

# **PicSorter**

## **Software Requirements Specification**

### **Document**

Team Members: Rayne Wang, Paige Hanssen, Jacob Tower, Ryan Sellar

**Version: 1.0**

**Date: (10/22/2017)**

## Table of Contents

<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms, and Abbreviations.	4
1.4 References	4
1.5 Overview	4
<b>2. General System Description</b>	<b>5</b>
2.1 System Context	5
2.2 System Modes and States	6
2.3 Major System Capabilities	6
2.4 Major System Conditions	7
2.5 Major System Constraints	7
2.6 User Characteristics	7
2.7 Assumptions and Dependencies	7
2.8 Operational Scenarios	7
<b>3. System Capabilities and Constraints</b>	<b>8</b>
3.1 Physical	8
3.2 System Performance Characteristics	8
3.3 System Security	8
3.4 Information Management	8
3.5 System Operations	8
3.5.1 System Human Factors	8
3.5.1 System Maintainability	8
3.5.1 System Reliability	8
3.6 Policy and Regulations	8
3.7 System Life Cycle Sustainment	8
<b>4. System Interfaces</b>	<b>9</b>

**List of Figures**

Figure 2.1: Use Cases	5
Figure 2.1.1: Individual Use Case #1	5
Figure 2.1.2: Individual Use Case #2	5
Figure 2.1.3: Individual Use Case #3	6
Figure 2.3: Major Capability Groupings	6

## **1. Introduction**

### **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to present a detailed description of the software PicSorter. It will explain and document the purpose, features, requirements, and constraints of the software. This document is intended for all developers and clients to review.

### **1.2 Scope**

The software produced will be a photo sorting app called PicSorter that allows the user to delete exact duplicates. The software shall allow the user to select a directory from their computer and all the photos in the directory and subdirectories will be sorted. Duplicates shall be grouped together for the user to compare and decide which photos to delete. After all changes are made, the user shall be able to permanently save these changes on their computer. The software is intended to help users who move photos from another source to their computers. The software will help them eliminate duplicates when they accidentally import the same photo multiple times.

### **1.3 Definitions, Acronyms, and Abbreviations.**

<b>Word</b>	<b>Definition</b>
SRS	System Requirements Specification
PNG	Portable Network Graphics (type of digital image file)
JPEG	Joint Photographic Experts Group (type of digital image file)

### **1.4 References**

N/A

### **1.5 Overview**

The rest of this SRS document contains the general system description in section 2 and system capabilities and constraints in section 3. Section 2 includes overall system structure and features and section 3 includes system securities and operations.

