

Jack-o-Lantern Tracker

DESIGN DOCUMENT

Team Number: sdded22-01

Client: Nathan Brockman

Advisor: Judith Islam

Team Members/Roles:

Kyle Goben – Team Lead, Frontend

Kiara Sta. Maria – Frontend

Omar Muhammetkulyyev – Backend

Phuoc (Johnny) Nguyen – Backend

Team Email: sddec23-01@iastate.edu

Team Website: <https://sddec23-01.sd.ece.iastate.edu/>

Revised: 04/23/2023 / Version 2.0

Executive Summary

Development Standards & Practices Used

Software practices

- Agile development
- Test-driven development (TDD)
- Continuous integration and delivery (CI/CD)
- Code reviews
- Automated testing
- Code documentation

Engineering standards

- 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes
- 29119-2-2021 - ISO/IEC/IEEE International Standard - Software and systems engineering - Software testing -- Part 2: Test processes
- 29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering
- W3C Standards for web design and application and web architecture

Summary of Requirements

Functional Requirements

- Tracking System for Stencil Status
- Stencil Recognition

Economic/Market Requirements

- The app must be maintainable throughout the years.
- The app must be accessible in the United States, especially in Ames, IA.

UI Requirements

- The UI should be intuitive and fluid across mobile or desktop browsers.
- The app must be simple and easy to use.
- The app must not show any possible, copyrighted stencil images.

Constraints

- The app must be accessible online through a browser on mobile and desktop.

- The app response time for searching and showing stencil info from AI should be very low.

Applicable Courses from Iowa State University Curriculum

- COM S 227 - Object-Oriented Programming
- COM S 228 - Introduction to Data Structures
- COM S 363 - Introduction to Database Management Systems
- SE309 - Software Development Practices
- SE317 - Introduction to Software Testing
- SE319 - Construction of User Interfaces
- SE329 - Software Project Management
- SE339 - Software Architecture and Design

New Skills/Knowledge acquired that was not taught in courses

- React/NodeJS
- Amazon Web Services
- AI Image Recognition
- TensorFlow

Table of Contents

1 Team	6
2 Introduction	7
3 Project Plan	11
3.1 Project Management/Tracking Procedures	11
3.2 Task Decomposition	11
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria	12
3.4 Project Timeline/Schedule	12
3.5 Risks And Risk Management/Mitigation	13
3.6 Personnel Effort Requirements	14
3.7 Other Resource Requirements	14
4 Design	15
4.1 Design Context	15
4.1.1 Broader Context	15
4.1.2 User Needs	15
4.1.3 Prior Work/Solutions	16
4.1.4 Technical Complexity	16
4.2 Design Exploration	16
4.2.1 Design Decisions	16
4.2.2 Ideation	17
4.2.3 Decision-Making and Trade-Off	17
4.3 Proposed Design	18
4.3.1 Design Visual and Description	19
4.3.2 Functionality	24
4.3.3 Areas of Concern and Development	24
4.4 Technology Considerations	25
4.5 Design Analysis	26
4.6 Design Plan	26
5 Testing	27
5.1 Unit Testing	27
5.2 Interface Testing	27
5.3 Integration Testing	27
5.4 System Testing	27
5.5 Regression Testing	28
5.6 Acceptance Testing	28
5.7 Security Testing (if applicable)	28
5.8 Results	28
6 Implementation	29
7 Professionalism	29
7.1 Areas of Responsibility	29
7.2 Project Specific Professional Responsibility Areas	31

7.3 Most Applicable Professional Responsibility Area	32
8 Closing Material	33
8.1 Discussion	33
8.2 Conclusion	33
8.3 References	34
8.4 Appendices	34
8.4.1 Team Contract	35

List of figures/tables/symbols/definitions

Figures

- Figure 1. Project schedule/Gantt chart (per week)
- Figure 2. Application block diagram
- Figure 3. Volunteer-side screens
- Figure 4. Admin homepage screen
- Figure 5. Admin Event Dashboard
- Figure 6. Admin Stencil Library
- Figure 7. Stencil Recognition Screens

Tables

- Table 1. Risk probability and mitigation
- Table 2. Tasks and estimated effort hours
- Table 3. Technology decision matrix

1 Team

1.1 TEAM MEMBERS

- Kyle Goben - Software Engineering
- Phuoc (Johnny) Nguyen - Software Engineering
- Kiara Sta Maria - Software Engineering
- Omar Muhammetkulyyev - Software Engineering

1.2 REQUIRED SKILL SETS FOR YOUR PROJECT

- NodeJS
- React / NextJS
- Relational database
- Experience with AWS or GCP
- CI/CD knowledge
- Client Interaction
- Teamwork skills
- Agile framework familiarity

1.3 SKILL SETS COVERED BY THE TEAM

- NodeJS
 - Kyle, Kiara, Johnny, Omar
- React / NextJS
 - Johnny, Omar, Kyle
- CI/CD
 - Kyle, Kiara, Johnny Omar
- Relational DBMS
 - Kiara, Omar, Johnny, Kyle
- Docker
 - Kyle
- AWS
 - Omar (familiar)

1.4 PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM

The team will use agile scrum methodology. This is an iterative project management methodology that consists of several relatively small iterations or releases, giving flexibility for adjustments throughout the project lifecycle. The team will iterate with sprints which will consist of finishing major features for release.

