Outlook

[CS 488T] Sprint 6 Report, Team 11 [stewartc]

From shelbyemailrelay@gmail.com <shelbyemailrelay@gmail.com>
Date Sun 3/16/2025 7:23 PM
To Stewart, Caleb <cstewart15@ewu.edu>

Caleb,

This report describes the activities of your EWU Senior Project team over the previous self-evaluation period (usually Saturday through Friday). It contains only public information. Private information and comments, etc. are available only to the instructor. If you notice any discrepancies or have questions, please contact Dan Tappan at dtappan@ewu.edu.

Sprint 6 Team Report

Team 11: Trademark ID & Analysis Engine

· Lane Keck
· Caleb Stewart
· Logan Taggart

Logged Hours

The team is generally free to work whenever they want during the sprint. The expectation for a team of three members is 45 hours total (15 per member) on average. However, this number will vary throughout the course.

Individual Hours:

		All Sprints							
Member	Hours	Total	Min	Max	Avg[1]	Avg[2]	Std[2]	Count[1]	Missed
Keck	8.0	45.0	3.0	9.0	7.5	7.5	2.1	6	0 (0%)
Stewart	6.0	51.0	2.0	16.0	8.5	8.5	4.3	6	0 (0%)
Taggart	8.0	40.0	3.0	9.0	6.7	6.7	2.5	6	0 (0%)
Team Total:	22.0								

[1]including and [2]excluding missed submissions for required sprints

Team Hours:

Sprint										
1	2	3	4	5	6	Total	Min	Max	Avg	Std
8.0	27.5	24.5	28.5	25.5	22.0	136.0	8.0	28.5	22.7	6.9

The following is optional descriptions of daily work that is not captured as activities below:

Taggart:

- Start working on statistics for still images
- Implementing similarity algorithm and converting app to be desktop app
- Worked on new similarity algorithm
- Optimizing desktop application
- Implementing similarity algorithm and converting app to be desktop app

Activities

Activities are member-defined units of work that are formally tracked from sprint to sprint (unlike the optional descriptions above). Every activity must be accounted for from its creation until it is completed or abandoned.

New Activities

These activities were created by during this sprint.

Keck

Activity 90: Return desired stats back to user

Update front and back end to return stats to user (two sprints expected)

Stewart

Activity 86: Implement new algorithm

After testing the new algorithm, implement into actual backend (one sprint expected)

Activity 87: Make algorithm more efficient

Try and reduce the time complexity of the algorithm (one sprint expected)

Taggart

Activity 85: Calculating image statistics

Calculate statistics for still image and prepare them for the front end. (two sprints expected)

Continuing Activities

These activities were continued from the previous sprint.

Activity 79.1: Implementing cosine similarity

Opened in Sprint 4 by Taggart; expected to take two sprints.

Original description: Getting cosine similarity to be able to recognize a specific logo in an image instead of just all logos.

Progress in Sprint 5 (expected to take one more sprint): After implementation cosine similarity alone does not work as well as was expected to recognize a specific logo, so we are currently in the process of constructing a different process.

Progress in Current Sprint (expected to take one more sprint): Our new similarity algorithm has been designed, implemented, and is working properly. We will now perform more testing on it to ensure that we have eliminated as much false positives as is possible.

Activity 84.1: Changing web app to be a desktop app

Opened in Sprint 5 by Taggart; expected to take one sprint.

Original description: Using electron.js to wrap React app

Progress in Current Sprint (expected to take one more sprint): Application now launches a desktop application when started and at same time launches the front and back end servers. There needs to be some changes made to ensure application works consistently across different devices.

Completed Activities

These activities were completed during this sprint.

Activity 76.1: Work on matching upload image

Opened in Sprint 3 by Keck; expected to take two sprints.

Original description: Match an uploaded logo with logo found by model

Progress in Current Sprint: We have completed this.

Activity 80.1: Create newly designed frontend

Opened in Sprint 4 by Keck; expected to take two sprints.

Original description: Frontend has recieved a new design, must implement it

Progress in Current Sprint: The frontend has been updated to match our desires

---

Activity 82.1: Revise Code in Backend

Opened in Sprint 5 by Keck; expected to take two sprints.

Original description: Implement new approach to comparing logos

Progress in Current Sprint: We have completed this

Activity 83.1: Proposing new Similarity Solution

Opened in Sprint 5 by Stewart; expected to take two sprints.

Original description: Find a better way to find similarities between two images

Progress in Current Sprint: Created a new algorithm solution using all 3 embedding algorithms.

Activity 88.1: Research embedding algorithms

Opened in Sprint 4 by Stewart; expected to take one sprint.

Original description: Research the best embedding algorithms to use with cosine similarity

Progress in Current Sprint: BEiT, CLIP, ResNet

Activity 89.1: Get cosine similarity working

Opened in Sprint 4 by Stewart; expected to take one sprint.

Original description: Be able to detect similarities between to images by embedding the image into the vector, and then applying cosine similarity

Progress in Current Sprint: Made this operation more accurate by combining it with Euclidean Distance

---

Activity 72.1: Setting up back end server

Opened in Sprint 2 by Taggart; expected to take two sprints.

Original description: Get Python and Flask back end to function

Progress in Current Sprint: The design and implementation of the new similarity algorithm has been completed.

Team Reflection

This section refers to the team's collective perception of and reflection on the project over this sprint.

The instructions are: Consider the following four pairs of questions hierarchically. They are not the same question. If you think they are, then you are likely not using an appropriate breadth and depth of software-engineering thought. This course is a practical application of the aspects of product, process, and people. We are trying to account for everything: not just to create a good product, but also to learn from the process to improve the people. Reflect on the experience of the entire team collectively over this sprint. You do not need to account for all work, just two examples that are most representative of easiest and hardest.

For reference, *understand* relates to the comprehension of what needs to be done; *approach* to how you think it should be solved; *solve* to implementing the actual solution; and *evaluate* to demonstrating to yourself and your team (if applicable) that the performance of your solution is consistent with everything else in the project. Remember The Cartoon from CS 350.

Understand

Easiest:	We understand how to update our project to match our changing needs. For example, our front end is constantly being updated to accomdate new backend implementation
Hardest:	It is hard to understand how our model exactly works which makes debugging difficult. The model does not tell us why it returns the things it does. False positives have been a struggle for a while

Approach

Easiest:	We want to return certain statistics back to the user such as how much space a logo takes up in an image or how confident two logos are similar. This is the easiest problem to approach
Hardest:	Implementing the ability for a user to submit a video is the hardest to approach because there are many ways to solve the problem. But down the road, we could get stuck trying out many different ways to solve this problem and realize that it may be too hard.

Solve

Easiest: Updating code in the front end is the easiest to solve because it uses React and to change anything or add anything to the website, you just make components that are easy to setup and implement.

Hardest: Turning our web app into a desktop app has been the hardest to solve this past sprint, it seems to work on one of our machines but not the rest. Solving this problem could take a while as the framework we are using is all new to us

Evaluate

Easiest: It is easy to evaluate when our project works because the model will draw a bounding box around corresponding logos.

Hardest: It is hard to evaluate how well certain embedding algorithms perform against eachother. This is why we have chosen to use all three embedding algorithms we found together.

Completion: 45

Contact: N/A

Comments: We feel confident with what we have so far, next quarter will be the real test if we can make our product fully functional as we envisioned.

Report generated on Sun Mar 16 19:23:34 PDT 2025