Software Requirements Specification

PicSorter

December 12, 2017

Version 4.0

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

RECORD OF CHANGES

*A - ADDED M - MODIFIED D – DELETED

VERSION NUMBER	DATE	NUMBER OF FIGURE, TABLE OR PARAGRAPH	**A* M D**	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER
1.0	10/22/2017		A	Creation of SRS	
2.0	11/14/2017	2.1, 2.1.1, 2.1.2, 2.1.3, 2.2	M	Version table created, use case and system capability diagrams redone, other small edits.	01
3.0	12/03/2017	Section 2 and section 3	M	Reorganizes requirements and added ID numbers	02
4.0	12/12/2017	2.1, 2.3	M	Updates all figures	03

Table of Contents

List of Tables and Figures

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.

Word	Definition
SRS	System Requirements Specification
PNG	Portable Network Graphics (type of digital image file)
JPEG	Joint Photographic Experts Group (type of digital image file)
FR	Functional Requirement
SC	System Constraint
U	Usability Requirement
P	Performance Requirement
S	Security Requirement

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.

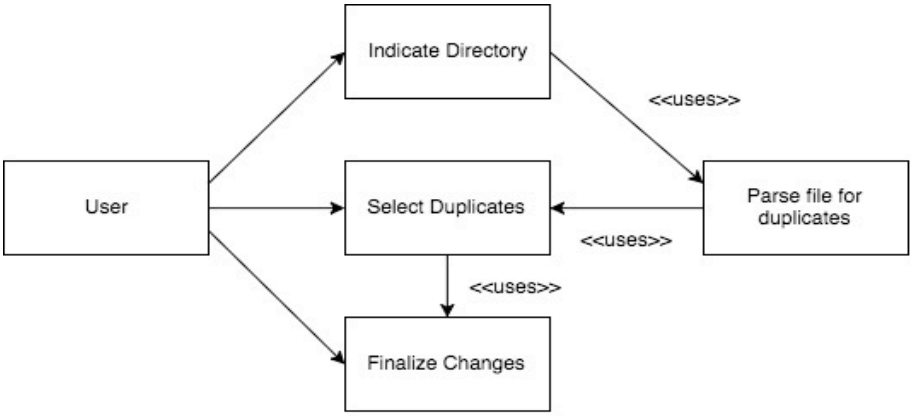
2. General System Description

Table 2.0: Functional Requirements and System Constraints

ID	Requirement
FR1.0	The program shall receive the indicated directory from the user through the user interface.
FR 2.0	The program shall be able to build a data structure that stores information parsed from image files.
FR 2.1	The program shall be able to search the data structure for photos which are thought to be identical based on a number of parameters and store duplicates.
FR 3.0	The program shall be able to display what are assumed to be duplicates to the user through the user interface.
FR 4.0	The user shall be able to write changes to the directory.
FR 4.1	The user shall be able to select whether to delete duplicates or move duplicates to another directory.
SC 1.0	The system shall operate on a directory of image files.
SC 1.1	The system shall only be able to parse PNG and JPEG images.
SC 2.0	The system shall have memory requirements under 512 MB.
SC 3.0	The system shall also be constrained by any permissions granted (or not granted) to it.

