CLARE: Lab Availability Tracker

Design Document

SDDec23-04

Mohamed Selim
Ahmad Nazar

Evan Uhlmeyer: Windows Tracking Developer, Meeting leader
Frank Sullivan: ML Developer, Team website and image manager
Jorge Ortiz: Canvas Integrator, Main client contact
Tyler Rolston: Linux Tracking Developer, ETG and external contact
Joel Neppel: Website Designer, Course assignment manager

sddec23-04@iastate.edu
https://sddec23-04.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

- ISO/IEC/IEEE 29119 (1 – 4)
- ISO/IEC/IEEE 23026-2015
- IEEE 1016-2009

Summary of Requirements

- Get an accurate count of the number of people in a lab
- Use a website to display the current usage of lab rooms
- Have the website be easy to use
- Integrate the website into Canvas

Applicable Courses from Iowa State University Curriculum

- Com S 227: Object-oriented Programming
- Com S 309: Software Development Practices
- Com S 319: Construction of User Interfaces
- Com S 474: Introduction to Machine Learning
- Cpr E 308: Operating Systems
- Cyb E 324: Legal, Professional, and Ethical Issues in Cyber Systems
- S E 329: Software Project Management
- S E 339: Software Architecture and Design
- S E 417: Software Testing

New Skills/Knowledge acquired that was not taught in courses

- Learning Management System platforms
- Python web frameworks
- Machine learning
- Frontend web technologies
- YOLOv8

Table of Contents

List of figures/tables/symbols/definitions

1 Team

1.1 Team Members

1) Evan Uhlmeyer

2) Frank Sullivan

3) Jorge Ortiz

4) Tyler Rolston

5) Joel Neppel

1.2 Required Skill Sets for Your Project

- Frontend/UI development
  - HTML, CSS, JavaScript
- Scripting for gathering and processing lab schedules and usage data
  - Python, JavaScript, Bash, PowerShell
- Database management
  - BudiBase, SQL
- Image processing
  - YOLOv8, OpenCV, Python

1.3 Skill Sets covered by the Team

Every team member has taken Com S 309 and has some backend and frontend development experience. Some team members have taken database management classes (such as Com S 363) or have otherwise worked with databases

1.4 Project Management Style Adopted by the team

Each member has a non-technical role to maintain team organization and a technical area they focus on.

The team creates a list of tasks and group members decide for themselves which tasks they will work on. This allows everyone to work on tasks relevant to their experience and comfort level.

1.5 Project Management Roles

- Evan Uhlmeyer: Windows Tracking Developer, Meeting leader
- Frank Sullivan: ML Developer, Team website and image manager
- Jorge Ortiz: Canvas Integrator, Main client contact
- Tyler Rolston: Linux Tracking Developer, ETG and external contact
- Joel Neppel: Website Designer, Course assignment manager

2 Introduction

2.1 Problem Statement

The lab rooms in Coover Hall at Iowa State University tend to be full many hours during the day depending on lab schedules, how many classes use it, and how close project deadlines are. Currently, there is only a limited website to check lab usage without going to the lab. This presents a problem for both Electrical and Computer Engineering (ECE) students and faculty by making it harder for students to work on classwork and harder for faculty to help students with work in the labs. Our project aims to resolve this problem by creating a website that displays current lab usage data and the lab's class schedule. This will allow both students and faculty to better plan their work in the labs.

2.2 Requirements & Constraints

- Create a website that displays the lab schedule and usage statistics
- Start with a single lab then expand to all Coover Hall labs
- Provide Canvas integration
- Have an easy way to update the lab schedules every semester
- Have lab capacity data on the website be current (updated within 10 minutes) **(constraint)**
- Website page loads within 3 seconds while on the ISU network **(constraint)**
- Website is easy to navigate and use
- Website presents data in a format which can be read quickly and easily

2.3 Engineering Standards

ISO/IEC/IEEE 29119 (1 – 4)

This standard is a collection of four international standards that apply to software testing. They include concepts and definitions used in the standard, test processes, test documentation, and test techniques. This standard applies to our project since every software project needs to have a robust approach to testing, to ensure that the project meets the requirements we set for it. (IEC 29119)

ISO/IEC/IEEE 23026-2015

This standard defines requirements for the life cycle of websites. This includes strategy, design, engineering, testing and validation, and management and sustainment for Intranet and Extranet Environment. The primary focus of our project is to develop a website to provide data to our users. This standard will be useful in defining requirements for how we manage our website after it is deployed. (ISO/IEC/IEEE 23026-2015)

IEEE 1016-2009

This standard dictates how Software Design Descriptions (SDDs) should be written. It includes the content and organization of these documents. This standard will be useful in guiding the writing of our projects design description as we move forward with our design process. (IEEE SA - IEEE 1016-2009)

2.4 Intended Users and Uses

Students

- Able to view lab schedules
- Able to see lab occupancy
- Able to see number of available workstations

Faculty

- Provide another tool to students
- Plan when to schedule office hours

TAs

- Determine when students are in the lab and would be a good time to help
- Plan when to schedule office hours

3 Project Plan

3.1 Project Management/Tracking Procedures

We will be using the agile method to manage our project. The lab availability tracker will have a public website, so we will want to get feedback from our client and users throughout its development. Agile will work better since there will be more opportunities to make usability or functionality changes before spending too much time on it.

GitLab will store our code to manage change history for our scripts and share work between team members.