## 2. General System Description

### 2.1 System Context

Figure 2.1: Use Cases

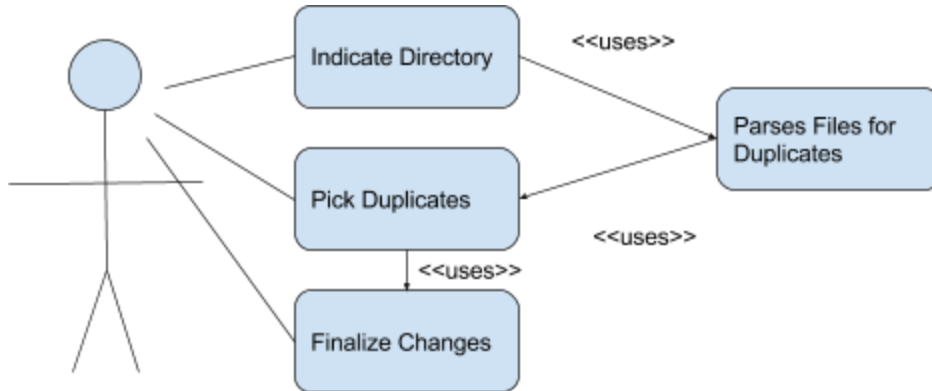


Figure 2.1.1: Individual Use Case #1

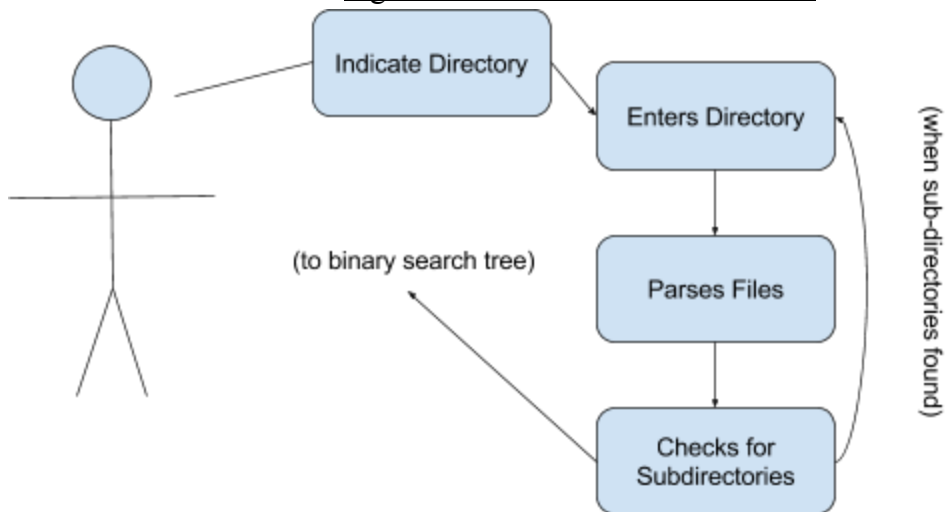


Figure 2.1.2: Individual Use Case #2

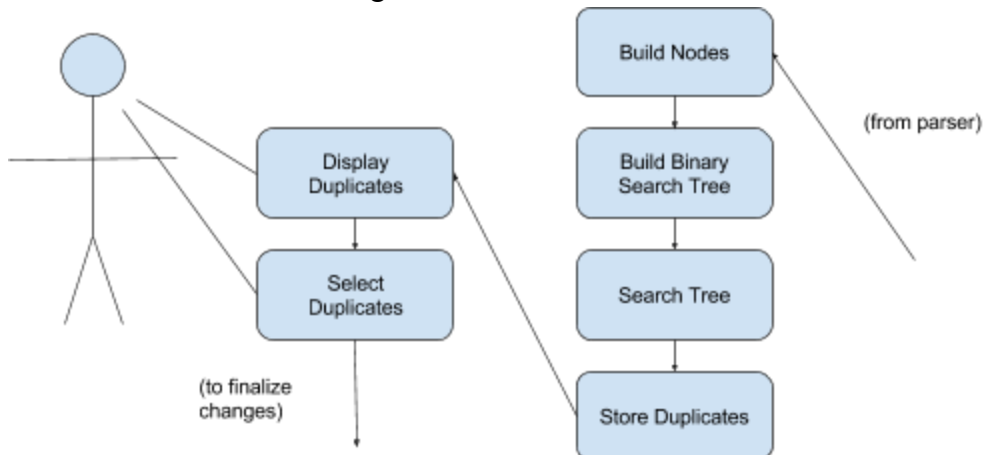
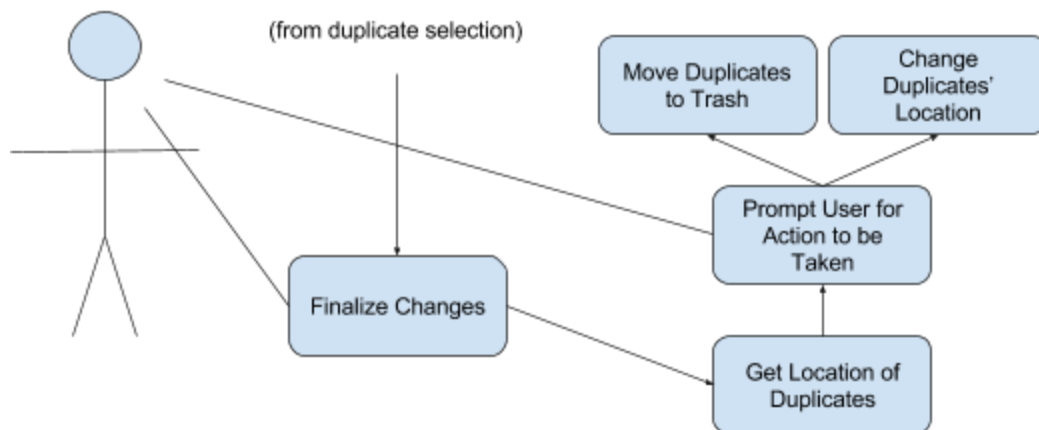