1.5 INITIAL PROJECT MANAGEMENT ROLES

- Kyle Goben - Team Lead and Client Interaction
- Kiara Sta Maria - Individual Component Design
- Phuoc (Johnny) Nguyen - Documentation Lead
- Omar Muhammetkulyyev - Testing Lead

2 Introduction

2.1 PROBLEM STATEMENT

Our project aims to solve the problem faced by Nathan, the manager at Reiman Garden, and his volunteer lead, Kathleen, during the Spirit of The Garden event held every year during the Halloween season. Hundreds of volunteers attend the event every year, and Kathleen must look up all the stencil information from the Excel file to update it, which takes up a lot of time and effort.

To solve this issue, we plan to develop a web application that both the admin and volunteers can use to update the information for the pumpkin/stencil during the event. All volunteers can update the information by themselves, saving time and effort during the event. The admin will control all printing and cutting processes and choose which stencils they want to appear during this year's event. Moreover, the admin will also update, confirm, and correct information if the volunteer updates the wrong status. The web application will help solve the problem and streamline the process, making the Spirit of The Garden event more efficient and effective.

2.2 REQUIREMENTS & CONSTRAINTS

Functional Requirements

The mobile website requirements encompass two categories—stencil status tracking system and stencil recognition.

Tracking System for Stencil Status

- a. The app must allow *admins* to...
 - **search** stencils using their code.
 - **edit stencil info** (i.e., name, category, code, image).
 - **sort or categorize** stencils as desired (i.e., by category, status, and week used).
 - **upload stencil PDFs/images** into the system (can be local as long as it's unified).
 - **update status** efficiently (status: printed, cut, traced, and carved).
 - **approve** stencils that are in the pending status queue (i.e., for ensuring that stencil code is written at the back)
 - **print stencils in one go**, using the same print settings for all stencils.
 - **select** stencils that will be used for the current year and which weeks they will be carved or displayed in the event.
- b. The app must allow *volunteers* to...
 - **input stencil code** to see its status (traced or carved).
 - click a button to **start or end the current stencil process**—tracing or carving.

Stencil Recognition

- a. The app must allow visitors and admins to use their phone cameras, take a picture of a jack-o-lantern, and display which category it's from.

Economic/Market Requirements

- The app must be maintainable for the admin users to be used throughout the years (i.e., let variables be settings editable for admins).
- The app must be accessible in the United States, more importantly, inside Ames, IA.

UI Requirements

- The UI should be intuitive and fluid across mobile or desktop browsers.
- The app must have a search box to type in stencil code for searching.
- The app must include as few clicks as possible for each operation.
 - i. For example, the app must have a parent radio button that will be used to specify the current process. If the current day focuses on carving pumpkins, the radio button will be set to “Carve” status (to avoid multiple button clicks when updating the status).
- The app must show thumbnails of stencil images so that users can know how it looks without loading another page (i.e., during the stencil selection process).
- The app must have a save button (or automatic save) so that user progress on edits/selections would not be lost.
- The app must not show any possible copyrighted stencil images.

Constraints

- The app must be accessible online through a browser on mobile and desktop.
- The app response time for searching and showing stencil info from AI should be very low.

2.3 ENGINEERING STANDARDS

1. **12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes ([link](#))**

This standard will guide this project’s development as it provides a common framework for software lifecycle processes. The project will utilize processes that are defined in this standard that are geared towards agile software development as well as pointers for requirements gathering and documentation.

2. **29119-2-2021 - ISO/IEC/IEEE International Standard - Software and systems engineering - Software testing -- Part 2: Test processes ([link](#))**

This standard will guide software testing practices in developing the web app. The group will consult manual and automated testing practices defined in this standard.

3. **29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering ([link](#))**

This standard will serve as a guide for good requirements definition on the software and how to apply these requirements effectively on the web application.

4. **W3C Standards for web design and application and web architecture ([link](#))**

This is a helpful resource for standards or recommendations for web development. The standards/recommendations regarding mobile web interface, APIs, HTML, CSS, and HTTP will be helpful for this project.

2.4 INTENDED USERS AND USES

1. Volunteer Admins

These are the primary users of our application who are the volunteers supervising the Spirits in the Gardens event. The admins want to make tracking the stencils' statuses less tedious and exhausting. They want an easy and fast-to-use application to help run the event efficiently. These superusers can:

- a. Initiate a new event for the upcoming year.
- b. Specify how many weeks the event will be.
- c. Enter, modify, and delete the stencil information from the system.
- d. Scroll through the stencils and select the ones that will be used in the next event.
- e. Select which stencils will be used for week one or two of the event.
- f. Simultaneously print stencils using the same printer settings.
- g. Print all stencils that were not brought back for cutting.
- h. Search for any stencil by its code.
- i. Update stencil status to Printed, Cut, Traced, or Carved.
- j. Approve stencils in the pending queue so they can check if the volunteer has added the stencil code at the back of the pumpkin.
- k. View stencil information easily by taking a picture of the carved pumpkin and identifying the stencil it was carved from.
- l. Sort or categorize stencils as desired.

