

# ENGG\*4450 : Large Scale Architecture

## Software Requirements Specification

Adam Erb  
Gregor Scott



### CONTENTS

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Scope . . . . .	2
1.3	Glossary . . . . .	2
1.3.1	Definitions . . . . .	2
1.3.2	Acronyms . . . . .	2
1.3.3	Language . . . . .	2
1.4	References . . . . .	2
1.5	Overview & SRS Structure . . . . .	2
<b>2</b>	<b>Overall description</b>	<b>3</b>
2.1	Product Functions . . . . .	3
2.2	Product Perspective . . . . .	3
2.2.1	User Interfaces . . . . .	4
2.2.2	Software Interfaces . . . . .	5
2.3	User characteristics . . . . .	5
2.4	Constraints . . . . .	5
<b>3</b>	<b>Specific Requirements</b>	<b>6</b>
3.1	Digital Library Functional Requirements . . . . .	6
3.1.1	Creating Libraries . . . . .	6
3.1.2	Adding & Removing Images . . . . .	6
3.1.3	Tagging Images . . . . .	6
3.1.4	Arranging Images . . . . .	6
3.1.5	Viewing Images . . . . .	6
3.1.6	Changing Formats . . . . .	6
3.2	Image Processing Requirements . . . . .	6
3.2.1	Crop Tool . . . . .	6
3.2.2	Scale Tool . . . . .	7
3.2.3	Contrast Adjustment . . . . .	7
3.3	Data Storage Requirements . . . . .	7
3.4	Error Checking Requirements . . . . .	7
3.5	Standards Compliance Requirements . . . . .	7
<b>4</b>	<b>Use Cases</b>	<b>8</b>
<b>5</b>	<b>Traceability</b>	<b>11</b>

# 1 INTRODUCTION

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to outline the proposed deliverables for the engg\*4450 semester project. This document will outline:

- 1) Project Scope
- 2) Project Features
- 3) Use Cases
- 4) Final Deliverables

## 1.2 Scope

The scope of this project will be to produce a Digital Image Library (DIL) for personal use, which will allow for the viewing, and organization, of digital images. Additional features of the software will be outlined in section 2.1 and state the minimum functionality that can be expected in the final submission; any additional functionality is not guaranteed. In addition to the software system, deliverables will also include a user manual, project source code, and any project documentation outlining construction of the system.

## 1.3 Glossary

### 1.3.1 Definitions

**Swing** Java graphical interface library.  
**Tag** a digital marker used to logically organize digital photos.  
**Window** a single graphical workspace contained in the Windows operating system.

### 1.3.2 Acronyms

**SRS** Software Requirements Specification  
**DIL** Digital Image Library  
**JAI** Java Advanced Imaging  
**GUI** Graphical User Interface

### 1.3.3 Language

In this document **system** & **product** can both be assumed referring to the final software to be submitted for the purposes of this project. When referring to sections of the Graphical User Interface (GUI), **window** refers to a window in the operating system, while **view** refers to a subsection of the "main" window (see: section 2.2.1). **Image** and **picture** are used interchangeably.

## 1.4 References

- [1] "Ieee recommended practice for software requirements specifications," *IEEE Std 830-1998*, p. i, 1998.

## 1.5 Overview & SRS Structure

The system detailed in this SRS is organized under the following heading: "Overall Description", "Specific Requirements" and "Use Cases". Overall Description gives a general, low level of detail, outline of the expectations, and background of the system. Requirements gives a highly detailed description of each element of the system. Use Cases outlines an example of a generic user interacting with the system.

## 2 OVERALL DESCRIPTION

### 2.1 Product Functions

The proposed product of this project will be a software system tailored to the creation, and viewing, of a DIL for personal use. The central functionality of this product will revolve around the importation, storage, and viewing of digital images, but will include several additional features to be outlined. The system will organize these digital images in libraries via digital “tags”, which will be assigned by the user with intention of logically organizing images by theme. Using these assigned tags, the user will then be able to selectively view, and reorganize, images in a given library. The product will also allow for digital images to be stored in different digital formats. Additional functionality will include three image processing tools, which will be available to apply on all image formats. These tools will include:

- 1) Cropping
- 2) Resizing
- 3) Contrast Adjustment

The product functionality will be accessible via simple graphical user interface. Some of the functionality that will *not* be implemented in this product will be:

- 1) Online storage
- 2) Support for third party extensions
- 3) Support for additional languages

### 2.2 Product Perspective

The system will be an application running on a Java Virtual Machine, and designed to run in the Windows XP operating system environment. It will rely on several pre-existing open-source libraries, which will be included within the software package. There will be *no* support for third party extension, and only system interfaces will be those required by the virtual machine, operating system, and supporting libraries. Refer to figure 1 to see the context the system will be running in.

