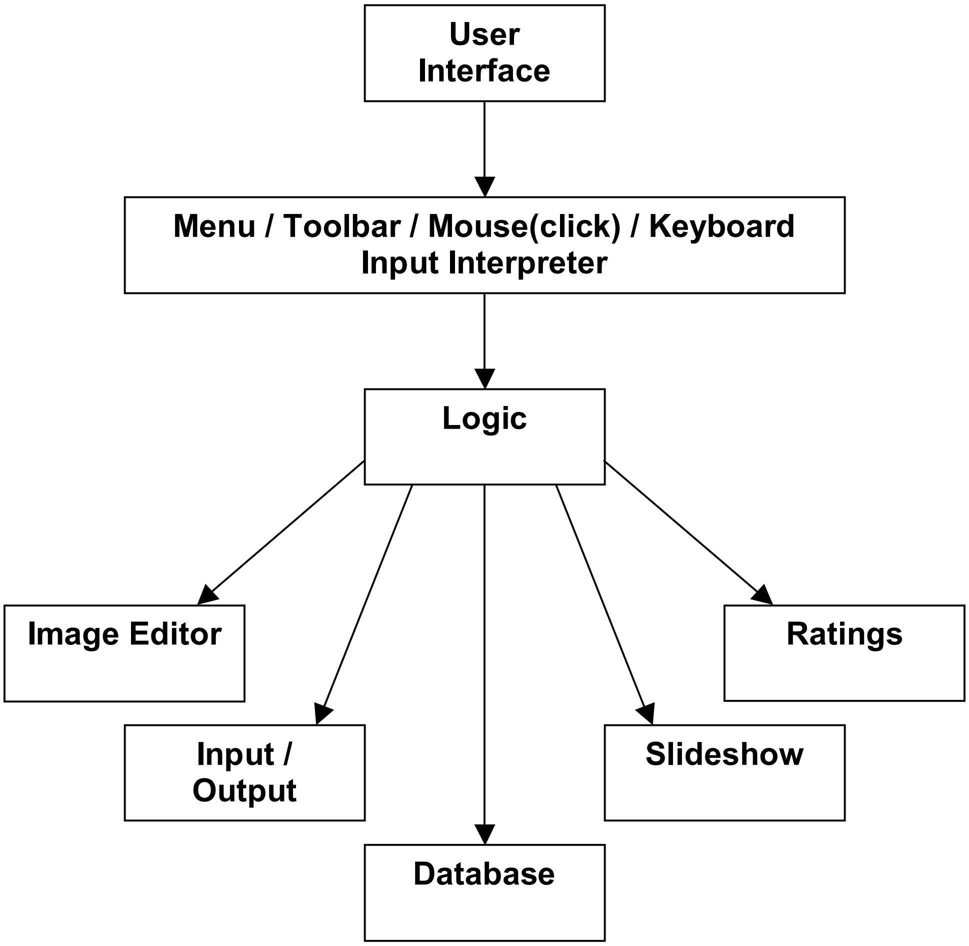
Top-level Design Document
Photo Wiz
Thomas Yoon (tyoon)

1. Levelized high-level component diagram



- **User Interface**
  This is the graphical user interface. It is the gateway between what the user wishes to do with the program. It also shows the results of the executed commands by the logic.

- **Menu / Toolbar / Mouse(click) / Keyboard Input Interpreter**
  This takes the input of what action the user wishes the program to perform. Different methods of interaction are the menu commands, easy-access toolbar buttons, left- and right-mouse clicking, and keyboard shortcuts.

- **Logic**
  The logic is essentially the brain of the program. It receives the information from the Input Interpreter and prepares it to be sent to other divisions.

- **Image Editor**
  The image editor implements the most common changes a user might want to use to change the appearance of a certain image. These features would include the ability to change orientation, contrast, brightness, colors, size, resolution, and quality. These changes would be shown in the GUI in the editor window.

- **Input/Output**
  The Input/Output's purpose is to manage the types of files that are worked upon. The user may choose to save an edited image as a different file type as the original.

- **Database**
  The database's purpose is to control the placement of albums as well as the images within them. It has the ability to create, delete, and rename directories. It also contains information on each image such that they may be sorted in a certain way, such as by name, size, and date.

- **Slideshow**
  The slideshow allows the user to view the contents of a specific folder, viewing each image only on the screen for a certain period of time before switching to the next image. It continues to do so until all images have been viewed.

- **Ratings**
  The user may rate the quality or preference of any image. The program can use this information to display which images rank highest or lowest.

2. External Dependencies

There are some possible external dependencies when dealing with the database as well as file types. We could write something on our own to take care of the information storage and management, but it may be easier and more efficient to use one as an external dependency. Also, we may need external dependencies pertaining to the Input/Output module when reading or writing as different file types.

3. Task Breakdown

- **Program Manager**
  This person is the head coordinator of the project. He sets the goals and deadlines for specific parts of development. He also stays in constant contact with each group and arranges meetings.

- **Programmers**
  This group does the actual coding of the program. If a module consists of more than one person working on it, one person will act as a sub-program manager for that module and has more influence on decisions. The different divisions are described above in the levelized high-level component diagram.

- **Tester**
  This person tests the program for errors. He does so by writing drivers and tries to make the program crash in any way possible.

- **Editor**
  This person generates documentation for maximum clarity for both external and internal descriptions. It helps during the development as well as after release.

- **Researcher**
  This person researches the availability and feasibility of implementing external dependencies that may be used in the program. Most work here would primarily be done before any relevant coding is started.

4. Group Organization

Program Manager			**1 person**
Programmers			**7 people**
	User Interface	1 person	
	Input Interpreter	1 person	
	Logic	1 person	
	Image Editor	2 people	
	Database	1 person	
	Input/Output	1 person	
	Slideshow		
	Ratings		
Tester			**1 person**
Editor			**1 person**
Researcher			
	This role can be played by the programmer of that specific part.		

5. Schedule

3-07	Final Design Document handed in.
3-12	Interface Proposals handed in.
3-15	Interface Comments handed in.
3-17	Meeting for discussion on interfaces.
3-21	Final Interfaces handed in.
3-31	Meeting for discussion on detailed designs.
4-04	Detailed designs handed in and begin coding.
4-14	Meeting for discussion of progress in coding and beginning of integration.
4-21	Finalize integration and begin testing.
4-28	In-class demo day.
5-09	Public demo day.

6. Assumptions

There were no assumptions since I originally did the specifications document.