2. Volunteers

The subcategory of primary users also includes the volunteers who help trace and carve the pumpkins at the Reiman Gardens. The volunteers will be interested in accessing the application through their Android or iOS devices and experiencing a user-friendly UI. As such, they will have only a subset of the admins' capabilities. Particularly, volunteers can:

- a. Search for any stencil information by entering its code.
- b. Submit a status update on the pumpkins showing whether they are traced or carved for the admins to approve.
- c. Take a picture of the carved pumpkin and get information about its stencil.

3. Visitors

The secondary users of our application will include the visitors who attend the Spirits in the Gardens. They will be interested in the following:

- a. Recognizing the stencil the pumpkin was carved from using their cameras with an AI
- b. Finding more information about the stencil, including its title, category, source, etc.
- c. Voting for the most liked stencil/pumpkin.

3 Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

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.

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.

3.2 TASK DECOMPOSITION

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 the 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.

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

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

3.4 PROJECT TIMELINE/SCHEDULE

Tasks	Pre-4Q2	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 (per week)

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

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 resource usage and monitor the process regularly.
Model is trained incorrectly	0.6	The issue has a 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.

Table 1. Risk probability and mitigation

3.6 PERSONNEL EFFORT REQUIREMENTS

Tasks	Effort (hrs.)
Set up development environments. <i>This includes setting up a temporary VM for the server and using GitLab.</i>	10
Create detailed screen sketches. <i>This includes designing and finalizing the screens for Volunteer Logging, Stencil and Event Management, and AI Stencil Recognition.</i>	40
Create database schemas. <i>This includes setting up a VM as a temporary server and using GitLab for version management.</i>	5
Setup CI/CD. <i>This includes setting up GitLab to build and update deployments every push.</i>	5
Implement application UI code. <i>This includes coding the UI screens based on the sketches using React.</i>	70
Implement backend API code. <i>This includes coding the REST APIs for the backend and doing CRUD operations for entities in the database.</i>	50
Implement stencil recognition AI. <i>This includes creating and training an AI algorithm for stencil recognition.</i>	50
Integrate frontend and backend code. <i>This includes connecting the backend and frontend code using the defined APIs.</i>	20
Deploy application into a cloud server. <i>This includes deploying the application to our chosen cloud server.</i>	10
Test app. <i>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.</i>	30

Table 2. Tasks and estimated effort hours

3.7 OTHER RESOURCE REQUIREMENTS

AWS server for Web Application:

- Amazon Simple Storage Service (S3)
- Amazon CloudFront
- Amazon Route 53
- AWS Identity and Access Management (IAM)

Carved pumpkin image data set to train the AI stencil recognition.

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

Public health, safety, and welfare

The project does not have an immediate, crucial impact on our stakeholder's health and well-being aside from providing convenience in their current processes. The number one goal of the project is to improve the efficiency of the user's pumpkin tracking, which in return can eliminate or reduce their stress and frustration.

Global, cultural, and social

The final web application product will potentially improve the work processes for the Reiman Gardens volunteer group. The hope is that volunteer practices geared toward the annual Halloween event will be more efficient and desirable.

Environmental

The project does not have any major environmental impact. Users will just utilize the final project as a desktop or mobile web application, which does not directly contribute to environmental hazards.

Economic

An economic concern that needs to be watched carefully is server cost. The project needs to use web resources in such a way that cost is as minimized as possible. As a result, the size and frequency of server requests should be carefully assessed during development.

4.1.2 User Needs

1. **Volunteer Admins** need to initiate a new event for the upcoming year, specify how many weeks the event will be, and enter, modify, and delete stencil information: to efficiently manage and organize the Spirits in the Gardens event. They need to select stencils for the upcoming event, print all stencils that were not brought back for cutting, search for any stencil by its code, and update stencil status: to ensure all stencils are available and ready for the event. They also need to view stencils' information and sort or categorize stencils: to maintain the quality of the event and ensure smooth operation.
2. **Volunteers** need to search for any stencil information by entering its code, submit a status update on the pumpkins showing whether they are traced or carved for the admins to approve, take a picture of the carved pumpkin, and get information about its stencil: to easily and effectively trace and carve pumpkins.
3. **Visitors** need to recognize the stencil that was carved on the pumpkin by using their cameras with AI, find more information about the stencil, including its title, category, source, etc., and vote for the most liked stencil/pumpkin: to enjoy and engage in the Spirits in the Gardens event.

4.1.3 Prior Work/Solutions

Our project has been specifically designed to meet the needs of our customers, and currently, there is no similar software available in the market. While a previous group of computer science majors has attempted a similar project, they only completed a sample product with a low level of completion, and we have not considered reusing their ideas.

If our group completes this web application, we anticipate the following advantages and disadvantages:

- Advantages:
 - A visually appealing and user-friendly interface.
 - Fewer operational steps and time saved compared to using pure Excel spreadsheets.
 - Comprehensive application where the admins can manage the event besides simply keeping track of the pumpkins.
 - Platform-independent solution for volunteer side use, i.e., iOS and Android users can access the app.
- Disadvantages:
 - The AI recognition system may not be completely accurate.
 - The server may require significant resources when there are many visitors.
 - Hosting the Spirits in the Gardens event may cost more than it did before.