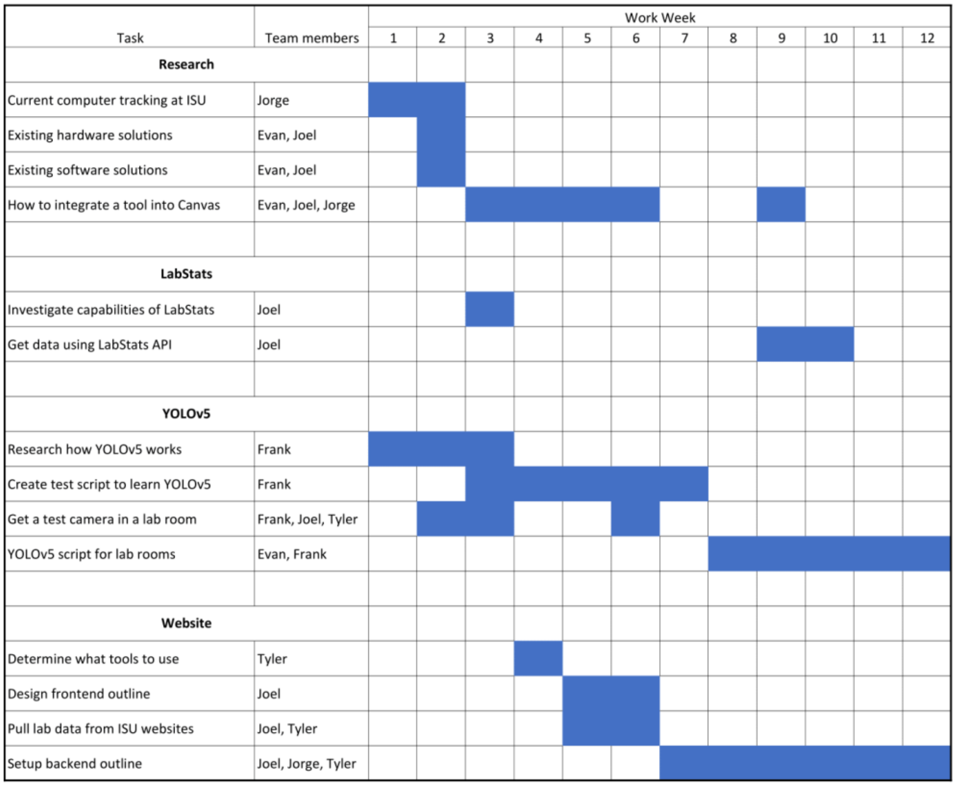
3.2 Task Decomposition

- Setup YOLOv8 to identify how many people are in a lab room
  - Python code to identify people in a controlled environment
  - Identify people in a real lab
  - Train models on various datasets
- Create a database that stores lab data
  - Add lab schedules for each lab room
  - Read and write current lab usage data
- Create a website to display lab data
  - Overview home page that displays list of tracked labs and current usage
  - Page for each lab room with more detailed information including lab schedule, capacity, current usage, and historical usage
  - Integrate website into Canvas
- Create a server to integrate components
  - Run the YOLOv8 code to get an image from a lab room and determine current lab usage
  - Write the lab usage data from YOLOv8 to the database
  - Display the current lab data on the website

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

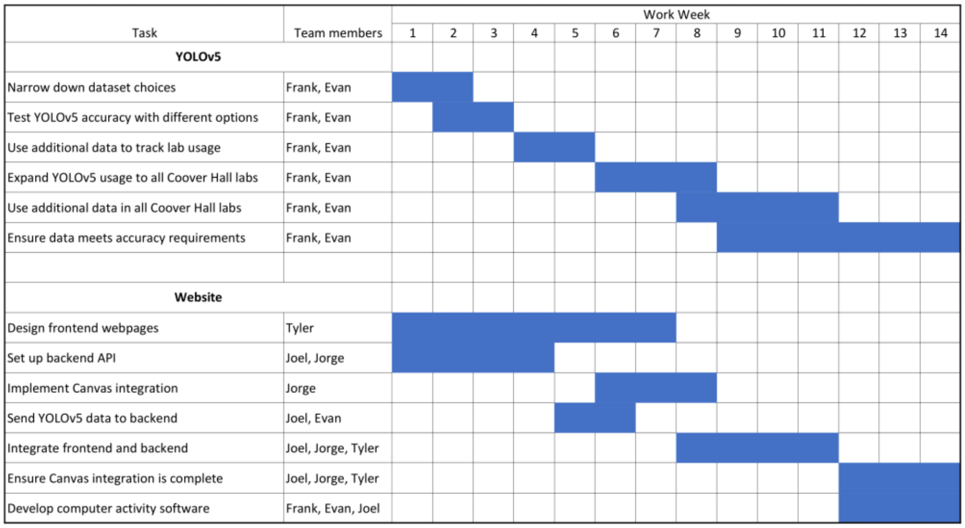
- ML will detect current lab usage at an accuracy of 90%
- Website load within three seconds of user access
- Number of labs tracked:
  - Milestone 1: Capable of tracking one lab in Coover Hall.
  - Milestone 2: Capable of tracking all labs in Coover Hall.
- Website will be integrated into Canvas and be able to be added to specific Canvas courses in platform
- Website displays accurate lab schedules for labs in Coover Hall