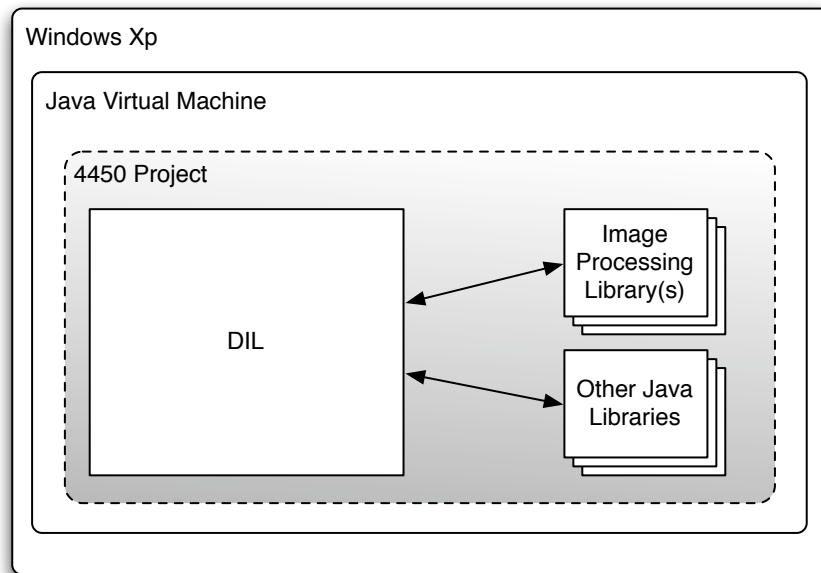


Fig. 1: Overview of running environment & hierarchy for proposed system.

### 2.2.1 User Interfaces

The user will interact with the software through: monitor, keyboard, & mouse; Input will be received via mouse and keyboard, and feedback provided through the monitor GUI. The GUI will be composed of several windows, with the “main” window containing the commonly used features in the system. All elements in the GUI will be defined under the Java Swing Library. Windows in the GUI are defined as follows:

#### Main Window

Always open while the software is running. Features the main display elements of the GUI, is split up into different “Views”, and supports drop-down menus. The different views in the main window are:

##### Tag View

The Tag View shows all tags presently defined in the system, and allows the user to click on one, or multiple tags, which will dictate the image’s names displayed in the File View.

##### File View

The File View will show image names based on the selected tags, or all image names, if no tag is selected.

##### Image View

Selecting an image name in the File View will prompt the image to be displayed in the Image Display View.

The user options defined by the drop down menu interface, will be configured in the following way:

- 1) File
  - a) New Library
  - b) Open Library
  - c) Save As...
  - d) Import Photos
    - i) Files
    - ii) Folders
- 2) Tag
  - a) Add
  - b) Delete
- 3) Filters
  - a) Crop
  - b) Scale
  - c) Contrast Adjustment
- 4) Help
  - a) Documentation

An overview of the main window can be seen below in figure 2.

#### File Navigation Window

Pre-defined by the Swing library this window will be open when importing or creating new files. Allows for navigation through the file hierarchy.

#### Filter Windows

Additional windows will be defined by each image filter in the system. These windows will be customized for each filter and allow for the user to input the desired filter parameters.

Additionally, keyboard configurations will reduce repetitive point and click motions. The following keyboard inputs will be supported:

- |                |   |
|----------------|---|
| <b>Enter</b>   | will allow user to confirm image filter parameters. |
| <b>Control</b> | will allow for the selection of multiple images.    |

**Qwerty** keys will be allowed for defining filenames in open & save operations.

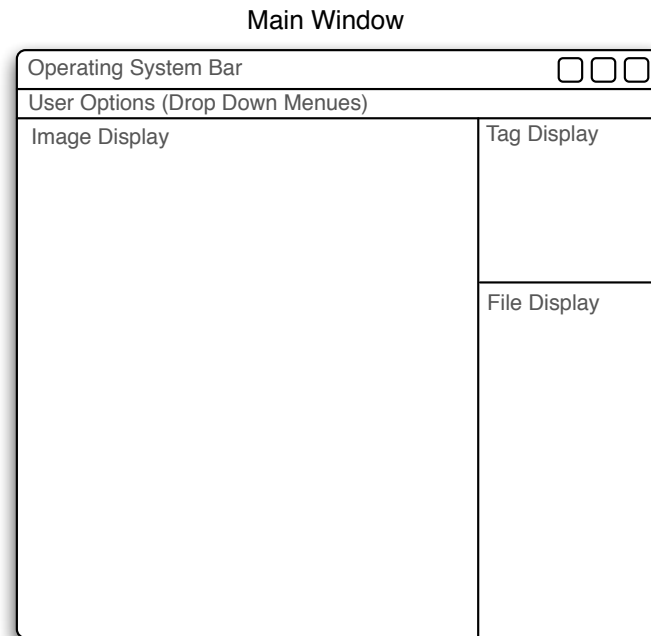


Fig. 2: “Main” window configuration for system GUI.