Figure 2.1.3: Individual Use Case #3



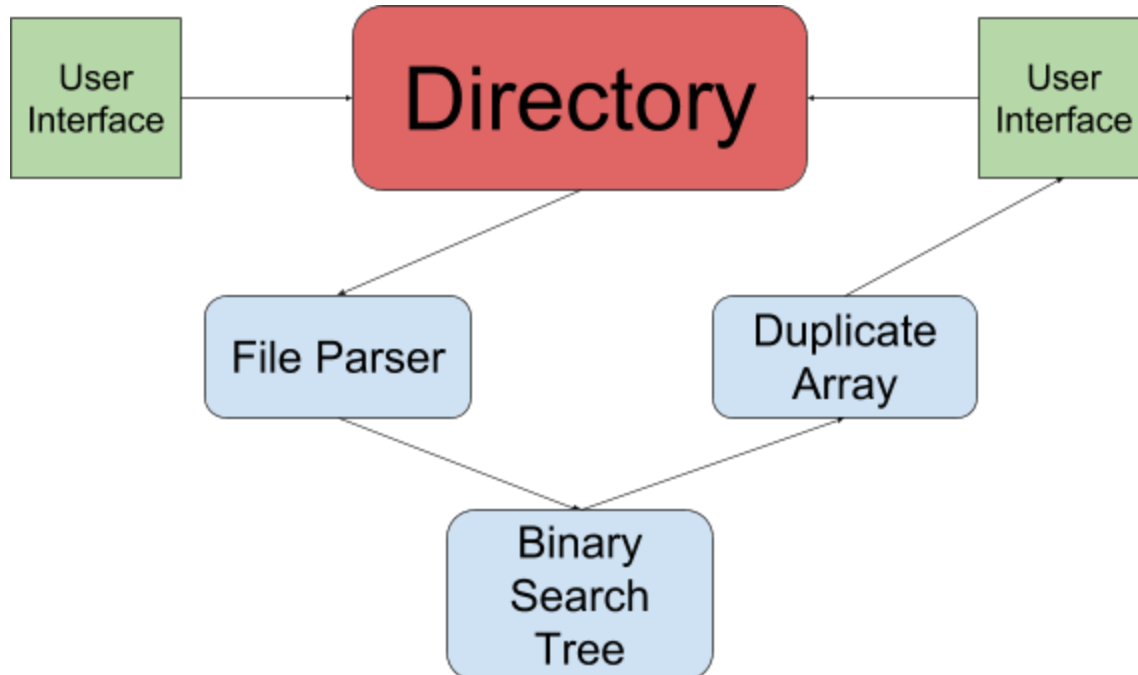
The system shall operate on files stored on the computer's hard disk, altering their location and marking them for deletion. Such operations shall only occur based on user permission. No file shall be permanently deleted without user permission. The user shall be able to indicate the operation to be performed on duplicates.

## 2.2 System Modes and States

N/A

## 2.3 Major System Capabilities

Figure 2.3: Major Capability Groupings



The program shall receive the indicated directory from the user through the user interface. The program shall be able to build a binary search tree made up of nodes using information parsed from image files. The program shall be able to search this tree for

photos which are thought to be identical based on a number of parameters and store duplicates in an array. The program shall be able to display what are assumed to be duplicates, the contents of the duplicate array, to the user through the user interface. The user shall be able to write changes to the directory and shall be able to select whether to delete duplicates or move them to another directory.