3.4 Project Timeline/Schedule



		Work Week											
Task	Team members	1	2	3	4	5	6	7	8	9	10	11	12
Research													
Current computer tracking at ISU	Jorge	■	■										
Existing hardware solutions	Evan, Joel		■										
Existing software solutions	Evan, Joel		■										
How to integrate a tool into Canvas	Evan, Joel, Jorge			■	■	■	■			■			
LabStats													
Investigate capabilities of LabStats	Joel			■									
Get data using LabStats API	Joel									■	■		
YOLOv5													
Research how YOLOv5 works	Frank	■	■	■									
Create test script to learn YOLOv5	Frank			■	■	■	■	■					
Get a test camera in a lab room	Frank, Joel, Tyler		■	■			■						
YOLOv5 script for lab rooms	Evan, Frank								■	■	■	■	■
Website													
Determine what tools to use	Tyler				■								
Design frontend outline	Joel					■	■						
Pull lab data from ISU websites	Joel, Tyler					■	■						
Setup backend outline	Joel, Jorge, Tyler							■	■	■	■	■	■

Gantt chart for spring semester



		Work Week													
Task	Team members	1	2	3	4	5	6	7	8	9	10	11	12	13	14
YOLOv5															
Narrow down dataset choices	Frank, Evan	■	■												
Test YOLOv5 accuracy with different options	Frank, Evan		■	■											
Use additional data to track lab usage	Frank, Evan				■	■									
Expand YOLOv5 usage to all Coover Hall labs	Frank, Evan						■	■	■						
Use additional data in all Coover Hall labs	Frank, Evan								■	■	■	■			
Ensure data meets accuracy requirements	Frank, Evan									■	■	■	■	■	■
Website															
Design frontend webpages	Tyler	■	■	■	■	■	■	■							
Set up backend API	Joel, Jorge	■	■	■	■										
Implement Canvas integration	Jorge						■	■	■						
Send YOLOv5 data to backend	Joel, Evan					■	■								
Integrate frontend and backend	Joel, Jorge, Tyler								■	■	■	■			
Ensure Canvas integration is complete	Joel, Jorge, Tyler												■	■	■
Develop computer activity software	Frank, Evan, Joel												■	■	■

Gantt chart for fall semester

- May 2023: Configure YOLOv8 to be able to accurately track people in select labs

- May 2023: Gather lab data and use that to accurately track computers currently in use in select labs
- August 2023: Set up basic website
- September 2023: Figure out how Canvas integration works and implement it to website
- September 2023: Expand YOLOv8 to accurately track people in all labs within Coover
- September 2023: Use lab data to determine computers used in all labs within Coover
- October 2023: Have website be fully functional
- November 2023: Ensure Canvas integration is complete
- November 2023: Ensure data has 90% accuracy
- December 2023: Ensure everything is completed

3.5 Risks and Risk Management/Mitigation

- Task: Setup YOLOv8 to identify how many people are in a lab room
  a. Risk: The machine learning algorithm is not accurate enough to be useful: p = 0.1
     i. Mitigation: Supplement the ML with a desktop login status
- Task: Create a database that stores lab data
  a. Risk: The ISU classes website is changed resulting in the schedule scraper breaking: p = 0.05
     i. Mitigation: Monitor the schedule updater automation and notify when it fails so it can be updated in a timely manner
- Task: Create a website that displays lab schedules and current usage
- Task: Create a server to integrate components
  a. Risk: Website is not able to scale to many users : p = 0.1
     i. Mitigation: Use a website builder tool, BudiBase, that is thoroughly tested and developed

3.6 Other Resource Requirements

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

- Cameras to monitor lab usage
- Server space
  - Database
  - Website hosting
  - Image processing

4 Design

4.1 Design Context

4.1.1 Broader Context

Area	**Description**	**Examples**
Public health, safety, and welfare	Affects the productivity of students, adds convenient options for faculty	Increases opportunities for students to use their time efficiently
Global, cultural, and social	Privacy concerns	Some students may not wish to be in view of a camera designed to identify people
Environmental	Energy usage Equipment production	Running the camera and server will consume energy, but we can make optimizations to reduce energy usage during times when lab activity is typically low We will need additional cameras that will have a manufacturing impact. Identifying if we can use old cameras can decrease new material usage.
Economic	Cost to buy cameras and run application	The total cost per lab room will need to be low and ideally will replace LabStats to save money Initial equipment cost can make app difficult to scale

4.1.2 User Needs

The ISU ECE Department needs a solution for lab tracking, in order to save money for other uses within the department.

ECE students at ISU need a reliable way to see lab availability since they are dependent on the labs in Coover Hall to complete their classes.

ECE professors at ISU need a reliable way to see lab availability in order to help their students meet the goals set for them as well as the goals students have set for themselves.

4.1.3 Prior Work/Solutions

A current lab availability website exists and uses a vendor service called LabStats to determine lab usage based on how many desktops are currently logged into. LabStats provides the basic functionality we require, including lab capacity and current lab usage, but it has limited capabilities. Not all labs will work with LabStats because LabStats does not support Linux. Lab capacity using LabStats also makes some assumptions that are not always accurate. For example, LabStats tracks users logged into desktops, so if someone comes to a lab to use other equipment and does not log into a desktop, the person will not be counted even though the space is being used.

There are many existing machine learning designs that can identify people from a camera feed. These are pre-trained models, so there may only be minor modifications required to have a reasonably accurate count of people.

Some solutions have additional hardware that can identify when someone is at a desk and can make reservations. This provides a reliable way to identify when a desk space is being used, but the additional hardware requires additional costs along with more technical management for each device.

4.1.4 Technical Complexity

Our design contains three distinct parts that will be working together to provide current lab usage data. The lab usage data will come from a camera using machine learning algorithms to identify people in a room. We will need a place to store the usage data along with other lab information, and this is where a database will come into play. The database will store lab information using multiple tables with relationships between them which provides quick and efficient access and storage of the data. Students will require a user-friendly interface to view lab data, and the website we are making will incorporate different tools and libraries to provide an elegant design.