4.1.4 Technical Complexity

Our project is of sufficient technical complexity based on the following metrics:

The project's design consists of two main components: front and backend. The frontend handles user interactions and displays information, while the backend processes and stores information. Additionally, we plan to integrate AI to identify stencils engraved on the pumpkin, a complex problem requiring extensive research.

The problem scope contains challenging requirements that match or exceed current solutions or industry standards. The project aims to create a web application instead of a mobile app to reduce maintenance costs and increase user convenience. The system will allow volunteers to update information directly into the system by phone, reducing the time and effort required for manual updates. We will also classify which system functions are used by which users to simplify operations. Furthermore, the project prioritizes research to complete the system's main functions before integrating AI for stencil identification.

4.2 DESIGN EXPLORATION

4.2.1 Design Decisions

1. Using React with Next.js for frontend/backend development.
2. MySQL will be used as a database management system.
3. Extending the app's intended users to include volunteers.
4. Developing web applications over Android or iOS native applications

4.2.2 Ideation

Using React/Next.js for frontend/backend development:

1. React with Next.js: We chose this option because of its popularity and ease of use, especially for building server-rendered React applications. This option also allows for easy code sharing between the client and server.
2. Angular: Angular is a popular JavaScript framework that offers robust features for developing scalable and dynamic web applications. However, we opted not to use it due to its steep learning curve and the team's lack of experience with the framework.
3. Vue.js: Vue.js is another popular JavaScript framework known for its simplicity and performance. However, we felt its ecosystem and community were less mature than React's, making it less suitable for our project.
4. Django with Python: Django is a Python-based web framework that offers many built-in features for web development, such as authentication and routing. However, we felt this option would be less flexible than React with Next.js.
5. Ruby on Rails: Ruby on Rails is a web framework designed for rapid development and known for its convention over configuration approach. However, we did not choose this option because the team lacked experience with Ruby.

4.2.3 Decision-Making and Trade-Off

To identify the pros and cons of the ideated options for our project's frontend and backend development, we used a decision matrix, which helps compare different options based on their criteria.

We identified the following criteria for our decision matrix:

1. Popularity and community support
2. Ease of use and learning curve
3. Performance and scalability
4. Features and capabilities
5. Compatibility with other technologies used in the project
6. Team's Familiarity

We then assigned a weight to each criterion based on its importance to the project. The weight scale ranged from 1 to 5, with 5 being the most important. We assigned the following weights to the criteria:

1. Popularity and community support: 5
2. Ease of use and learning curve: 4
3. Performance and scalability: 4
4. Features and capabilities: 3
5. Compatibility with other technologies used in the project: 2
6. Team's Familiarity: 5

Here are the results of our decision matrix:

Option	Popularity and Community Support	Ease of Use and Learning Curve	Performance and Scalability	Features and Capabilities	Compatibility with Other Technologies	Team's Familiarity	Total Score
React with Next.js	5	4	4	5	5	4	102
Angular	4	3	3	4	4	2	74
Vue.js	4	4	3	4	5	3	85
Django with Python	3	4	4	3	4	2	74
Ruby on Rails	3	3	3	3	3	1	59

Table 3. Technology decision matrix

Based on the results of our decision matrix, we chose to use React with Next.js as the best option for our project. It had the highest score of **102**, indicating that it was the most suitable option based on our criteria.

4.3 PROPOSED DESIGN

Our team has made significant progress toward developing the web application. We began by creating a detailed Figma design for the volunteers and admin website, which helped us to visualize the different features and functionalities required for the system. We have also implemented a user-friendly UI for volunteers and established a reliable connection to the database on the server. We are still in the design phase and discussing with customers, so we have not been able to test any functionality of the project.

