

PROJECT MANAGEMENT PLAN
FOR THE
PicSorter

DECEMBER 12, 2017

Version 4.0

Team Name: PicSorter
Team Member 1: Rayne Wang
Team Member 2: Ryan Sellar
Team Member 3: Paige Hanssen
Team Member 4: Jacob Tower

PREFACE

This Project Management Plan (PMP) is for the development of PicSorter by a software engineering team consisting of four members: Jacob, Paige, Rayne, and Ryan. This product is intended to help the general users with finding exact photo duplicates and having the option to delete them.

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	10/8/2017		A	Creation of PMP	
2.0	11/13/2017	Section 4 and 5	M	Requirements and member roles updated	01
3.0	12/3/2017	Section 6 and 7	M	Development process updated	02
4.0	12/12/2017	Section 5	M	Schedules updated	03

Table of Contents

Section	Page
SECTION 1. OVERVIEW	1
1.1 Project Summary	1
1.1.1 Purpose, Scope, and Objectives	1
1.1.2 Assumptions and Constraints	1
1.1.3 Project Deliverables	1
1.1.4 Master Schedule and Budget Summary	1
1.2 Evolution of the Plan	1
SECTION 2. REFERENCES	2
2.1 Standards and Documents	2
SECTION 3. DEFINITIONS	3
SECTION 4. PROJECT ORGANIZATION	4
4.1 External Interfaces	4
4.2 Internal Structure	4
4.2.1 The Project Manager (PM)	4
4.2.2 Quality Manager (QM)	4
4.2.3 Configuration Manager (CM)	4
4.2.4 Scrum Master (SM)	4
4.3 Project Roles and Responsibilities	5
SECTION 5. MANAGEMENT PROCESS	6
5.1 Start-up	6
5.1.1 Estimation	6
5.1.2 Staffing	6
5.2 Work Planning	6
5.2.1 Work Activities	6
5.2.2 Schedule Allocation	7
5.3 Project Controls	7
5.3.1 Requirements Control	7
5.3.2 Schedule Control	7
5.3.4 Quality Control	7
5.3.5 Project Reporting and Communication	8
5.3.6 Metrics Collection	8
5.4 Risk Management	8
5.5 Project Closeout	8
SECTION 6. TECHNICAL PROCESS	9
6.1 Process Model	9
6.2 Methods, Tools and Techniques	9
6.3 Project Infrastructure	10
6.4 Product Acceptance	10
SECTION 7. SUPPORTING PROCESSES	11
7.1 Configuration Management	11
7.2 Independent Verification and Validation	11

7.3	Documentation	11
7.4	Quality Assurance	11
7.5	Reviews and Audits	12
7.6	Problem Resolution	12
7.8	Process Improvement	12
7.8.1	Systems/Software Process Improvement Lead	12
7.8.2	Systems Engineering Process Group	12

LIST OF FIGURES

Figure	Page
Figure 6-1. PicSorter Process Model	9

LIST OF TABLES

Table	Page
Table 4-1. PicSorter Activities and Roles.....	5
Table 5-1. PicSorter Work Schedule	6
Table 5-2. Project Plan	7
Table 7-1. PicSorter Documentation.....	11

SECTION 1. OVERVIEW

1.1 PROJECT SUMMARY

1.1.1 Purpose, Scope, and Objectives

The objective of this project is to develop a software that will assist users in finding photo duplicates and having the option of deleting them. The product will allow user to navigate to the appropriate directory, scan all the folders and subfolder for duplicates, and delete unwanted photos. The product will analyze photos based on file size and accurately find exact duplicates.

1.1.2 Assumptions and Constraints

The project is based on the assumptions that the software is designed in Java and will be used on a computer not mobile device. The project's constraints include meeting a deadline, product must be reliable and fully functional, the architecture should be open for new features and maintenance after delivery, and software must be user-friendly.

