2 Project Plan

2.1 Project Management/Tracking Procedures

Which agile, waterfall, or waterfall+agile project management style are you adopting? Justify it with respect to the project goals.

We, as a team, have decided to adopt an agile project management style. In our project, having weeklong sprints to regulate our tasks will be best. Due to the close interaction with our client, getting feedback quickly after small product iterations will be important.

What will your group use to track progress throughout this and the next semester? This could include Git, Github, Trello, Slack, or any other tools helpful in project management.

Our team will use Git and GitLab to track our progress and tasks on the project. Keeping track of workload and individual modifications will be important for our sprints. Being able to lay out tasks on the GitLab Boards feature will also be helpful during sprint planning.

2.2 Task Decomposition

To solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt an agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project.

Frontend

1. Create screen sketches for the admin and volunteer sides of the application.
2. Implement UI for the Volunteer Stencil Logging application using React
    a. Add functionality to retrieve stencil information when the user inputs stencil code
    b. Add functionality to allow volunteers to update stencil status
3. Implement UI (Stencil Library and Event Management) for the admin side of the application using React
    a. Add functionality to create, update, view, and delete stencil information
    b. Add functionality to create and update information about an event (i.e., Spirits of the Garden 2023 Event)
    c. Add functionality to map stencils to a yearly Halloween event
    d. Allow admins to view statistics after an event has ended
    e. Allow admins to sort and filter stencil by desired fields
4. Implement the Stencil Recognition screen for event visitors

Backend

1. Define database schemas for stencils and event
2. Establish a secure connection to the database
3. Implement endpoints to allow creation and update of stencils and events.
    a. Create controllers to process the CRUD operations of stencils and events.

    b. Set up the request handlers that map to the appropriate controllers
4. Implement the necessary middlewares that are needed to process and store the images

Testing

1. Ensure that the volunteer and admin sides of the application are working as expected.
    a. Test functionality of the UI components with Jest or React Testing Library
    b. Test functionality of the backend APIs and controllers
    c. The preliminary user test will be done during this year's event.
2. Based on testing results, add enhancements or edit features as deemed necessary.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

What are some key milestones in your proposed project? It may be helpful to develop these milestones for each task and subtask from 2.2. How do you measure progress on a given task? These metrics, preferably quantifiable, should be developed for each task. The milestones should be stated in terms of these metrics: Machine learning algorithm XYZ will classify with 80% accuracy; the pattern recognition logic on FPGA will recognize a pattern every 1 ms (at 1K patterns/sec throughput). ML accuracy target might go up to 90% from 80%.

In an agile development process, these milestones can be refined with successive iterations/sprints (perhaps a subset of your requirements applicable to that sprint).

Milestones

1. Set up development environments for both front and backend
2. Create detailed screen sketches in Figma for the application's UI
3. Create database schemas
4. Set up CI/CD for the application
5. Implement UI for the application using React
    a. Volunteer Stencil Logging - for *volunteers*
    b. Stencil Library and Event Management - for *admins*
    c. Stencil Recognition - for *event visitors*
6. Implement backend API for event and stencil management
7. Implement AI algorithm for stencil recognition
8. Integrate frontend and backend code
9. Deploy application into the cloud server
10. Test the app during this year's Spirits of the Garden event

2.4 Project Timeline/Schedule

• A realistic, well-planned schedule is an essential component of every well-planned project