### 2.2.2 Software Interfaces

The system *will* have several dependancies to operate correctly. These will include Java image processing libraries, the Java Environment, and the Operating System. Software dependancies will be as follows:

- 1) Java(TM) Runtime Environment, Standard Edition version 1.5.0\_20
- 2) Windows XP Service Pack 2 or higher
- 3) Java Advanced Imaging version 1.1.3
- 4) ImageJ Library version 1.43

### 2.3 User characteristics

The target user for this software has the assumed characteristics:

- 1) An english speaking adult (limited language support)
- 2) Any education level
- 3) Intending personal use

### 2.4 Constraints

The following constraints must be met in the final submission:

- 1) The project will have no budget.
- 2) Any software dependancies must be open-source &/or public domain.
- 3) Project must be complete by November 30<sup>th</sup>,2010.
- 4) Software must be able to support a minimum of 20 photos.
- 5) Software must be able to support digital image formats outlines in section 3.1.6.

### **3 SPECIFIC REQUIREMENTS**

#### **3.1 Digital Library Functional Requirements**

The central functionality of this system is as a DIL, and supports the following functionality, which will be outlined below:

- 1) Creating and managing individual libraries.
- 2) Adding & Removing images from libraries.
- 3) Defining, Adding, & Removing tags to images.
- 4) Viewing & Sorting Images based on tag values
- 5) Restoring in different formats.

##### *3.1.1 Creating Libraries*

The system will allow the creation, and manipulation, of DILs. Libraries will be able to keep track any number of digital images, but will only be required to support up to 20. Only one library can be loaded in the program at a time.

##### *3.1.2 Adding & Removing Images*

The system will allow images to be added to or removed from libraries. These operation will never effect the original image or its file location.

##### *3.1.3 Tagging Images*

All organization in the DIL will defined by digital tags associated with each image file. The user will be able to define new tags, add & remove tags from either single, or groups, of digital images, and organize digital images in the library based on these tags. All tags in the library will be displayed in the "Tag Display" section of the main window.

##### *3.1.4 Arranging Images*

By selecting one or more of the tags in the Tag Display, the File Display will show the images with those corresponding tags. If no tag is selected in the Tag Display the File Display will default to displaying all images in the Library.

##### *3.1.5 Viewing Images*

Viewable images will be displayed in the main window File Display. The user will then be able to click on images in the File Display, and they will be previewed in the Image Display of the main window. If multiple images have been selected, the most recently selected image of the group will be displayed in the Image Display.

##### *3.1.6 Changing Formats*

The system will also support changing image formats. The user will be able to select single files and re-save them under new digital image file formats. The following formats will be supported by the system (see: section 3.5).

#### **3.2 Image Processing Requirements**

In addition to the simple functionality in 3.1, the system will also support three simple image processing tools:

- Crop
- Scale
- Contrast Adjustment

##### *3.2.1 Crop Tool*

The crop tool will allow single images to be cropped at a time. The user will be able to select an image from the File Viewer, and define the parameters of the crop. The crop transformation will then be applied to the image.

### 3.2.2 Scale Tool

The scale tool will allow the user to select and rescale a single image in the library. The user will be able to select an image from the File View, and then apply the scale transformation, based on new height, width, or scale parameters supplied by the user.

### 3.2.3 Contrast Adjustment

The contrast adjustment tool will allow the user to select a single image, and change the brightness, and contrast parameters of the image. The user will be able to select the image in the File View, and apply the adjustment.

## 3.3 Data Storage Requirements

Information about the Library will be stored in a binary file, at a location specified by the user, with the extension format “.diglib”. The .diglib file will contain tag information, and references to the image files contained in the library, but will *not* contain image data itself. The .diglib file will point to its images by means of relative addressing, and will rely on the user to notify it of any changes to image file location if they occur (see: section 3.4).