Using a camera to identify people in a lab environment is a unique goal for machine learning. It will have people partially blocked by monitors, but we can expect people to be in a grid layout in the lab. These unique parts of identifying people in a lab will be a challenge to do reliably. The website and database for the lab data will also match industry standards to have load times people are used to along with a user-friendly interface.

4.2 Design Exploration

4.2.1 Design Decisions

1. How often will we get a new frame from the camera and run the YOLOv8 script to count how many people are in the lab?
2. What camera will we use in the labs?
3. What will we use to make the website?

4.2.2 Ideation

When considering what to use for our website, we needed to consider how to add integration for other websites and applications (such as Canvas) as well as where we would retrieve the website data from.

“What will we use to make the website?”

1. Create a website to display seating count information
2. Integrate the website into Canvas
3. Retrieve site info from data table we maintain
4. Update the website with data we collect from the camera and computer login tracking
5. Use existing website development tools to reduce our workload

4.2.3 Decision-Making and Trade-Off

We originally considered making an application that could work with existing apps like MyState, but that added a significant amount of overhead, so we decided to focus on making a web-based application with possible Canvas integration.

Retrieving data from the existing LabStats implementation is convenient but has a higher long-term cost, as well as not being compatible with labs running Linux. The downside of instead getting data from sensors or webcam feeds is that extra time must be spent developing a new system that may yield worse results. However, it has the possibility of having a lower maintenance cost.

OpenCV runs in C++ and Python, but YOLOv8 uses Python, so Python is the preferred choice.

4.3 Proposed Design

4.3.1 Design Visual and Description

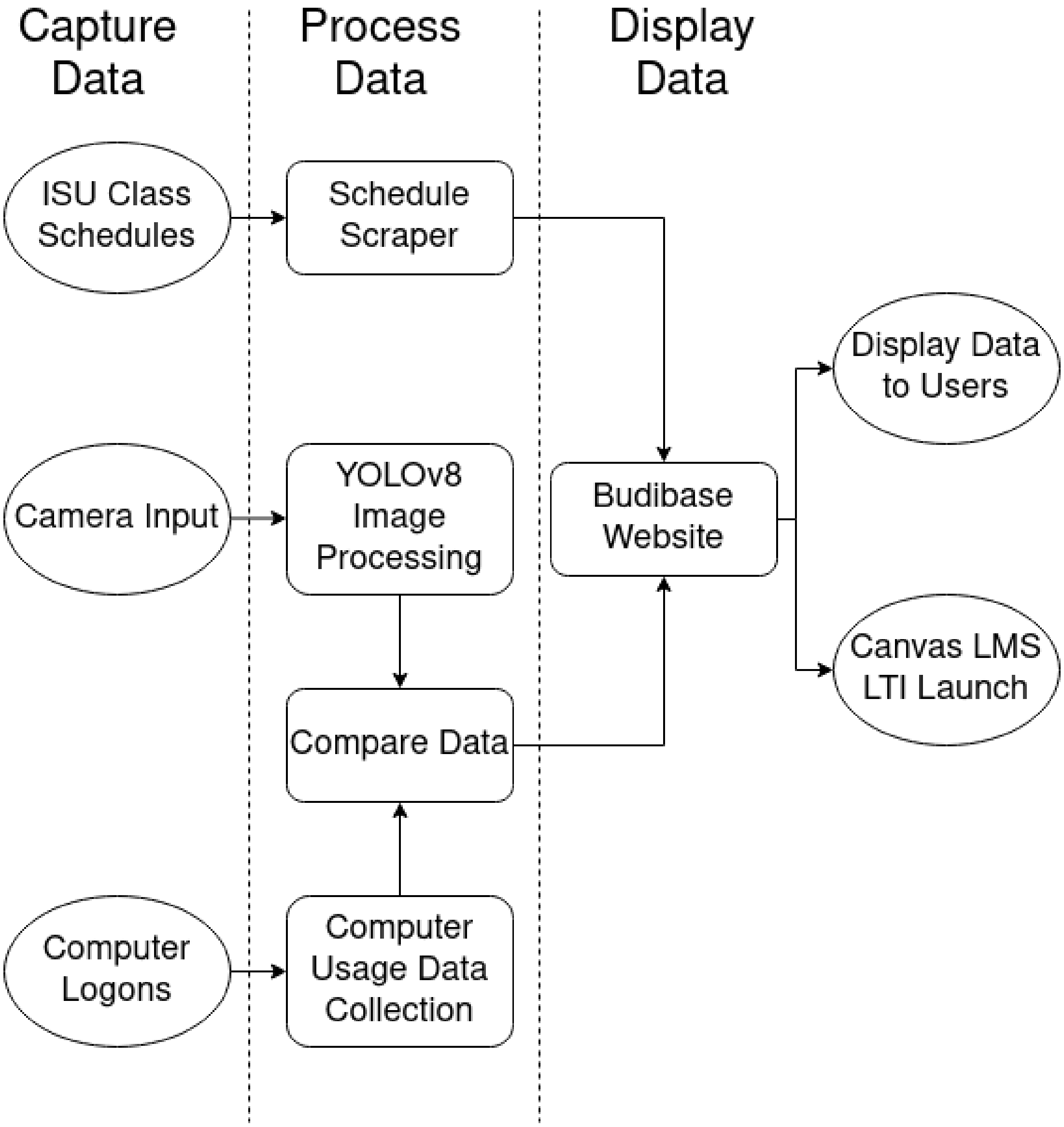


Figure 1

Our design diagram (Fig. 1) has three types of elements. The ovular elements represent Input/Output, the square elements represent software components, and the cylindrical element represents our database.