1.1.3 Project Deliverables

A fully functional software and a brief user manual will be delivered by the end of this academic semester to the appropriate file server. The software will contain at least the minimal function of finding exact photo duplicates and deleting unwanted duplicated.

1.1.4 Master Schedule and Budget Summary

The master schedule is as following:

- Project Management Planning state: due October 8, 2017
- Software Requirements Specification state: due October 22, 2017
- Software Design state: due November 5, 2017
- Implementation state: due December 10, 2017
- Testing state: due December 12, 2017
- Total Development time: 11 weeks

1.2 EVOLUTION OF THE PLAN

All members will be assigned scheduled tasks that will be completed and uploaded on Git. Unscheduled changes can also be uploaded on Git and available for all group member to review. All changes, scheduled and unscheduled, should be documented on GitHub or Google drive and all members should frequently check to stay up-to-date. Initial version of the software will only be changed and updated when all members understand and agree with the revision. During this process, group members will still be able to add on to the initial version.

SECTION 2. REFERENCES

2.1 STANDARDS AND DOCUMENTS

All documents will conform to the team's programming, documentation, and testing standards.

The standards and documents listed below are referenced in this document:

[1] Software requirements management plan (due October 22, 2017)

SECTION 3. DEFINITIONS

IDE - Integrated Development Environment

PMP - Project Management Plan

SDD – Software Design Document

SRMP - Software Requirements Management Plan

SRS - Software Requirements Specification

VCS - Version Control System

SECTION 4. PROJECT ORGANIZATION

4.1 EXTERNAL INTERFACES

The singular external organizational interface is Professor Zadia Codabux, whom we will be working with and using as our customer to develop our project.

4.2 INTERNAL STRUCTURE

The internal structure of the project organization is comprised of four team members: Paige Hanssen, Ryan Sellar, Jacob Tower, and Rayne Wang. Rayne is the Project Manager who will oversee the process and progress of the project, Ryan is the Quality Manager who ensures the quality of the project and testing, Paige is the Configuration Manager who is responsible for the VCS, and Jacob is the Development Manager who oversees the development process and helps to organize and schedule tasks. Each project team member will be responsible for completing tasks related to the development of the project itself.

4.2.1 The Project Manager (PM)

The primary role of the Project Manager is to be responsible for the scheduling of the project so that it will be completed on time. The PM is responsible for keeping team members on track and organizing team meetings when necessary. In meetings, the PM will be the chairman, and is responsible for forming and assigning tasks. Additionally, the PM is responsible for communication with external entities such as Professor Zadia Codabux, especially when it comes to handling requirements and organizing meetings. The PM is also responsible for keeping all documentations up to date.

4.2.2 Quality Manager (QM)

The Quality Manager's goal is to ensure the quality of the completed project as well as the testing process. The QM examines project documents for clarity and consistency and arranges document reviews and approvals. The QM also oversees any monitoring and testing of the project.

4.2.3 Configuration Manager (CM)

The Configuration Manager is responsible for the creation of the Git repository and maintaining that repository throughout the duration of the project. The CM ensures that the repository is being used correctly to minimize repository problems as much as possible. The CM also reviews and approves pull requests on Git.

4.2.4 Development Manager (DM)

The main role of the Development Manager is to make sure the development process is smooth. The DM is in charge of organizing and scheduling tasks so that things are clear and the team knows what needs to get done. The DM is also responsible for the improvement of team performance.

4.3 PROJECT ROLES AND RESPONSIBILITIES

TABLE 4-1. PicSorter ACTIVITIES AND ROLES

Activity	Organizational Unit
Accessing directory and photos in directory	Paige, Jacob
Organizing photos and create factory pattern	Ryan
Hash table	Rayne
Finding and comparing duplicates functionality	Ryan, Jacob
Multiple selection	Jacob
Deletion	Ryan
User Interface	Paige, Jacob

SECTION 5. MANAGEMENT PROCESS

5.1 START-UP

5.1.1 Estimation

This project will have no costs because there is no need for wages. The schedule for the project is limited by the constraints of the class, but estimation will be made using division of the project into basic tasks and building a PERT and CPM to estimate the time and order in which the tasks will be completed. This will also be useful in estimating how much slack will be available for what parts of the project.