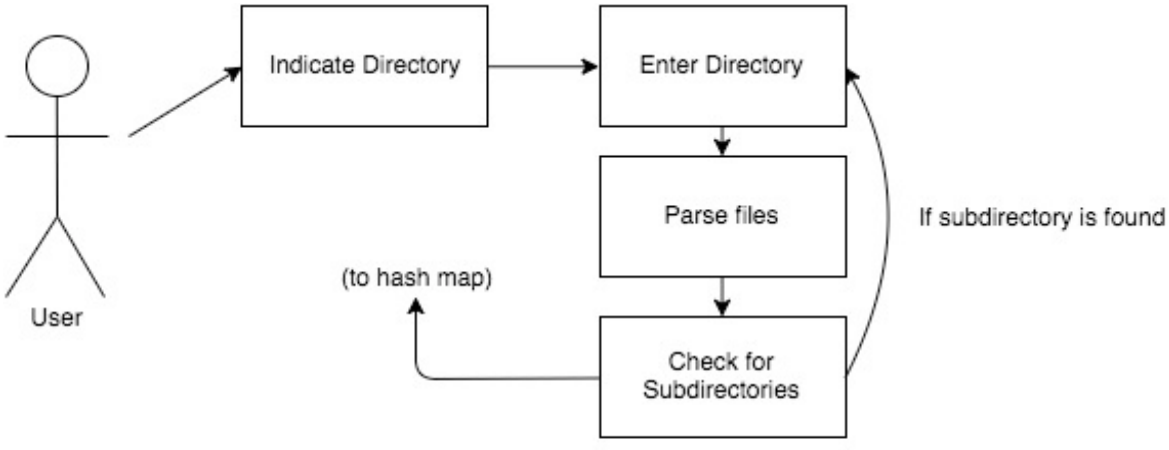
2.1 System Context

Figure 2.1: High-level Use Case



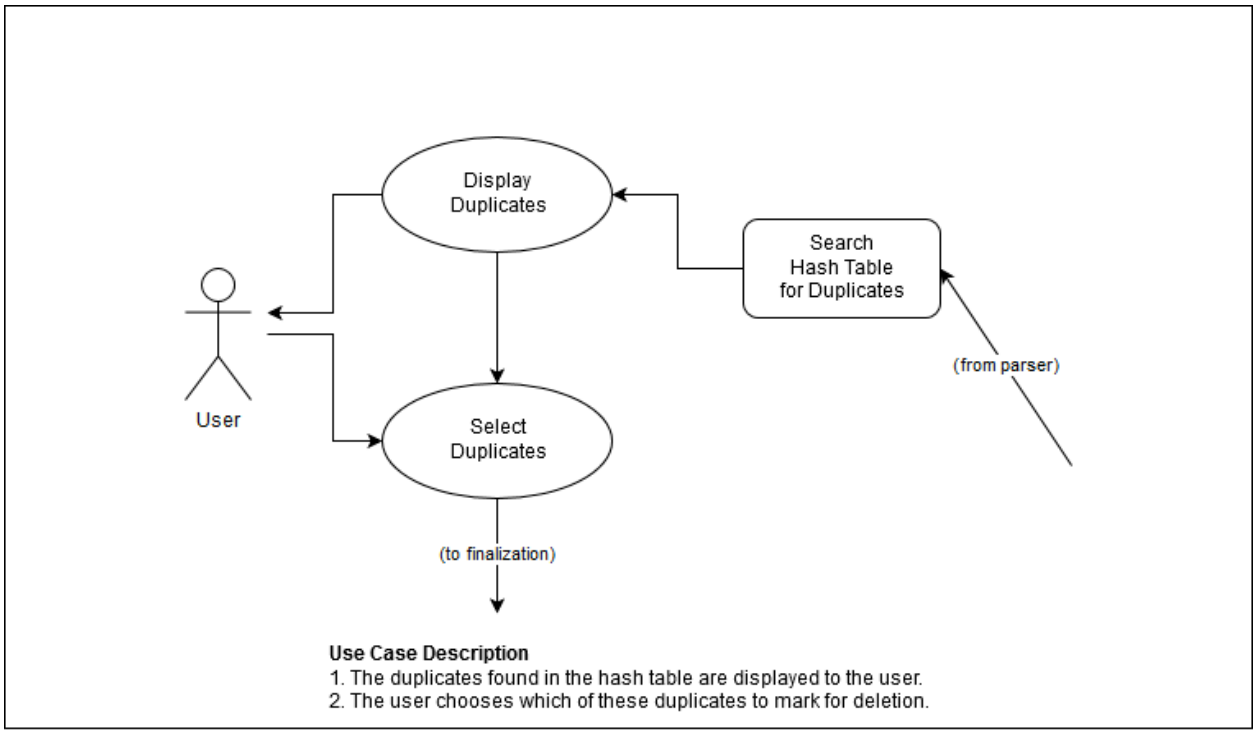
Description:
The high-level use case of the PicSorter application.
1.a. The user can selects a folder from the directory.
1.b. The user selects a group of duplicates.
1.c. The user finalizes changes to a group of duplicates.
2.a. The selected folder is searched for duplicates.
2.b. The user finalizes any changes made to duplicates.
3.a. The user can select duplicates from parsed folder.
4.a. The user can finalize any changes made to selected duplicates