4.3.1 Design Visual and Description

I. Block Diagram

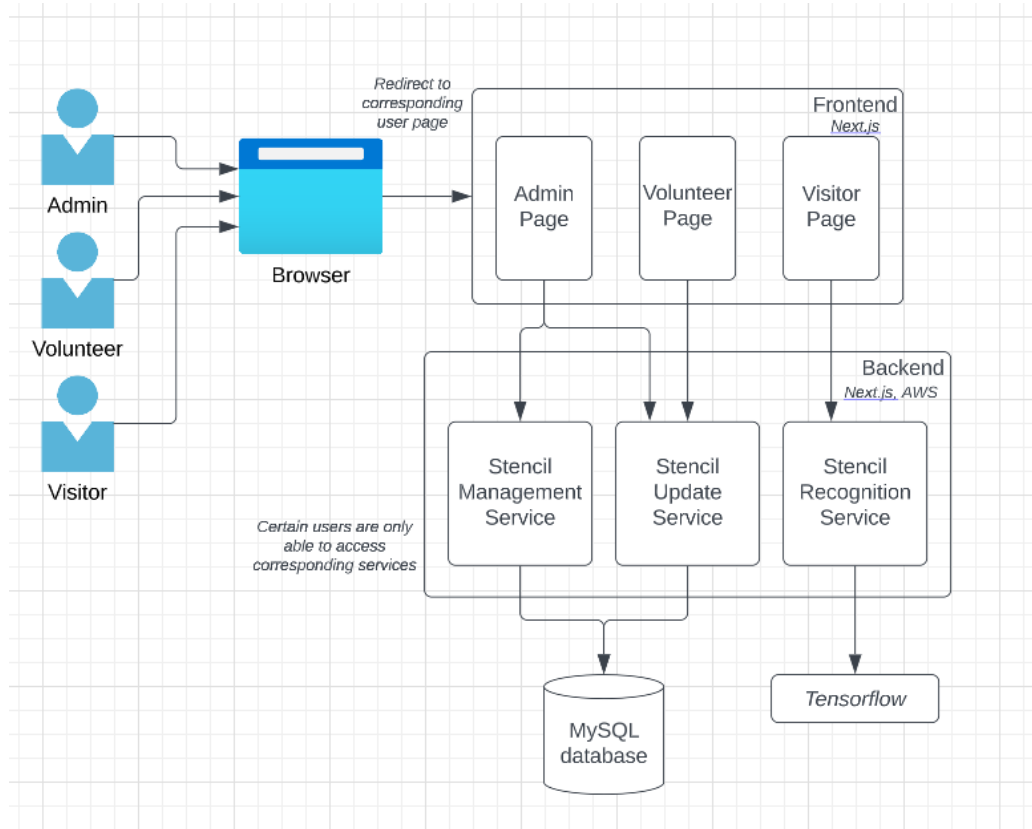


Figure 2. Application block diagram

Users will access the app using a browser. Admins will mainly use a desktop browser, while volunteers and visitors will use a mobile browser. These users will access the app through different URLs showing the corresponding pages. The frontend and backend sides of the app will be written using the Next.js framework. The stencil recognition will use Tensorflow. Each frontend page uses a certain service in the backend. Stencil management pertains to event-stencil mapping, whereas stencil update pertains to the creation, deletion, or update of stencil information. The stencil recognition part of the app will use Tensorflow for model/image training.

II. Screen sketches

- **Volunteer-side Pages** - for stencil status updates during the event

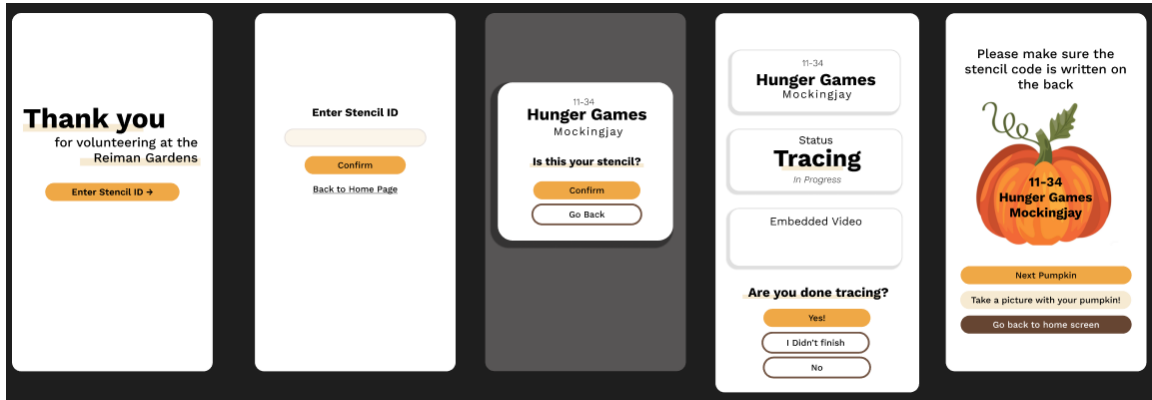


Figure 3. Volunteer-side screens

The image above is our current user interface design for our project's Volunteer Stencil Logging part with the following details:

- A homepage screen with a Thank you message for volunteers.
- Input textbox to enter the stencil ID a volunteer has chosen to trace or carve into a pumpkin.
- A confirmation page where the volunteer will verify if the code they have inputted is correct.
- Stencil dashboard page where the volunteer can see the current status of the stencil they are working on and update the status of their stencils as they finish. If the volunteer has finished tracing or carving, they can move on to the next pumpkin, where they will enter a stencil ID again (second page).

