person who is the glue for putting the parts of the program together. He will be in charge of the design of the program. But this person will also code to help the team because it would be a waste for someone who knew so much of the details of the components and not code any parts of it. They will not have a large coding role though.

User Interface Designer/Implementer:
This person will define the look and layout of the user interface. They will also be in charge of which inputs will be included based on his research and his own thoughts. He will be the one who will have to understand the abilities and limits of whichever toolkit he decides to use to design the GUI.

Coding Team:
There will be one lead coder so that he can be responsible for dividing up the coding so that each person in the team has a proper amount of work. But as the lead coder, he will be responsible for decisions in how the structure of the code will be as well as implementation of it.

Librarian:
This is the person who will be responsible for the things outside coding and designing. Their responsibilities will be:

- Makefile writing
- Internal documentation: responsible for specifications documentation and for its updating at any times
- External documentation: plan and write the user manual

Testing Engineer:
This person will write code to test the program. He will try and "break" the program to see if it is stable. Also he will exhaustively test the program to make sure that all the planned functionality is working properly and is there.

4) Schedule

This is a seven week schedule.

Week 1:
- Finalize specs
- Assign jobs
- Finalize top level design

Week 2:
- Start design of the user interface
- Write the makefile and decide which libraries are to be used
- Complete component layout and design of each

Week 3:
- Complete design of the user interface so that it will be clear what is going to go into it and how it will look.
- Establish an approach to coding this program.
- Pseudo code
- Plan integration strategy

Week 4:
- Figure out which user interface functions are not feasible, by talking to the coding team.
- User interface is started by coding team
- Program manager should check schedule to check they are running on time and have significant progress

Week 5:
- Should be halfway done with components
- User interface should be nearly complete
- Integrate components. Even if some are not done or functioning, basics should work and integration should be started so that it is a less painful integration in the end.
- Start some testing by the testing engineer(s)
- Specifications and top level design should be final and need no more changes by librarian.

Week 6:
- Finish coding
- Finish user interface
- Finish user manual
- Finish integration
- Start thorough testing of entire program

Week 7:

- Find and fix bugs
- Freeze code
- Prepare to present

6) Assumptions

Some assumptions made were the toolbar inputs. I am not sure that the original specifications had it where all the functions would be in the toolbar. I am unclear in how the overlap in functions would be with inputs, if at all. Some of the organization of the images using the database and the albums were unclear. I am assuming that the albums will be stored in the database. So there, I assumed that the album will access the database. I also assumed the added descriptions in the albums. It was not clearly defined so I figured it would be like a regular picture album where they could be captions and other information under the albums.