### **2.4 Major System Conditions**

The system shall operate on a directory of image files. Files in a different format than those supported shall not be able to be parsed (PNG, JPEG). The system shall have memory requirements under 512 MB.

### **2.5 Major System Constraints**

The major system constraint is on the number of photos. A binary search tree which grows too large, because of too many photos, may cause the program to crash if it takes up more memory than the system has been allocated. The system shall also be constrained by any permissions granted (or not granted) to it. Since the system requires access to the directory for file parsing and moving, this is critical to the system's operation.

### **2.6 User Characteristics**

The software has one user in the system. The user shall be able to have the software access photos in a directory and its subdirectories and find duplicates to delete.

### **2.7 Assumptions and Dependencies**

We are assuming the user's computer has enough storage to run the program because the program will have to remember photo addresses when sorting. The user shall allow the software to access the selected directories. The user shall also have to be able to run Java Virtual Machines.

### **2.8 Operational Scenarios**

The system shall be used to find photo duplicates. Users who import photos from another device to their computers shall benefit from this software because it shall allow them to delete the photos they imported multiple times. For example, a user might import photos from their phones to their computers once a month. However, they can't remember which photos they've already imported. They can import all the photos again and use PicSorter to find duplicates, compare them, and delete unwanted photos.

### **3. System Capabilities and Constraints**

#### **3.1 Physical**

N/A

#### **3.2 System Performance Characteristics**

The user shall be able to view sets of duplicate photos pulled from their directory. The sets of duplicates shall be generated in five seconds or less. This behavior shall not slow by greater than two seconds for up to five thousand pictures being searched. The program shall not crash when searching through fewer than ten thousand photos.

#### **3.3 System Security**

A user's photos shall not be saved directly in the application. Instead, an address shall be saved to pictures to access and manage them, but a user's photos shall not be remembered by the application.

#### **3.4 Information Management**

The software shall create and maintain a binary search tree containing nodes with references to the users photos. This tree shall be parsed upon the user submitting a request and all photos satisfying duplicate conditions shall be pulled to be viewed by the user. All photos shall still be stored on the user's hard drive, but may be deleted by the user from the software.

#### **3.5 System Operations**

##### **3.5.1 System Human Factors**

The system shall be able to run on its own and should not rely on human factors.

##### **3.5.1 System Maintainability**

Once the software is built and fulfills the requirements, there shall be no maintenance of the system.

##### **3.5.1 System Reliability**

A user should be able to rely on the system to handle their photos without creating permanent changes not approved by a user.

#### **3.6 Policy and Regulations**

The software shall be affected by the protection of personal information. However, the software only remembers photo addresses not the actual photo, thus, the software shall not be limited by the regulation.

#### **3.7 System Life Cycle Sustainment**

The system shall track the amount of time spent sorting through a collection of photos for use by the developers in determining whether the software is operating within its performance characteristics. Duplicates must be verified by the user, so while testing the system shall be run many times on different photo sets to ensure that it is able to accurately determine duplicates. For determining the accuracy of sorting methods, the system should also be able to display which parameters applied to a found duplicate so the developers can refine sorting methods that might return an incorrect photo.



## **4. System Interfaces**

### **User - Application Interface**

The user - software application interface shall be accessible by opening the application and shall present the user with options of actions they can take. In the application, the user shall be able to select a folder of pictures by opening the file directory. Once photos are sorted, the user shall be able to open groups of pictures to compare them. Finally, in the application, the user shall be able to delete selected pictures from groups of duplicates. See figures 2.1 and 2.1.1.

### **Application - Hardware Interface**

The application shall access the hardware system when the file directory is opened and a folder of pictures is selected for use. The application shall again access the hardware system when deleting photos so that the selected items are moved from their original address to the trash bin of a computer, where they can be recovered. See figure 2.3.