- **Admin-side Pages** - for stencil library and stencil management

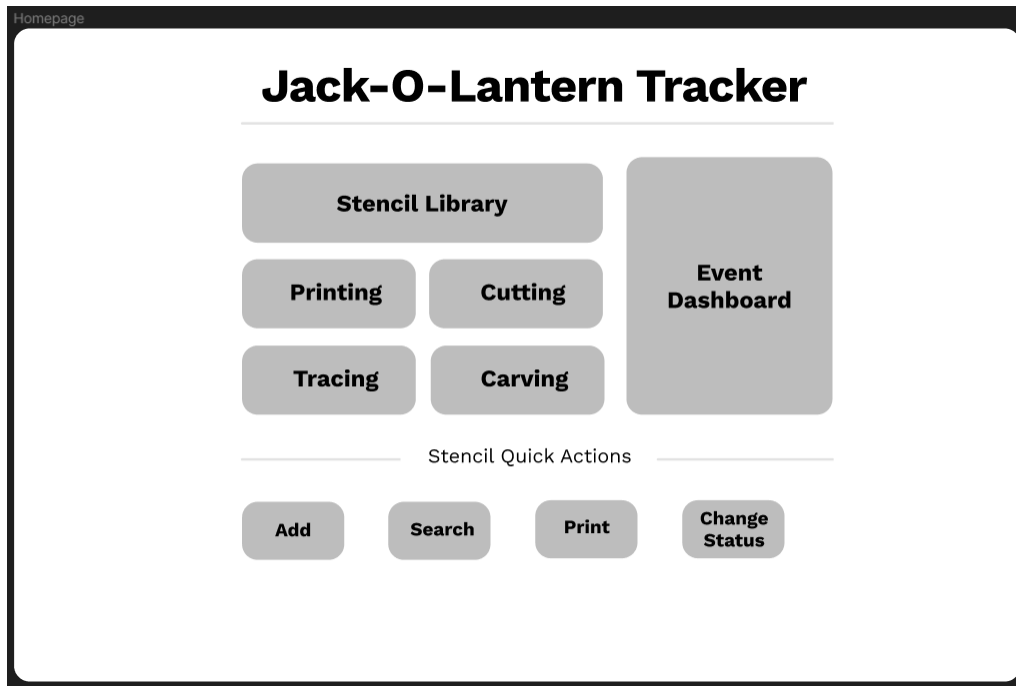


Figure 4. Admin homepage screen

The image above is the admin homepage with the following details:

- The stencil library and the event dashboard are the two main navigation screens. These two screens will be explained further below.
- Users can also select Printing, Cutting, Tracing, and Carving statuses; stencils with the selected status for the current event will be shown.
- Quick stencil actions include adding, searching, printing, and updating stencils. These quick actions will conveniently redirect the user to the action. For example, the add quick action will show the add new stencil screen in the stencil library.

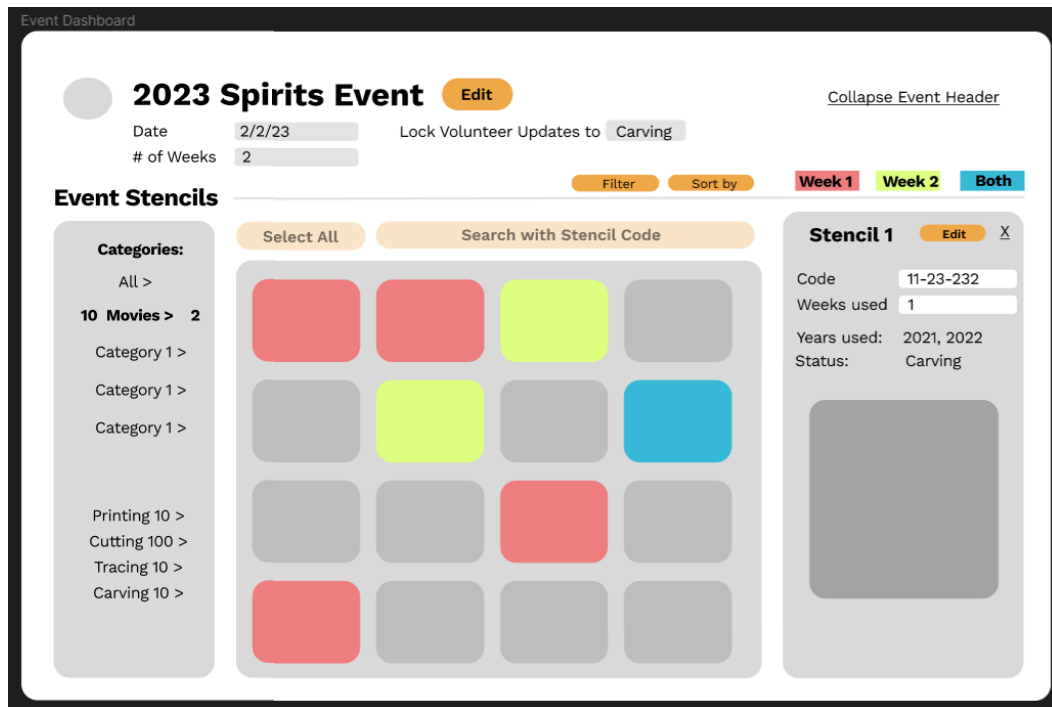


Figure 5. Admin Event Dashboard

The image above is the current design for the admin’s event dashboard page with the following details:

- A header to display or edit the general information about the Spirits of the Garden event for the current year.
- A navigation bar at the left side wherein a user can select a category or status, filtering the displayed stencils.
- Search for a stencil using a stencil code.
- Filter and sort by stencil subfields (i.e., category, most recently used, least recently used, etc.) at the top right where displayed stencils will be updated as the user desires..
- Legends at the top right for which week a stencil is used during the event.
- A display form at the right side of the screen to show and edit stencil information.