5.1.2 Staffing

Number of staff required is unimportant. Four students will make up the staff. All staff members will take part in every step of the project, though coding tasks will be divided up among the members to ensure that multiple tasks may be worked on simultaneously.

5.2 WORK PLANNING

5.2.1 Work Activities

Work activities include the following:

- Project Management Plan
- Software Requirements Specification
- Software Design
- Implementation
- Coding
- Testing
- Deployment

The work schedule is as following:

TABLE 5-1. PicSorter WORK SCHEDULE

Due Date	Task
October 8	Brainstorm ideas and features of software
October 22	Determine requirements
November 5	Determine software design
December 5	Software implementation
November 11	-Access directory and build hash map
November 18	-Build factory pattern
November 25	-Find, compare, and delete duplicates
December 10	-User Interface
December 12	Software Testing

5.2.2 Schedule Allocation

The schedule allocation will be accomplished through the creation of a critical path network to allocate time based on the relationships between the tasks. The major exterior time constraint on the project is the end of the semester and the due date for the project itself, which is December 13th.

5.3 PROJECT CONTROLS

5.3.1 Requirements Control

TABLE 5-2. PROJECT PLAN

Task Name	Responsible	Start	End	Duration	Status
Access Directory	Paige, Jacob	11-Nov	18-Nov	7	Complete
Hash Table	Rayne	11-Nov	18-Nov	7	Complete
Entry and Entry Factory	Ryan	11-Nov	18-Nov	7	Complete
Compare Duplicate	Jacob	18-Nov	25-Nov	7	Complete
Button and Observable	Rayne	18-Nov	28-Nov	10	Trashed
Multiple Selection	Jacob	23-Nov	7-Dec	14	Complete
Deletion	Ryan	18-Nov	22-Nov	4	Complete
Additional Algorithm	Ryan	23-Nov	5-Dec	12	Trashed
User Interface (Home)	Paige	20-Nov	10-Dec	20	Complete
User Interface (Duplicate Comparision and Deletion)	Jacob	20-Nov	10-Dec	20	Complete
Testing	Rayne	6-Dec	12-Dec	6	Complete
Testing	Ryan	8-Dec	11-Dec	3	Complete
Testing	Paige	8-Dec	12-Dec	4	Complete
Tesing	Jacob	8-Dec	12-Dec	4	Complete

5.3.2 Schedule Control

The team will have weekly meetings to ensure that the project will be completed on time. For each meeting, all group members will present what he/she has done in the past week and what he/she plans to accomplish in the following week. Group members will also present problems they face and work with the team to find a solution. The project manager will compare team members' progress to the master schedule and the task assignment document to determine if the team is on track. If the team falls behind schedule, the master schedule and project plan will be modified accordingly.

5.3.4 Quality Control

Project members will continuously review and test their own code during implementation. Software design will be carefully studied to understand exactly where issues may arise should they arise. Testing will be done by a project member other than the member who wrote the code. There will also be a testing plan that will be performed after the development of the software.

5.3.5 Project Reporting and Communication

All project materials that need to be shared among group members will be shared via GitHub. Any immediate communication should be through text. The group will have weekly meetings to communicate face-to-face.

5.3.6 Metrics Collection

A shared Google Document will be used to retain project metrics. Project metrics will be collected weekly prior to our weekly meetings. Members will self-report their progress in a weekly manner and all new metrics will be validated and analyzed during the weekly meetings. New code will be uploaded on to Git and will be merged during meetings.

5.4 RISK MANAGEMENT

The risk factors and tracking methods are as follows:

- The group has limited experience. Therefore, strictly following the project plan, extensive testing, and communication with potential customers will be extremely important to ensure that project will finish on time.
- In order to avoid major design faults, extensive testing will be performed in the design stage.
- In order to avoid incompatibility between different members' programs, extensive testing will be performed on the software as a whole in addition to testing individual sub-parts.