Figure 2.1.2: Individual Use Case #1



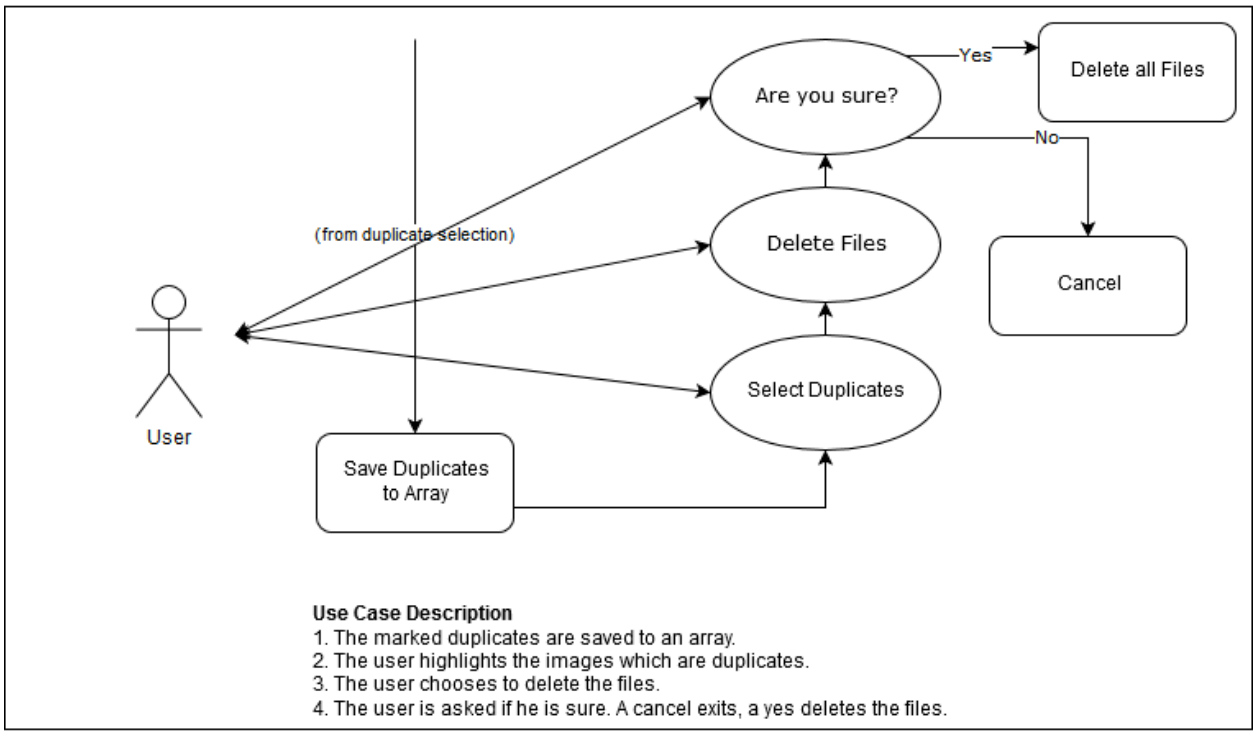
Use Case Description
1. The user indicates the directory.
2. The directory is entered.
3. Files from the directory are parsed.
4. The directory is checked for subdirectories.
5.a. If subdirectories are found, go back to step 2.
5.b. Files are put into a hash map.