Our three inputs are at the left of this diagram, including Camera Input, desktop login tracking, and the ISU class schedule website. The input from lab cameras is fed through software running the YOLOv8 image detection framework. Camera data and desktop usage data are processed and stored on the main database. The third input, classes.iastate.edu, is an official ISU website that lab schedules are pulled from. The lab schedules will be stored on the main database as well. All this data is used by the website backend to either send to the frontend for user display or displayed in the integrated Canvas page.

4.3.2 Functionality

As people use a lab throughout the day, a website will display the current lab usage along with a schedule of the classes in the lab, as well as historical usage. The lab usage will be updated periodically using a camera to count how many people are in a lab. Students wanting to use the lab can quickly check the lab usage at any time to know if there is an open space for them.

Our proposed design will meet all of the requirements by having a website that displays the desired data with an easy-to-navigate layout and a load time of a matter of seconds. The count of people in a lab space is expected to be updated at least once every 10 minutes so that the website will always have current data.

4.3.3 Areas of Concern and Development

The data is real-time to maximize its usefulness for users, but due to the use of cameras with YOLOv8, this approach requires a lot of processing power and takes up a lot of space memory-wise. We update the camera data every 10 minutes so the data will be recent enough to be useful without constantly needing large amounts of resources. The 10-minute update interval was chosen so the usage would update at least once during the time in between classes, which is when we expect lab occupancy to change the most.

The application should be visible, unlike previous implementations, so that it can be found and used easily. Our plan to improve visibility is to use Canvas integration. This would allow students to find it on the class Canvas page. One potential area of concern is instructors not enabling the Canvas embed feature and, consequently, students not seeing it on the Canvas page. This is partially mitigated by the embed feature being easy to find in the apps, but spreading awareness of our website will still be difficult.

4.4 Design Evolution

Our design from last semester included using LabStats as a supplement for the machine learning data, but we decided to develop our own login tracking software instead; this had the added benefit of allowing us to track machines running Linux instead of only Windows. Additionally, the machine learning framework was updated from YOLOv5 to YOLOv8 due to the noticeable increase in precision. Finally, we switched to using BudiBase for the web framework instead of Flask which drastically reduced the time it took to develop the frontend.

4.5 Technology Considerations

Selecting a camera will impact how well the ML script works. A higher resolution will provide more detail to YOLOv8, but it will take longer to process.

Another big decision which web framework to use. Some, like Flask, have a lot of built-in features that simplify development. Others will require us to choose many different technologies that will offer more flexibility and control but increase development time and complexity. Using a codeless solution offers a significantly faster, easier development cycle with more stability from a thoroughly tested tool, but it will not have as much flexibility.

4.6 Design Analysis

LabStats costs the university a flat fee per computer, and our solution aims to remove that cost by creating an alternate tool.

4.7 Design Plan

The backend will be the main control point that will handle the lab data to present to the user. The camera runs on a separate script that regularly takes a still frame and counts the number of people in the room. It communicates with the backend to store the number of people in the lab. The backend combines the camera data with desktop usage and returns information requested by a user. These events may be happening simultaneously, so it will need to handle concurrent use of the data. Users access the data through a website. This main interface will need to be easy to find and get information from.

5 Testing

5.1 Unit Testing

The backend will be the main part using unit tests. The main part of the backend will be handling API requests, and each request will be handled in its own function that will be tested individually. These functions are the main way people will be getting information along with the camera giving lab usage information, so these functions will need to be very reliable. We will design any custom API functions to have a defined and limited operation so they can be easily and thoroughly tested.

5.2 Interface Testing

The main software component that will require interface testing is the API for the webserver. We will need to ensure that data provided to the API is correctly interpreted by the server, and data given to the user by the API is correctly interpreted by the user's software. We will also require interface testing between the camera and the machine learning algorithm to ensure that we interpret the image format correctly.

5.3 Integration Testing

There are a few critical integration paths in our design.

1. Integrating YOLOv8 output with login tracking data
    a. The data from both is generated very differently but needs to be compatible with each other
        i. Test by mixing YOLOv8 output with tracking data and ensuring there are no compatibility issues
2. Integrating data into our database and ensuring combined data is consistent
    a. If the data does not integrate properly into our database, it cannot be displayed
        i. Run unit tests with different sets of camera info and lab data and ensuring head count is expected
3. Integrating our website into Canvas
    a. If done improperly, Canvas integration can completely break and make our website inaccessible from Canvas, defeating the purpose of the project
        i. Ensure we properly interact with Canvas API

5.4 System Testing

System level testing will largely be providing input to the camera and checking output on the website. By providing data to the camera, it will traverse our whole system before it is displayed on the website. We will most likely not use testing tools to do this testing, but rather this will be conducted at late stages of implementation by members of the group, rather than being automated.

- Check real lab capacity by counting people in lab, ensure website is displaying the correct count for the lab
    - Check timing of accurate data as part of this testing
- Check lab schedules at classes.iastate.edu, ensure that this data is displayed accurately on the website

5.5 Regression Testing

We will be using GitLab which has some features to automate tasks, and we will use this feature to run our tests whenever a merge request is completed. The automated tests will consist of all the unit tests and any other tests that do not require human interaction to complete.