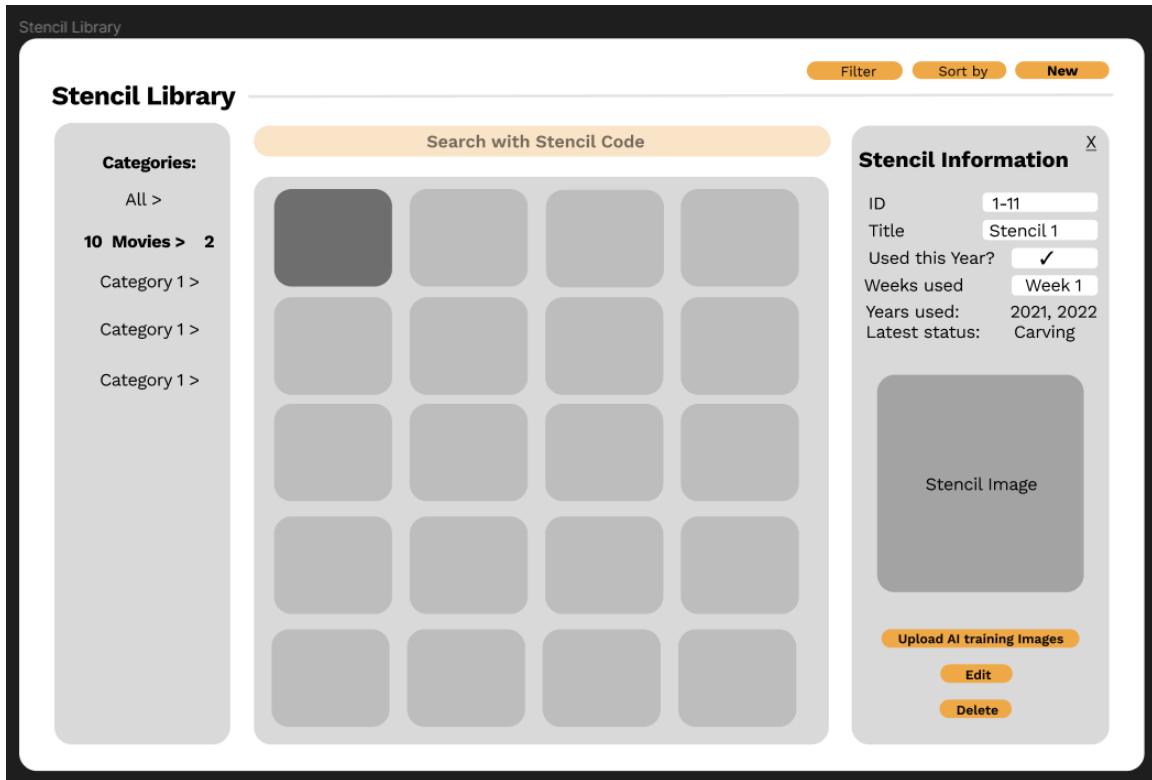


Figure 6. Admin Stencil Library

The image above is the current design for the admin's stencil library page. This screen is mostly the same as the Event Dashboard except that all stencils are displayed here (i.e., stencils are not specific to an event).

- **Visitor-side pages** - for stencil recognition

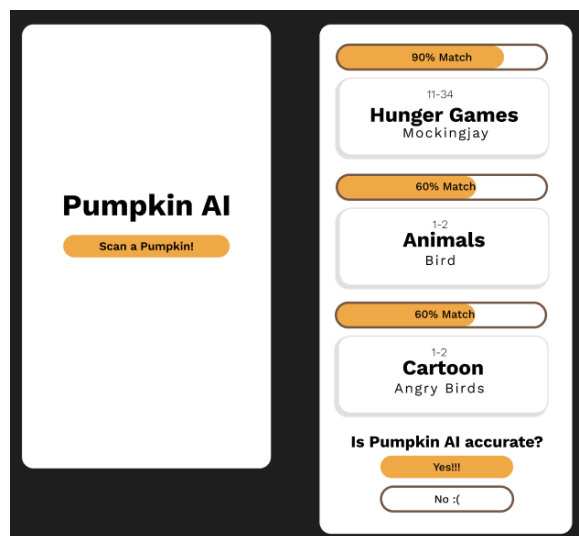


Figure 7. Stencil Recognition Screens

The image above is the current design for the stencil recognition part of the application. This is how the pages will be shown to the public (visitors), but admins or volunteers can also access these pages. The screens have the following details:

- A homepage to greet users and prompt them to scan a pumpkin using their cameras.
- After they have finished taking a picture of a pumpkin, the results will be shown (like on the second page). Then, they can say if the program was correct; data gathered here can be used to improve stencil recognition results in the future.

4.3.2 Functionality

The final project will be divided into three parts: stencil management for admins, stencil logging for volunteers, and stencil recognition for visitors.

- Admins will use the app during the event preparation. They will access the application through a web browser on their desktop. They will be able to add and edit stencil information as needed.
- Volunteers will use the app during the event preparation days for pumpkin tracing and carving. They will access the app using mobile browsers to update the status of their chosen stencils. If they don't have phones, the admins will update the pumpkin stencil status.
- Visitors will use the app during the Spirits of the Garden event. While strolling through the Jack-O-Lanterns, visitors can use their mobile browsers to access our site, where they can use their cameras to take a picture of a pumpkin, and they will be able to see information about which a jack-o-lantern utilizes a stencil.

