SE 3XA3: Development Plan
Google Images Downloader

Team 201, CAS Dream Team
Sam Crawford, crawfs1, 400129435
Joshua Guinness, guinnesj, 400134735
Nicholas Mari, marin, 400132494

January 31, 2020

Contents

1 Introduction

This document outlines many aspects of how we intend to develop our product, including our team roles, technological details, and project management.

2 Team Meeting Plan

Meetings will take place every lab session, which is twice a week on Tuesday and Wednesday from 2:30 - 4:30 pm in ITB 236, when possible. ~~If extra meetings are needed,~~ Extra meetings will be scheduled when needed on a case-by-case basis at a time and location that fits for everyone in the group.

The roles individuals take in meetings are outlined below in Section 4.

An agenda will be set by the project manager (Joshua Guinness) before the meeting and will be sent to the Facebook group chat so everyone in the group is aware. The project manager will chair the meeting and will ensure that this agenda is adhered to during the meetings. Any next steps, deliverables, or actions that need to take place will be agreed to by the individuals at the end of the meeting.

3 Team Communication Plan

In order to properly communicate between team members on important issues related to code and development, we will make use of ~~git issue~~ the git issue tracker when necessary. This also provides us with a logged record of all the issues and problems that occurred during the development process. Other communication, like discussing tasks or diagnosing bugs, will also be done via Discord and Facebook Messenger.

For team meetings and general planning, our team will make use of Discord for remote meetings and Facebook Messenger for meeting planning and coordination.

4 Team Member Roles

All members will also act as developers, designing and implementing the code for our program.

4.1 Joshua Guinness - Project Manager

- Responsible for meeting minutes and other supporting documentation
- Will ensure project deadlines are adhered to and everyone is kept on track

4.2 Nicholas Mari - Software Tester

- Responsible for testing the software and ensuring it meets our requirements

4.3 Sam Crawford - Git Specialist

- Responsible for fixing issues arising with Git, such as merge conflicts

5 Git Workflow Plan

~~We will be using~~ Trunk-based development will be used to organize the workflow of the project. ~~for our workflow. We will branch~~ Branches will be made off of the master branch only when needed for implementing a specific feature; otherwise ~~we will be working~~ work will be done in the master branch, especially for smaller updates like modifying documentation. This will ~~allow us to~~ help avoid frequent and messy merge conflicts. As we are just getting started, and since we're working with a small team, using this method of code development will allow us to develop our code and write our documentation quickly with minimal wait times or delays [1].

6 Proof of Concept Demonstration Plan

In order to provide a proof of concept for our product, the basic functionality of the application will be demonstrated. A Python script will be created that can download multiple images from the Internet given a keyword. However, as the proof of concept intends to focus on the basic functions, the additional features such as input flags and white-listing ~~, a vetting process, or a GUI~~ will not be implemented in this version. At the onset of this project, Google had changed their method of handling images; instead of having the image URLs directly in the HTML, they resolved it dynamically using JavaScript. This means that the original implementation no longer works. Since mitigating this issue will be a significant hurdle, the proof of concept will only focus on creating a product that allows the user to download images. This proof of concept will demonstrate that a solution to this problem is still feasible, despite the change in Google's service.

7 Technology

Programming Language: Python 3
IDEs: Visual Studio/Sublime Text
Testing Framework: pytest
Document Generation: doxygen
Linters: ~~pylint/~~flake8

8 Coding Style

We will be basing our style guide off of Google's style guide for Python [2], with some modifications. Linting will be done ~~as per our own discretion, either with pylint or flake8; this is subject to revision in the future and will possibly be standardized.~~ with flake8, instead of the pylint used by Google. We will not be considering Python 2 compatibility, type annotating functions or variables, or using shebang lines, unless we decide it is advantageous during the development process. All functions, variables, classes, modules, etc. will be named with camel case convention, with the first letter's capitalization depending on the type (ie. classes will use `UpperCamelCase` and functions will use `lowerCamelCase`); the rationale for this is improving readability and reducing long function names.

9 Project Schedule

The Gantt Chart can be found in the project schedule folder on the repository or by clicking the link in this section.

10 Project Review

The development process of google-images-downloader was relatively smooth, and allowed for the successful implementation of the desired program. As previously mentioned, due to Google's change to how images are stored on the Google Images results page, the original implementation of a solution to this problem was no longer functional, so a redesign of the solution was necessary. Fortunately, the original program used selenium to navigate the results page to load more images, and expanding on its use allowed for the NavigatePage module to "click" on each image to load its URL, so it could be parsed from the HTML and stored for future use. The functionality of moving the downloaded images to a server was a new feature added, which was not present in the original implementation.

There were some things in the development process that could have been improved upon. While the intention was to use the git issue tracker as a way to assign, prioritize, and keep track of ongoing tasks, this tool wasn't used very much, which led to ambiguity of responsibility for tasks, as well as a general lack of organization of what tasks were to be completed. In addition, the library used to parse the file type of a downloaded image from its data had a lot of known bugs, the main one being issues recognizing JPG/JPEG images. This was resolved someone in an ad hoc manner in the Output module, but it definitely could have been done in a more systematic way with more time, or a better alternative could be found.

References

[1] Konrad Gadzinowski. *Trunk-based Development vs. Git Flow*. https://www.toptal.com/software/trunk-based-development-git-flow.

[2] Google. *Python Style Guide*. http://google.github.io/styleguide/pyguide.html.

Table 1: Revision History

Date	**Developer(s)**	**Change**
Jan 28	Joshua	Added team member information
Jan 29	Sam	Added bibfile
Jan 29	Sam	Added section content
Jan 29	Sam	Added table of contents and Introduction
Jan 29	Nick	Added Gantt Chart link
Jan 30	Nick	Updated Gantt Chart link
Jan 30	Joshua	Spelling, grammar, sentence structure, formatting
Apr 5	Sam	Revisions for Rev 1
Apr 5	Sam	Added Project Review