The automated test should catch most unintended changes in individual components, but it will be harder to make automated tests for the system as a whole. The manual tests to test more complex system functions will require manual actions that will take time, so we will only perform these tests monthly or after a large change.

5.6 Acceptance Testing

We will collect data on how often the occupancy number refreshes. We will compare the data by manually counting the people in a frame to validate the accuracy of our data. After implementation and release, the users of our system could be involved in a survey on their experience with CLARE.

5.7 Security Testing

For security testing, the main thing we need to worry about is the security of images being provided by the cameras. Our team will perform some light penetration testing to ensure that the security is robust and will make use of static analysis tools.

We will also perform testing for image distortion in the interest of avoiding personally identifiable information in images. There are various datasets that can be used for testing models on distorted data. We will test the performance of these datasets, using pretrained models as our baseline.

5.8 Results

All our tests are passing for our limited scope currently. BudiBase provided a strong foundation for our website that was thoroughly tested; our scripts for custom requests and automation are all short scripts that we tested with various inputs before releasing them into production.

The object detection testing showed that models trained on our chosen dataset met our accuracy requirements. Models trained on distorted data were promising, but the accuracy was not sufficient for our current implementation. The login tracking scripts work for machines running Windows and Linux.

6 Implementation

Our team split into two main teams; one team focused on data collection and processing while the other focused on development of the website.

The data team was split between machine learning object detection and computer activity data. The machine learning component uses the YOLOv8 machine learning framework to count the number of people in a room and pass the information to a database on the backend. The YOLO models were pretrained on the COCO dataset. Various datasets were used to train the YOLO models and the pretrained models were used as a baseline for comparison. An additional dataset using images retrieved from a Coover Hall lab was created and annotated. This dataset was used to test the accuracy of each machine learning model in a realistic environment.

The computer activity team further split into two teams, one for Linux and the other for Windows. Tracking on Linux is done using iNotify to monitor when a log file is modified. When this file is modified, we can infer that someone has logged in/out. Tracking on Windows is done using security events which start a task when someone logs in/out. The website has a REST API for updating the logged in status for each computer that the tracking scripts use.

The camera group will start by continuing the script to process the camera data. The script currently gets a still frame from the camera and counts the number of people in a room. Additions to the script will be to give the count of people to the backend and improve the accuracy of the count. This team will also likely research additional techniques for image recognition, such as creating our own dataset based on footage from labs.

Output from the current Computer Vision functionality can be found in Figure 2.

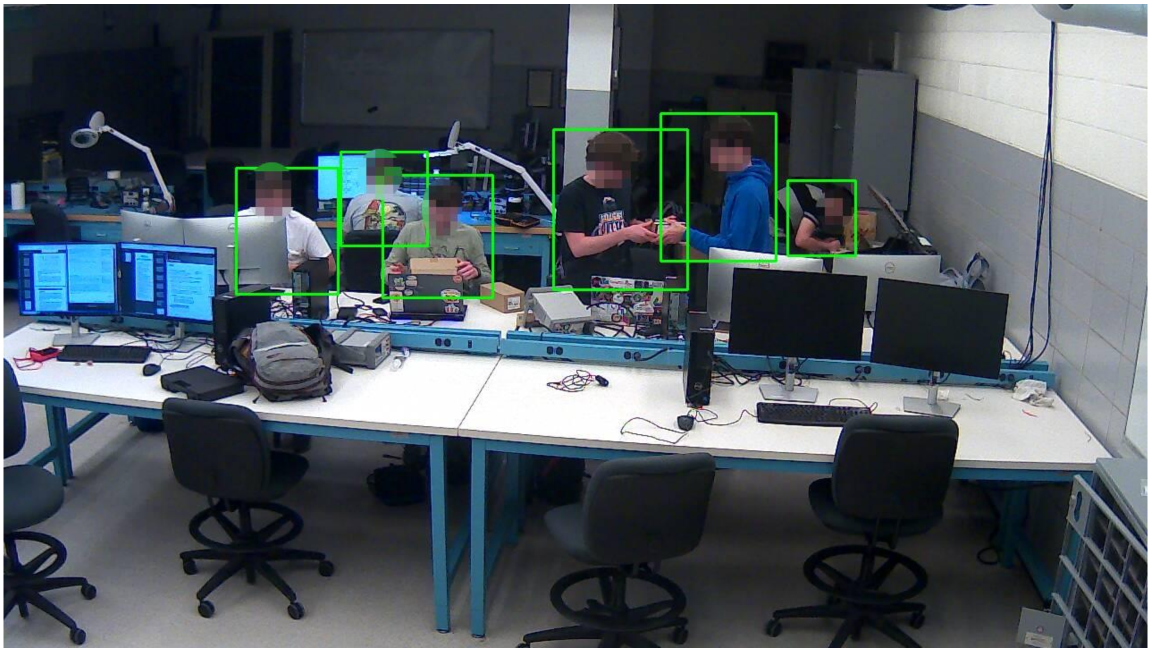


Figure 2

The website group decided to utilize a tool called BudiBase, a low-code website development solution. Using BudiBase drastically reduced the amount of time needed to create the website since it included many building blocks which included setting up hosting, connecting to the MySQL database, displaying basic tables and graphs, and handling a wide range of REST API requests. Canvas authentication was also implemented entirely within BudiBase.

Each lab room has its own page to show detailed data including the current usage and room capacity. The main feature is the lab’s schedule which is a custom table created with Svelte that displays all the classes using the lab in a clear way.