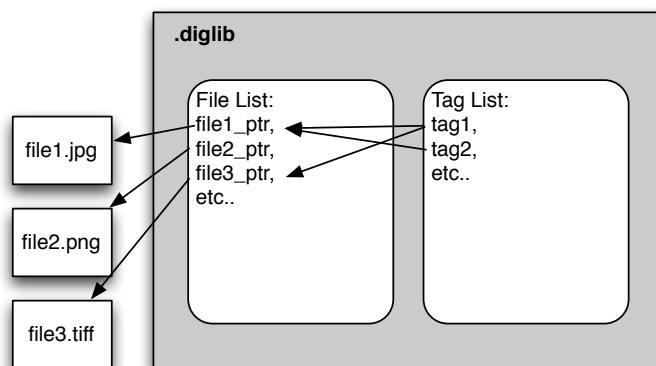


Fig. 3: Library storage format.

## 3.4 Error Checking Requirements

The system will implement basic error checking to defend against bad user input & corrupt files. Checks for bad user input will encompass:

- Checking for incorrect input types (eg: non-numerical input, when numerical input is required).
- Checking for out of range inputs, or unrealistic inputs, on image processing operations.
- Checking for bad tag definitions (Using non-alphanumeric tags).
- Checking for bad, corrupt, or unsupported digital image formats (see: section 3.5).

Additional corrupt file checks for corruption will be performed by the system when opening up a library file.

## 3.5 Standards Compliance Requirements

The system will be compliant with the following digital image formats:

- .jpg
- .bmp
- .png
- .tiff

## 4 USE CASES

- UC1:** User opens a library.  
**Precondition:** The program has started, and a library file exists.  
**Flow of Events:**
- 1) *User* selects File→Open-Library from drop-down menu in main window.
  - 2) *System* displays a file navigation window.
  - 3) *User* navigates and locates existing library file, clicks open.
  - 4) *System* closes the file navigation window, and loads library data.
  - 5) *System* displays library information in main window.
- Alternative Flow 1:** At step 2 library file can not be located.  
*User* clicks cancel.  
*System* closes file navigation window, returns to main window.  
**Postcondition:** Library has been loaded and image names are displayed in File Display.  
**Related Use Cases:** *none*
- UC2:** User creates new library.  
**Precondition:** Program has been started.  
**Flow of Events:**
- 1) *User* selects File→New Library from drop-down menu in main window.
  - 2) *System* displays a file navigation window.
  - 3) *User* selects location, and defines name for new library.
  - 4) *System* builds new library file in the specified folder.
- Alternative Flow 1:** At step 3 *User* choses to cancel.  
*System* closes file selection window and returns to main window.  
**Postconditions:** *System* creates library file.  
**Related Use Cases:** *none*
- UC3:** User imports photos from file to library.  
**Precondition:** program has started, and library is open.  
**Flow of Events:**
- 1) *User* selects File→Import Photos →Files option from drop-down menu.
  - 2) *System* displays a file navigation window.
  - 3) *User* selects one, or multiple files, in file navigation window, and clicks import.
  - 4) *System* closes file navigation window, imports photos to library, and updates/displays main window.
- Postcondition:** Photos are imported to library, and displayed main window.  
**Alternative Flow 1:** At step 3 *User* choses cancel instead.  
*System* closes file navigation window and returns to main window.  
**Related Use Cases:** *none*
- UC4:** User imports all photos from a folder to the library.  
**Precondition:** A library is currently loaded.  
**Flow of Events:**
- 1) *User* selects File→Import Photos→Folder option from drop-down menu.
  - 2) *System* displays a file navigation window.
  - 3) *User* selects a folder, clicks “import”.
  - 4) *System* closes file navigation window, and imports all photos in folder, and sub-folders, into library.
- Alternative Flow 1:** At step 3 *User* choses cancel.  
*System* closes file navigation box, returns to main window.  
**Postconditions:** Program updates the library with the new photos and displays them

in the main window.

**Related Use Cases:** *none*

**UC5:**

User selects one or more photos for editing or tagging.

**Precondition:** A library with one or more photo is loaded.

**Flow of Events:**

- 1) *User* selects one or more photos by:
  - a) single clicking on single photo.
  - b) holding keyboard "Control" key and single clicking on multiple photos.
- 2) *System* highlights photo names in File Display to indicate selection.

**Alternative Flow 1:** At step 2 *User* single clicks again to deselect photo(s) in the File Display.

*System* un-highlights photo names to indicate they are no longer selected.

**Postcondition:**

- Files are highlighted in File Display to indicate selection.
- If single file is selected, it is displayed in Image Display.
- If multiple files are selected, last selected image is displayed in Image Display.

**UC6:**

User performs image processing operation on a selected photo.