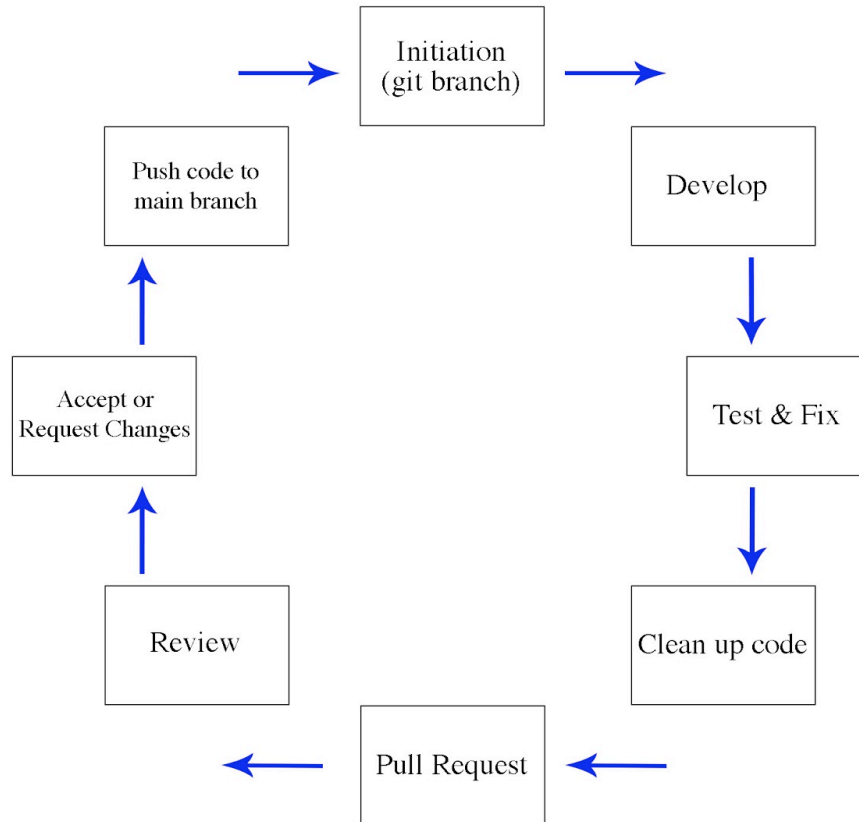
5.5 PROJECT CLOSEOUT

Project closeout will concern preparing the final presentation of the project to the class on December 13th. This will entail providing a working product and an explanation of the project's goal and development process. Also included in this final presentation will be an explanation of what was learned about software development by going through the process of developing software and how the project could have been improved based on what was learned first-hand.

SECTION 6. TECHNICAL PROCESS

6.1 PROCESS MODEL

FIGURE 6-1. PicSorter PROCESS MODEL



- Diagram to display the flow of information and separation between major work activities and supporting processes
 - Initiation (Git branch) > Develop main functionality > Test and fix until working > Additional items added > clean up code > pull request > review > changes/acceptance > push code to merge branch
- The biggest milestone that will be achieved is completing the functionality that correctly and efficiently sorts and organizes images
- Second major milestone is completing the UI, because that means the project is basically done
- Tasks will be generated every week and must be completed on time
- Large tasks will be broken up into smaller tasks or given a lengthier amount of time to complete

6.2 METHODS, TOOLS AND TECHNIQUES

Java will be the programming language of choice to build PicSorter and we will use Git as our VCS to keep everything accessible and organized. We will work on personal computers and the Colby College lab computers that are provided to develop the project. It is expected that

documents will be written cleanly in a method that is easy to read, with comments throughout each document that give brief summaries of functionality and purpose. Files, methods, titles, requests, etc. must be named clearly. When developing, team members must work from a branch separate from the main project branch. It is a standard that everything must be tested thoroughly before a Git request is made and each file must be reviewed closely by two members before it is accepted and pushed to the main branch of code.