7 Professionalism

7.1 Areas of Responsibility

Area of responsibility	Definition	NSPE Canon	IEEE Code of Ethics
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	(2) A conflict of interest can reduce the quality of an engineer's work since they may not want it to be successful or will be more focused on something else (3) Maintain timeliness by providing accurate estimates (6) High-quality work can only be performed when the engineer is qualified These codes differ from NSPE because IEEE is more specific is what an engineer should do.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	(2) Having a conflict of interest may make the engineer make a decision that will benefit themself more than their client which can increase costs (4) Rejecting bribes will help keep material costs competitive

			These codes relate more to costs than the NSPE Canon.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	(1) Making decisions good for the public will be truthful and benefit stakeholders (3) Says to be honest (6) Overstating abilities is a form of deception (9) Performing false or misleading actions may damage reputations NSPE emphasizes being objective with public statements while IEEE focuses on truthfulness.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	(1) Accept responsibility for decisions that may affect the public, disclose factors that may cause harm (9) Avoid injuring others through malicious actions IEEE is different in that it specifies that not disclosing certain information may endanger the public (i.e., specific ways to value public safety).
Property Ownership	Respect property, ideas, and	Act for each employer or client as faithful	(2) Be faithful to clients and employers

	information of clients and others.	agents or trustees.	by disclosing conflicts of interest (4) Bribery will only cause someone to behave in bad faith (7) Giving proper credit to contributions is respectful and allows transparency (9) Malicious actions may damage the reputation of employers or clients This differs from NSPE in that it gives specific actions that should be taken or actions that should be avoided while NSPE just says to act faithfully.
Sustainability	Protect environment and natural resources locally and globally.	Adhere to principles of sustainable development to protect the environment for future generations	(1) Do not damage the environment This item differs from NSPE because it does not directly reference sustainability but instructs to protect the environment which is directly related to sustainability.
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and	(5) Improving the understanding of potential consequences of technology allows one to be more ethical in their conduct

		usefulness of the profession.	(9) Avoiding false or malicious actions allows someone to maintain their honor and reputation The NSPE places more emphasis on how an individual's actions affect their profession overall. IEEE does mention accepting the personal obligation to the profession, but also makes note of responsibilities toward colleagues and communities.

7.2 Project Specific Professional Responsibility Areas

Work Competence (High): We have clear cut obligations and an expectation to push out a high-quality product for the student population to use. We plan to have our product be used by the general student population, so it must be reliable and easy to use.

Financial Responsibility (Low): It is that our project will require cameras for hardware testing and cost-free software that we will develop. In the event of the project requiring additional hardware, we will need to take cost into account, but since we are limiting scope to only Coover hall, we will not need to produce more than 10 – 20 hardware units.

Communication Honesty (Medium): Since our project deals with providing data to users, we will need to ensure that our data is accurate to avoid misleading users. The data we will provide is not essential, and providing inaccurate information will inconvenience users, not endanger them. It is also important to be honest with our Client/Advisor since they have a stake in the project.

Health, Safety, Well-Being (Low): Our project will need to consider data security, especially if we implement a design that uses a video camera to monitor labs. Our project does not have any components that we expect to have an effect on the physical health or safety of our users.

Property Ownership (Medium): It is important to respect the property of the University as we implement our design. The university owns the property our product will interact with. It is also important to respect our users and protect information about them, such as which people are in a lab at a specific time.

Sustainability (N/A): Because of our project's primary focus on software, environmental sustainability is not a major focus of our group. If our project does require hardware implementation, it will be a long-standing installation, not a disposable device that needs to be recyclable or degradable.

Social Responsibility (High): The focus of our project is to provide a resource that benefits our community (ISU). At the same time, our project may reflect on ISU if it is used directly by ISU as a method to help with lab congestion.

7.3 Most Applicable Professional Responsibility Area

The most applicable responsibility area for our group is Work Competence. Our project relies most highly on our ability to perform consistent, effective work to complete our product. Each member of our group is proficient in software development, which is paramount to our project's success. It is also important for our team to meet deadlines, and be on time in general, which we have succeeded at so far. Our team has been on time for our meetings, present in lectures, and able to complete our weekly goals.

8 Closing Material

8.1 Conclusion

Our primary goal for this semester was to implement our solution in at least one lab room. We were successful in creating a website that provides lab schedules and can show the status of one lab in Coover Hall. We had hoped to expand our solution to every lab in Coover Hall, but we have shown that our implementation is viable and could realistically be implemented in other rooms.

Expanding our solution to more lab rooms and implementing our login tracking software would require more time and security testing. However, all data collection aspects of our project are functional and can update their respective fields on the database. The machine learning parts of our project meet the functional requirements, but more work could be done in regard to image distortion for security purposes.

Due to privacy and security requirements enforced by ISU, as a student project group we were not able to get our solution integrated into the Canvas live instance. We will need to continue working closely with the Center for Excellence in Learning and Teaching to get our application properly reviewed and approved before it can be added to the live instance.

Potential future work for this project includes expanding our implementation to other rooms, finalizing Canvas integration, continuing testing other datasets, and experimenting with other input sources (e.g., IR scanners and door sensors).

9 Appendix

9.1 Operation Manual

All source files used for this project are located within our groups's GitLab page through Iowa State. Files can be found at sd / sddec23-04 · GitLab (iastate.edu) . All passwords including the password for our website and our database will be handed off to Dr. Mohamed Selim upon completion of the Fall semester of 2023. Dr. Selim can be contacted at myoussef@iastate.edu.

The following sections contain operation information for each of the major components of our system.