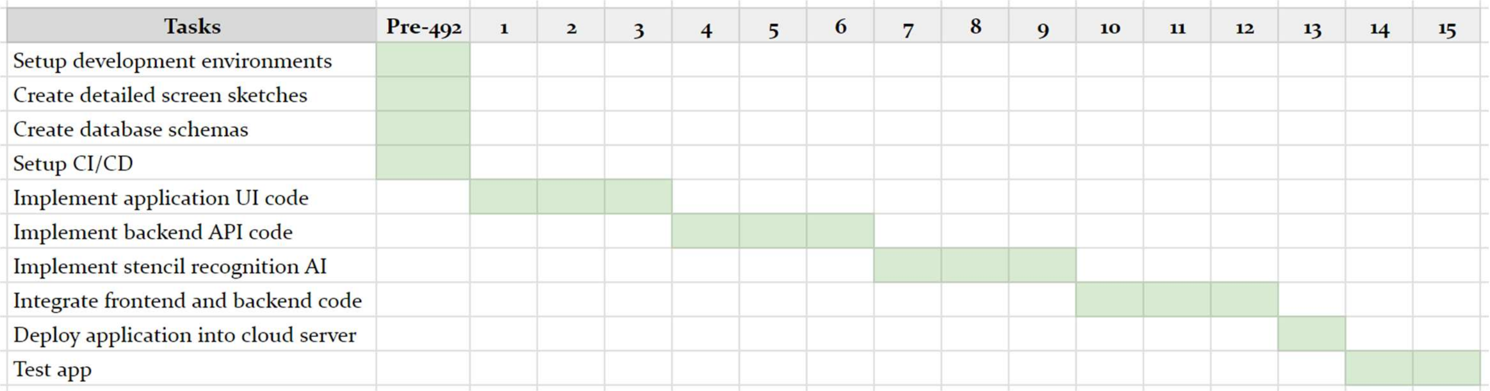
• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Project schedule/Gantt chart can be adapted to an Agile or Waterfall development model. A sprint schedule with specific technical milestones/requirements/targets will work for agile.



Tasks	Pre-492	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Setup development environments	🟩															
Create detailed screen sketches	🟩															
Create database schemas	🟩															
Setup CI/CD	🟩															
Implement application UI code		🟩	🟩	🟩												
Implement backend API code					🟩	🟩	🟩									
Implement stencil recognition AI								🟩	🟩	🟩						
Integrate frontend and backend code											🟩	🟩	🟩			
Deploy application into cloud server														🟩		
Test app															🟩	🟩

Figure 1. Project schedule/Gantt chart

2.5 Risks And Risk Management/Mitigation

Consider for each task what risks exist (certain performance targets may not be met; certain tools may not work as expected) and assign an educated guess of probability for that risk. Develop a risk mitigation plan for any risk factor with a probability exceeding 0.5. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

The agile project can associate risks and risk mitigation with each sprint.

Risks	Risk Probability	Mitigation
Not meeting the desired deadlines due to poor allocation of time	0.2	So far, the team has been very conscious of the deadlines, but to make sure we are on track we hold weekly team meetings and bi-weekly gatherings with our client and advisor to update on our statuses.
Major software or hardware issues with backend/database	0.3	Due to the random nature of the issue, we plan on working extra hours should the issue

		arise.
Excessive resource use when training AI model	0.5	To mitigate the issue, we will set limits on the resource usage and monitor the process regularly.
Model is trained incorrectly	0.6	The issue has higher probability due to the limited data we have available to train the model on. We will contact our client's previous team who are more experienced at this for advice.

2.6 Personnel Effort Requirements

Include a detailed estimate in a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in the total number of person-hours required to perform the task.

Tasks	Effort (hrs.)
Set up development environments. *This includes setting up a temporary VM for the server and using GitLab.*	10
Create detailed screen sketches. *This includes designing and finalizing the screens for Volunteer Logging, Stencil and Event Management, and AI Stencil Recognition.*	40
Create database schemas. *This includes setting up a VM as a temporary server and using GitLab for version management.*	5
Setup CI/CD. *This includes setting up GitLab to build and update deployments every push.*	5
Implement application UI code. *This includes coding the UI screens based on the sketches using React.*	70
Implement backend API code. *This includes coding the REST APIs for the backend and doing CRUD operations for entities in the database.*	50
Implement stencil recognition AI. *This includes creating and training an AI algorithm for stencil recognition.*	50
Integrate frontend and backend code. *This includes connecting the backend*	20

and frontend code using the defined APIs.	
Deploy application into a cloud server. *This includes deploying the application to our chosen cloud server.*	10
Test app. *This includes testing our app's overall functionality and effectiveness during this year's event in October. Improvements will also be made based on feedback.*	30

2.7 Other Resource Requirements

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

We are developing a product that is entirely software-based for our client. This software product must be hosted locally or remotely on a cloud server. Beyond hosting the software, the client must also supply supporting materials such as a laptop to view the software and printed QR codes for the volunteers.