Figure 2.1.2: Individual Use Case #2



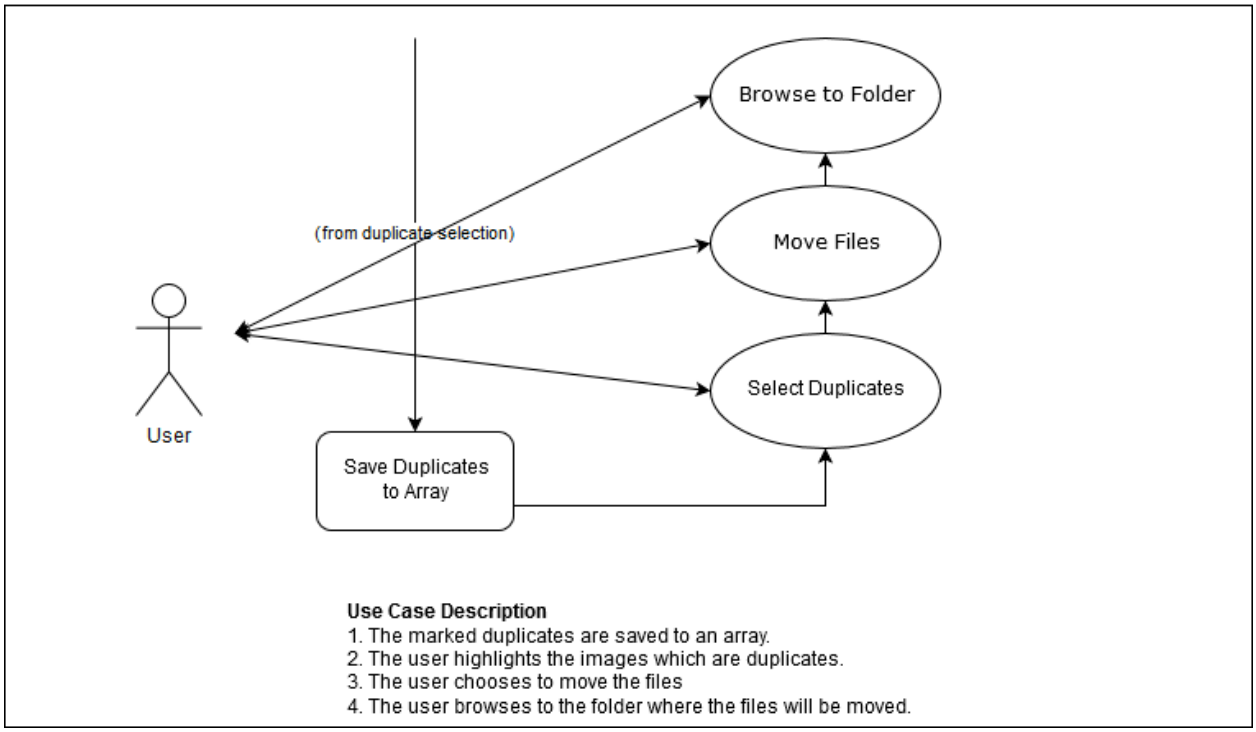
Use Case Description
1. The duplicates found in the hash table are displayed to the user.
2. The user chooses which of these duplicates to mark for deletion.

Figure 2.1.3: Individual Use Case #3



Use Case Description
1. The marked duplicates are saved to an array.
2. The user highlights the images which are duplicates.
3. The user chooses to delete the files.
4. The user is asked if he is sure. A cancel exits, a yes deletes the files.

Figure 2.1.4: Individual Use Case #4



Use Case Description
1. The marked duplicates are saved to an array.
2. The user highlights the images which are duplicates.
3. The user chooses to move the files
4. The user browses to the folder where the files will be moved.