**Precondition:** A single photo has been selected and is displayed in the Image Display (ref: UC5).

**Flow of Events:**

- 1) *User* chooses image processing operation by:
  - a) *User* navigates to Filters→Crop.
    - i) *User* clicks on image in two places to mark the top left, and bottom right corners of the crop.
    - ii) *System* displays borders of crop, overlaid on image.
    - iii) *User* presses "Enter" key on keyboard to apply crop.
  - b) *User* navigates to Filters→Scale.
    - i) *System* prompts for a ratio to scale to in a new window.
    - ii) *User* enters ratio in new window, clicks "OK".
    - iii) *System* closes new window.
  - c) *User* navigates to Filters→Contrast Adjustment.
    - i) *System* prompts in new window with slider bar to adjust the contrast.
    - ii) *User* selects new contrast level, clicks "OK".
    - iii) *System* closes new window.
- 2) *System* applies UC7
- 3) *System* applies operation to selected image.

**Alternative Flow 1:** *User* cancels flow by clicking "cancel" at any of the *System* prompts. *System* returns to photo display with out applying any changes. **Postcondition:** Operation applied to image.

**Related Use Cases:** See UC5 for photo selection, See UC7 for file saving

**UC7:**

User Saves Changes

**Precondition:** A single photo has been edited

**Flow of Events:**

- 1) *System* prompt to save proceeding another use case (ref: UC6) or *User* chooses File→Save As...
- 2) *System* displays a file navigation window.
- 3) *User* selects location, format, and enters a file name to save the edited photo under, or chooses current location and format, and clicks Save.

4) *System* preforms the action and updates library file.

**Alternative Flow:** At step 2 *User* clicks “cancel” in file navigation window.  
*System* returns to photo display as before flow started.

**Postcondition:**

- File is saved and library is updated.

**Related Use Cases:** See UC5 for save prompts

**UC8:** User tags one or more photos in the library.

**Precondition:** One or more photos have been selected. (ref: UC5)

**Flow of Events:**

- 1) *User* selects Tag→Add option from drop-down menu.
- 2) *System* prompts *User* with text input box requesting a tag.
- 3) *User* enters and existing tag or a new tag.
- 4) *System* determines if tag exists or if a new tag has been entered.
- 5) *System* displays details of the actions about to be preformed to *User* in a dialog bog with a “Cancel” and “Apply” options.
- 6) *User* selects “Apply”
- 7) *System* applies all changes to the library and Main Window.

**Alternative Flow:** At step 6 *User* clicks “cancel” in prompt.

*System* returns to photo display as before flow started.

**Postcondition:** Tag Display includes the new tag, library updates.

**UC9:** User deletes a tag from the library, removing it from photos.

**Precondition:** Library with photos and tags is open.

**Flow of Events:**

- 1) *User* selects Tag→Delete option from drop-down menu.
- 2) *System* prompts *User* with text input box requesting a tag name.
- 3) *User* enters an existing tag.
- 4) *System* determines if the tag exists and which photos have the tag.
- 5) *System* displays details of the actions about to be preformed to *User* in a dialog bog with a “Cancel” and “Apply” options.
- 6) *User* selects “Apply”
- 7) *System* applies all changes to the library and Main Window.

**Alternative Flow 1:** At step 4 *System* does not find the tag.

*System* system displays information box telling *User* the tag entered does not exist.

**Alternative Flow 2:** At step 6 *User* clicks “cancel” in prompt.

*System* returns to photo display as before flow started.

**Postcondition:**

- Tag removed from Photos.
- Tag removed from Library.
- Tag remove from Tag Display.

**UC10:** User removes photo from the library.

**Precondition:** A library with one or more photos is loaded.

**Flow of Events:**

- 1) *User* selects one or more photos (ref: UC5).
- 2) *User* selects File→Remove Photo from drop-down menu in Main window.
- 3) *System* prompts *User* to “Apply” or “Cancel”.
- 4) *User* selects “Apply”
- 5) *System* removes photo from library and reloads File Display .

**Alternative Flow:** At step 4 *User* selects “cancel”

*System* returns to state before flow started. **Postcondition:** Files that selected are removed from library and no longer appear in the File Display.

**Related Use Cases:** *none*

## 5 TRACEABILITY

	Feature											
	3.1.1.1	3.1.1.2	3.1.1.3	3.1.1.4	3.1.1.5	3.1.1.6	3.2.1	3.2.2	3.2.3	3.3.0	3.4.0	3.5.0
UC1	x											
UC2	x											
UC3		x										
UC4		x										
UC5			x	x	x							
UC6							x	x	x			
UC7						x				x		
UC8			x	x								
UC9			x									
UC10		x										

Fig. 4: Use Case/ Feature Relationship.