Outlook

[CS 488T] Sprint 3 Report, Team 11 [stewartc]

From shelbyemailrelay@gmail.com <shelbyemailrelay@gmail.com>
Date Sun 2/23/2025 5:05 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 3 Team Report

Team 11: Team 11

· 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	9.0	20.0	3.0	9.0	6.7	6.7	2.6	3	0 (0%)
Stewart	8.0	26.0	2.0	16.0	8.7	8.7	5.7	3	0 (0%)
Taggart	7.5	14.0	3.0	7.5	4.7	4.7	2.0	3	0 (0%)
Team Total:	24.5								

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

Team Hours:

Sprint							
1	2	3	Total	Min	Max	Avg	Std
8.0	27.5	24.5	60.0	8.0	27.5	20.0	8.6

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

Taggart:

- Continued work on React frontend
- Worked on proof of concept Python YOLO integration
- Research
- Research
- Worked on React frontend
- Got backend responding to basic HTTP request

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 76: Work on matching upload image

Match an uploaded logo with logo found by model (two sprints expected)

Stewart

Activity 74: Applying YOLO Model

Figuring out how to use and apply the YOLO model we trained (one sprint expected)

Activity 75: Research how to apply similarities

Research and apply how to apply similarities between logos found (two sprints expected)

Taggart

Activity 77: Implementing YOLO model in Python

Getting boundary box to be drawn around logos in a still image. (one sprint expected)

Continuing Activities

These activities were continued from the previous sprint.

Activity 70.1: How to utilize YOLO model

Opened in Sprint 2 by Stewart; expected to take one sprint.

Original description: Figure out how to use all features of the YOLO model

Progress in Current Sprint (expected to take one more sprint): Still figuring out all the features we can utilize from YOLO to make our lives easier

---

Activity 71.1: Setting up front end website

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

Original description: Get React front end to be functional

Progress in Current Sprint (expected to take one more sprint): I have got our React front end up and running, and have essential functions of program working.

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 (expected to take one more sprint): I have got the Python backend integrated and working with our trained model, and have began to setup our API routes with the Flask framework.

Activity 73.1: Integrate front end with back end

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

Original description: Get React app to work with Python backend

Progress in Current Sprint (expected to take one more sprint): I have the React app making requests to one endpoint on the backend. This still needs work, but it does provide basic functionality that is currently needed for us.

Completed Activities

These activities were completed during this sprint.

Activity 66.1: Learn YOLO

Opened in Sprint 1 by Keck; expected to take two sprints.

Original description: Learn the services that YOLO offers

Progress in Current Sprint: I can say I understand YOLO for the most part.

Activity 67.1: Creating plan for frontend and backend

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

Original description: Decide code layout for front end and back end

Progress in Current Sprint: We have a complete plan for frontend and backend

---

Activity 68.1: Implementing YOLO in backend

Opened in Sprint 2 by Keck; expected to take three sprints.

Original description: Use YOLO library to create testing code

Progress in Current Sprint: We have YOLO working in our backend for simple testing purposes

---

Activity 69.1: Train YOLO model

Opened in Sprint 2 by Stewart; expected to take one sprint.

Original description: Finish training the YOLO model

Progress in Current Sprint: We finished training the YOLO model with the LogoDet-3k dataset from kaggle

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 the frontend and backend framework we are setting up as well as how the model detects logos in images. The frontend will be able to send images using a HTTP post request and the backend processes and detects logos within the image.
Hardest:	Since we have our model working for still images the hardest part to understand is how we can continously detect a logo within a video or even track an uploaded logo within an image. These are things we are going to have to keep researching.

Approach	
Easiest:	It is easiest to approach hosting our frontend and backend as the services we are using are super easy to use and can deploy our project from GitHub.
Hardest:	Solving the 'matching uploaded logo' problem is the hardest to approach, there is a multitude of ways to solve it but recently we have had trouble making it work.

Solve	
Easiest:	Implementing the model in the backend is the easiest to solve as YOLO is easy to use and Python makes coding easier.
Hardest:	Designing a frontend website is not any of the team's strengths. Although we do not want to make our website pretty, we at least want to make it presentable. Once we figure out our video and tracking logos problem, it should be easy to implement.

Evaluate	
Easiest:	We have a working model (somewhat) it is easy to evaluate its working because the model draws bounding boxes around logos in the images. The backend sends the results to the frontend so if nothing is showing up we know its wrong.
Hardest:	It can be hard to evaluate why the model sometimes fails to detect a logo or even detects a logo that isn't there. This is a machine learning model and it can't tell you why it fails.

Completion:	15
Contact:	N/A
Comments:	The project is going well. Our only concern is the challenge of tracking a logo throughout a video.

Report generated on Sun Feb 23 17:05:41 PST 2025