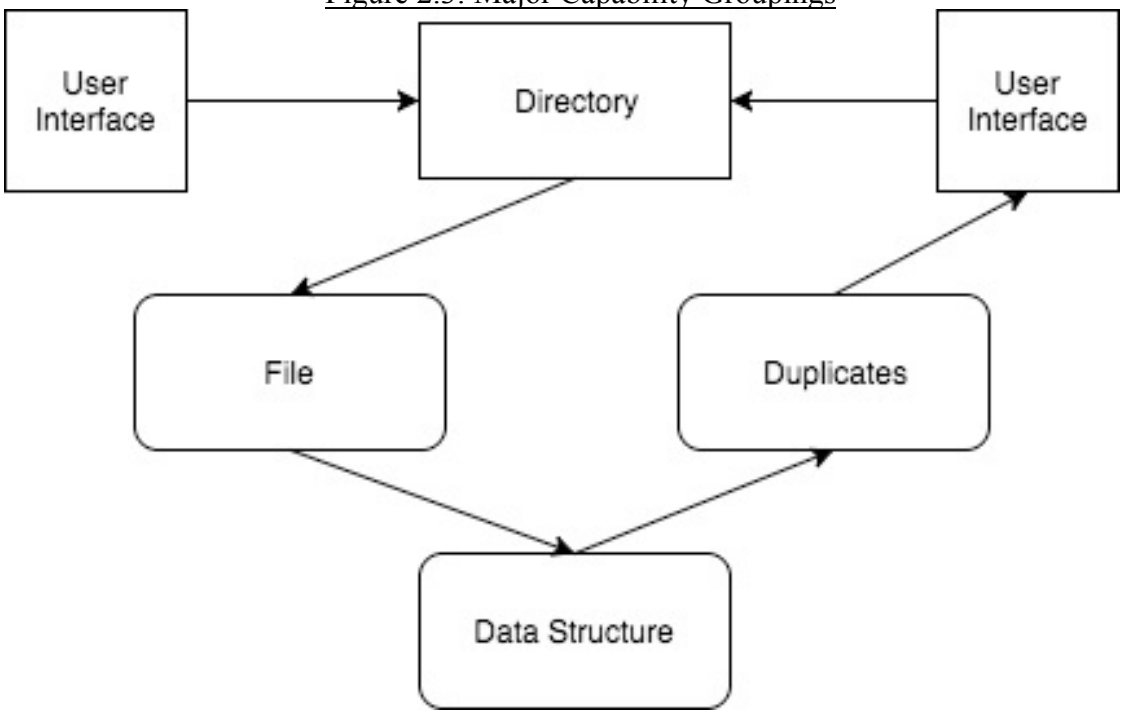
The system shall operate on files stored on the computer's hard disk, altering their location and marking them for deletion. Such operations shall only occur based on user permission. No file shall be permanently deleted without user permission. The user shall be able to indicate the operation to be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

2.2 System Modes and States

N/A

2.3 Major System Capabilities

Figure 2.3: Major Capability Groupings



See requirements FR 1.0 to 4.1.

2.4 Major System Conditions

The system shall operate on a directory of image files (see SC 1.0). Files in a different format than those supported shall not be able to parse (PNG, JPEG) (see SC 1.1). The system shall have memory requirements under 512 MB (see SC 2.0).

2.5 Major System Constraints

The major system constraint is on the number of photos. A data structure that 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 (see SC 3.0). 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 (see FR 1.0 to 4.1).

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 (see SC 3.0). 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

Table 3.0: Non-Functional Requirements

ID	Requirement
U 1.0	The program shall be able to run on its own without human factors
P 1.0	A set of 2000 photos shall be organized into a data structure in 5 seconds or less.
P 1.1	The system shall not slow by greater than one seconds per 1000 photos for up to five thousand pictures being searched.
P 1.2	The program shall not crash when searching through fewer than ten thousand photos.
S 1.0	The system shall not remember the user's photos after the user is done using the program.
S 1.1	The system shall use references not photos when possible.

3.1 Physical

N/A

3.2 System Performance Characteristics

See requirements P1.0 to P1.2.

3.3 System Security

A user's photos shall not be saved directly in the application when groups of duplicates are created (see S1.0). Instead, a reference shall be saved to pictures to access and manage them, but a user's photos shall not be remembered by the application (see S1.1).

3.4 Information Management

The software shall create and maintain a data structure with references to the user's photos (see FR2.0 and S1.1). This data structure shall be parsed upon the user submitting a request and all photos satisfying duplicate conditions shall be pulled to be viewed by the user (see FR2.1 and FR3.0). All photos shall still be stored on the user's hard drive, but may be deleted by the user from the software through either the home UI or the picture comparison UI (see FR4.1).

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 (see U1.0).

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 (see S1.1).

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.