Outlook

## [CS 488T] Sprint 7 Report, Team 11 [stewartc]

**From** shelbyemailrelay@gmail.com <shelbyemailrelay@gmail.com>
**Date** Mon 3/24/2025 9:40 AM
**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 7 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 | 7.0 | 52.0 | 3.0 | 9.0 | 7.4 | 7.4 | 1.9 | 7 | 0 (0%) |
| Stewart | 6.0 | 57.0 | 2.0 | 16.0 | 8.1 | 8.1 | 4.0 | 7 | 0 (0%) |
| Taggart | 5.0 | 45.0 | 3.0 | 9.0 | 6.4 | 6.4 | 2.4 | 7 | 0 (0%) |
| Team Total: | 18.0 | | | | | | | | |

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

Team Hours:

| Sprint | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | Total | Min | Max | Avg | Std |
| 8.0 | 27.5 | 24.5 | 28.5 | 25.5 | 22.0 | 18.0 | 154.0 | 8.0 | 28.5 | 22.0 | 6.6 |

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

Taggart:

- Working on packaging front and back end
- Working on packaging front and back end
- Working on packaging front and back end
- Working on packaging front and back end

## 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 93:** Style new stats that are returned

Make the stats look better (two sprints expected)

Stewart

**Activity 91:** Border color

Allow the user to select the border color of the bounding box (one sprint expected)

**Activity 92:** Crop logos

Crop the logos and display results/metrics about each logo (one sprint expected)

Taggart

**Activity 94:** Packaging Front and Back end

Working on packaging each as a standalone then combining together. (one sprint expected)

## Continuing Activities

These activities were continued from the previous sprint.

**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 Sprint 6** (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.

**Progress in Current Sprint** (expected to take one more sprint): Application now works on both Windows and Mac but needs more work to package as a standalone application.

## Completed Activities

These activities were completed during this sprint.

**Activity 90.1:** Return desired stats back to user

Opened in Sprint 6 by Keck; expected to take two sprints.

**Original description**: Update front and back end to return stats to user

**Progress in Current Sprint**: Completed by Caleb

**Activity 86.1:** Implement new algorithm

Opened in Sprint 6 by Stewart; expected to take one sprint.

**Original description**: After testing the new algorithm, implement into actual backend

**Progress in Current Sprint**: Successfully implemented new algorithm into backend

**Activity 87.1:** Make algorithm more efficient

Opened in Sprint 6 by Stewart; expected to take one sprint.

**Original description**: Try and reduce the time complexity of the algorithm

**Progress in Current Sprint**: Reduced time complexity slightly by removing redundant calculations

**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 Current Sprint**: The new algorithm has been tested throughly and is accurate and robust.

---

**Activity 85.1:** Calculating image statistics

Opened in Sprint 6 by Taggart; expected to take two sprints.

**Original description**: Calculate statistics for still image and prepare them for the front end.

**Progress in Current Sprint**: The area statistic is now being calculated and displayed.

### 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:** | The easiest aspect to understand is the process of gathering and returning statistics for logos within an image. We have a clear goal of measuring logo size (by area) and similarity, and the underlying concept is simple: we are gathering data that will be useful for the end user. |
| **Hardest:** | One of the hardest aspects to grasp fully is the behavior of the image recognition model, particularly how it determines similarity between logos. Despite our best efforts, we're still encountering issues with false positives, which make it hard to predict the outcomes. The internal workings of the model are not transparent, so we're left to debug based on results without clear explanations for the decisions it |

makes. Although this isn't a huge problem in our project, this still leads to some unexpected results/issues.

### Approach

**Easiest:** Quickly adapting the frontend for new information being sent from the backend it a fairly simple and straightforward approach.

**Hardest:** We are getting to the end of just image processing and just starting to get into the realm of video processing. We need to find a way to approach processing videos efficiently, which could involve experimenting with different approaches, which could lead to delays and rework. Our current flexibility in solution design (just processing images), while necessary, is also a source of uncertainty for the problem of processing videos.

### Solve

**Easiest:** Getting the results and metrics of each image found within the image. It is just simple area calculation / the total area of the image. This returns the percentage of how much area the logo takes within the image. We also display the confidence level in the form of 'low', 'medium', 'high', which is an easier way for users to interpret how confidence the logo similarity search has been.

**Hardest:** Transforming the web app into a desktop app has been one of the most difficult challenges. While it works fine on one system, there are inconsistencies across different machines. The framework we are using (Electron.js) is new to us, so troubleshooting these cross-platform issues has been time-consuming and frustrating.

### Evaluate

**Easiest:** The easiest thing to evaluate is if our desktop app with Electron is working across multiple machines. If we can get Electron to work on both our Windows and Mac laptops, then we can say our desktop app is working.

**Hardest:** The hardest thing to evaluate right now is the time complexity of our logo similarity search. We know it takes 3-4 seconds for the similarity search to process one image, so we are unsure how this will impact processing a video. This is something we will need to try and evaluate later when we actually start to process videos.

**Completion:** 50%. We are on pace to finish.

**Contact:** N/A

**Comments:** We are on pace to have a working solution as we initially proposed at the beginning of the quarter.

Report generated on Mon Mar 24 09:40:06 PDT 2025