6.3 PROJECT INFRASTRUCTURE

There is no required IDE, computer or operating system for the project. However, each team member must have access to a computer on the Colby College network from which they are able to work at any time and must have a GitHub account to access the project repository, as Git will be used as the VCS for the project.

6.4 PRODUCT ACCEPTANCE

No acquiring organization to accept our product.

SECTION 7. SUPPORTING PROCESSES

7.1 CONFIGURATION MANAGEMENT

The configuration management for this project will be done through the use of the version control software, GitHub. By saving all versions of the project through this software, all changes will automatically be logged and all members of the project group will be able to view the time and contents of changes made, as well as being made aware when conflicting versions of the project have been created.

7.2 INDEPENDENT VERIFICATION AND VALIDATION

In addition to independent evaluations of the project's validity, the project group will return to Professor Zadia Codabux who will represent the customer in order to validate that the project is being produced in agreement with the original intent of the software. Verification of the software will be done by a separate group member not responsible for writing the code being verified, and working versions of the software will be tested from the conclusion of the first iteration of work onward.

7.3 DOCUMENTATION

TABLE 7-1. PicSorter DOCUMENTATION

Document Type	Format Standard	Estimated Page Count	Peer Review Type
Project Management Plan	IEEE Std 1058-1998	12	Internal
Software Requirements Specification	IEEE Std 1233a-1998	10	Internal
Software Design Document	IEEE Std 1016TM-2009	15	Internal

The documentation for this project will be completed incrementally, with a total of three documents being produced throughout the lifetime of the project. The Project Management Plan (PMP) will be produced first, with the first version being finalized October 8th. The first version of the Software Requirements Specification (SRS) will be completed October 22nd, and the System for Distributed Database (SDD) will be completed November 5th. All documents will initially be written jointly by each member of the project group. It will then be reviewed throughout the project and kept up-to-date by the project manager until being finalized by December 12th. Professor Zadia Codabux will also complete a review of each deliverable upon completion of the first draft.

7.4 QUALITY ASSURANCE

The Quality Manager will be responsible for assuring that each document produced as a result of the project meets with the standards of each deliverables specification. All team members will

strive to meet these same standards in their contributions to the project. Third party review of each document will be done by Professor Zadia Codabux at the dates specified for the initial draft of each.

7.5 REVIEWS AND AUDITS

Through GitHub, the entirety of the code base will be stored in a central location to allow each team member to review changes. Any requests posted will be thoroughly reviewed and discussed as necessary amongst the team before the code is altered to ensure that the project stays in line with the initial intent and maintains a high level of quality.

7.6 PROBLEM RESOLUTION

Problems that occur during the project will have attention called to them through posting a request for a code update on GitHub, where all project members can view it. The team member responsible for handling the section of the code where the problem occurred will then correct the issue as needed. GitHub will track all changes made as well as requests to document all fixes that are made.

7.8 PROCESS IMPROVEMENT

The following paragraphs provide data on the PicSorter efforts for continuing process improvement.

7.8.1 Systems/Software Process Improvement Lead

The Process Improvement Lead will be responsible for leading a regular process improvement meeting at the conclusion of each main task. The Process Improvement Lead will ensure that all members at this meeting adhere to root cause analyses of the problems that they encountered during the coding process so that trends can be detected.

7.8.2 Systems Engineering Process Group

At the conclusion of each task cycle, all project group members will participate in a process improvement meeting in which significant problems that occurred during the process will be discussed with a focus on avoiding similar difficulties from arising at later stages in the project. All members will be responsible for reflecting on the aspects of their work that could be most improved on and expected to call attention to inefficiencies they have noticed anywhere in the project. Git will be used to track problems and solutions that have come up during the coding process and will be referred to during this meeting to ensure that all problems are discussed.