9.1.1 Linux Tracking Software

9.1.1.1 Requisite Software

Ensure that curl and inotify are installed. These should be included in most Linux distribution kernels.

9.1.1.2 Installation and Execution

The Linux tracking software is installed via git clone. The software is run from the root user so that tracking is persistent. There is a log stored in the same directory as the executable which stores crash reports.0

9.1.2 Windows Tracking Software

Windows tracking is separated into two separate PowerShell scripts, located on the GitLab mentioned in section 9.1. To install this software, a user with administrative privileges must download these two scripts to the target computer. After these scripts are downloaded, events can be set up in the Windows Task Scheduler to run them on logons and logoffs. For the startup script, create an event in the Task Scheduler which runs on security event 4624 to listen for a successful logon. For the logoff script, create an event in the Task Scheduler which runs on the security event 4647 which listens for a user initiating logoff. Once these two events are defined and linked to their respective scripts, the computer status will be updated on logins and logoffs in the database.

9.1.3 Website and Database

BudiBase contains an import and export functionality for apps, the Tarball located in the team Git reporitority can be imported in the BudiBase settings Import/Export tab. The tabs in the main design screen are where most of the website development code is at. The data tab connects to the MySQL database and is where custom SQL queries are written. The design tab has a drag-and-drop GUI to design the webpages. The automation tab contains the scripts for generating historical lab usage.

9.1.4 Computer Vision Framework

The computer vision component of our project uses the YOLOv8 machine learning framework and OpenCV for retrieving images using cameras. A Python script is used to retrieve webcam data and perform object detection to get a “best guess” for the number of people present while another

Python script is used to update the appropriate tables in the database. A Bash script is used to link these two scripts together.

9.1.4.1 Requisite Software

The machine learning processing software is written in Python and requires several additional packages. Notably, the software uses the opencv-python and ultralytics packages to retrieve images from the webcam and perform machine learning operations on retrieved data. Additionally, the machine learning software requires a trained YOLOv8 model and must be configured such that the model is used during calls to YOLO in the script.

9.1.4.2 Requisite Hardware

A camera is required for the software to perform object detection. The software must be configured to retrieve input from the intended camera. It is recommended that the server running the scripts uses an NVIDIA GPU for the best results from YOLOv8.

9.1.4.3 Installation and Execution

The machine learning processing software is installed via git clone. Additional Python packages may need to be installed using pip. The software is run by calling the Bash script. The object detection script will be called at regular intervals and the data will be passed to the row update script.

9.2 Team Contract

Team Members:

1) Evan Uhlmeyer

2) Frank Sullivan

3) Jorge Ortiz

4) Tyler Rolston

5) Joel Neppel

Team Procedures

1. Day, time, and location for regular team meetings:

- Wednesday at 4:30, face-to-face

2. Preferred method of communication updates, reminders, issues, and scheduling:

- Discord

3. Decision-making policy:

- Consensus; discuss all ideas first, but decide as a group

4. Procedures for record keeping:

- Evan will keep meeting minutes, post them as a PDF to group members

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

- Everyone should show up on time and make an effort to attend all meetings. If they cannot, they should give notice in the team Discord.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

- All members should complete their assigned tasks on time and participate in group assignments.
- If a member is stuck on something, they should ask for help.

3. Expected level of communication with other team members:

- Attend team meetings and reply to Discord messages within a day of a message needing a response.

4. Expected level of commitment to team decisions and tasks:

- Voice any opinion they have.

Leadership

1. Leadership roles for each team member:

- Evan Uhlmeyer: Windows Tracking Developer, Meeting leader
- Frank Sullivan: ML Developer, Team website and image manager
- Jorge Ortiz: Canvas Integrator, Main client contact
- Tyler Rolston: Linux Tracking Developer, ETG and external contact
- Joel Neppel: Website Designer, Course assignment manager

2. Strategies for supporting and guiding the work of all team members:

- Having a friendly, open team dynamic where members can be comfortable asking questions.
- Doing quick check-ins when we see each other outside of team meetings.

3. Strategies for recognizing the contributions of all team members:

- Work on the weekly reports together and review the progress made by the whole team.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the

team.

- Every team member has taken Com S 309 and has some backend and frontend development experience.

2. Strategies for encouraging and support contributions and ideas from all team members:

- Every team member has company work experience that brings different management and development styles.

3. Procedures for identifying and resolving collaboration or inclusion issues:

- Team members should be direct about their issues and discuss them as they happen, so they do not build into bigger issues.
- Mention in advance if a group member anticipates other obligations (e.g., due dates for other classes) affecting their level of effort for the project

Goal setting, Planning, and Execution

1. Team goals for this semester:

- Create the basic layout and structure for a lab availability website set up
- Learn about website development

2. Strategies for planning and assigning individual and teamwork:

- Create a list of tasks and let team members volunteer for what they want to do

3. Strategies for keeping on task:

- During meetings, any team member may simply say “Let’s get back on track guys” when they feel the discussion is digressing.
- Research ahead of time, bring ideas or discussion points to meetings

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

- Discuss infractions privately or bring up concerns to group early on, attack the problem instead of the person.

2. What will your team do if the infractions continue?

- Have a group meeting to discuss infractions, speak openly about failure to meet expectations, formulate plan to move forward.
- If the situation escalates, go to an instructor or advisor to resolve issues.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1) Joel Neppel	DATE 2/19/2023
2) Frank Sullivan	DATE 2/19/2023
3) Tyler Rolston	DATE 2/19/2023
4) Jorge Ortiz	DATE 2/19/2023
5) Evan Uhlmeyer	DATE 2/19/2023