4.3.3 Areas of Concern and Development

1. Primary concerns: Our main concern is the complexity of the UI for the admin website, as it requires significant effort to develop. In addition, we have doubts about the effectiveness of the AI camera feature, which may not be feasible to implement in the project.
2. Immediate plans to address the concern:
 - a. Conduct user testing sessions with volunteer admins to gather feedback on the UI and identify areas that need improvement.
 - b. Simplify the UI design to make it more user-friendly and intuitive.
 - c. Integrate user feedback and make necessary adjustments to the UI.
 - d. Explore alternative technologies to the AI camera, such as GPS location and mapping, and determine the feasibility of implementing them in the project.
 - e. Research GPS mapping technologies and integrate them into the application, enabling visitors to see their location in the event and information about the pumpkins in the surrounding area.
3. Questions for clients, TAs, and faculty advisers:
 - a. What are your thoughts on substituting the AI camera function with GPS location and mapping? Is this a feasible alternative?
 - b. What alternative technologies or approaches could we consider to simplify the admin UI and make it more user-friendly?
 - c. How can we ensure that the system meets the needs of all stakeholders, including admins, volunteers, and visitors?

- d. What features or functionalities should we prioritize to ensure the project's success?
- e. How can we effectively test and evaluate the system to ensure it meets all requirements and addresses user and client needs?

4.4 TECHNOLOGY CONSIDERATIONS

Our project will use:

- React with Next.js for frontend/backend development.
- MySQL will be used as a database management system.
- Jest and Selenium will be used for testing.

Strengths:

- React with Next.js is a popular and widely used framework for developing web applications with great documentation and community support.
- MySQL is a well-established and widely used database management system with a strong reputation for reliability and scalability.
- Jest and Selenium are popular testing frameworks that are easy to use and offer various testing options.

Weaknesses:

- Although React with Next.js has many benefits, it can be difficult for beginners to learn, especially for those without prior experience in JavaScript or web development.
- MySQL can be complex to set up and manage, requiring advanced knowledge of database administration.
- Jest and Selenium can be time-consuming to set up and configure for testing.

Trade-offs:

- Using React with Next.js can offer improved performance, but it requires a trade-off regarding increased complexity and potential difficulty in learning.
- MySQL is a reliable and scalable database management system that requires complexity and administration time trade-offs.
- Jest and Selenium are popular testing frameworks that offer many options, but they require configuration and setup time trade-offs.

Possible solutions and design alternatives:

- Consider using a simpler frontend framework or library, such as Vue.js or Angular, if the complexity of React with Next.js is too high.
- Consider using a cloud-based database management system, such as Amazon S3, Amazon RDS, or Google Cloud SQL, to simplify database management and reduce administration time.
- Consider using a simpler testing framework, such as Mocha or Chai, if Jest and Selenium are too complex to set up and configure.

4.5 DESIGN ANALYSIS

Our design works because it addresses the needs of different stakeholders involved in the Spirits of the Garden event. The web application provides separate interfaces for admins, volunteers, and visitors, ensuring a tailored experience for each user group. The Next.js framework ensures front and backend communication, while TensorFlow is planned for stencil recognition.

In addition to email communication, biweekly meetings have been crucial thus far in fostering collaboration and facilitating progress. These meetings allow the team to review designs and brainstorm effective solutions to solve the problem. They serve as a platform for clarifying expectations, discussing design plans, and providing feedback to workflows.

We intend to maintain this collaborative approach to ensure a successful outcome as we continue the project. We will persist with email communication to keep everyone informed and address any concerns promptly. Biweekly meetings will remain a key component of our project management, enabling the team to stay aligned, assess progress, and make informed decisions.

4.6 DESIGN PLAN

The proposed design for 4.3 consists of three main modules—stencil management, updates, and recognition. Our client reiterated that the stencil recognition part of our requirements was the lowest priority. As a result, the team has planned on designing and implementing stencil management and updates first. It is crucial for the team to successfully develop and test these parts of the application so that our project can be proved useful to our client.

Then, the team will dedicate its efforts to implementing the stencil recognition part of the application. The stencil recognition AI will not be as accurate as it needs to be due to the limited amount of data (stencil images) to train the AI. However, the team will create a good baseline for the AI where admin users can just upload new images for AI training into the system as they get more pictures in the future.

5 Testing

5.1 UNIT TESTING

Testing units will include each component in our app. These will include each pumpkin component on the volunteer side and each stencil entry on the admin side. Each page will also be tested as a component of our large application. Testing the image classification will also be important for classification accuracy. This part will be tested within TensorFlow with their testing libraries. The JS part of our application will be tested with Jest, ensuring that components behave as intended and display the correct information.

5.2 INTERFACE TESTING

Jest will also be used in interface testing. We will be testing each API endpoint for its existence, as well as intended responses to input. Each API endpoint is hosted in Next.JS, and it will be simple to test each endpoint as they are created will be simple. These tests will be run at the start of each deployment in development and production to ensure that our app works as intended.

5.3 INTEGRATION TESTING

In our design, two critical integration paths require thorough testing:

- Firstly, it is essential to ensure that the data between the main components, such as the volunteer and admin sides, is essential are consistent. This is critical because any inconsistency in data could lead to errors or incorrect information being displayed to the users, which could result in a negative user experience. Therefore, we will test the data consistency between components using integration testing tools such as Selenium, which can test the functionality and flow of the entire application.
- Secondly, AI stencil recognition is another critical integration path that must be tested. We must ensure that the stencil recognition works effectively with our current data. This is crucial as any inaccuracies in the stencil recognition could lead to incorrect pumpkin information, resulting in a negative user experience. We will use TensorFlow's testing libraries to test the accuracy of the AI stencil recognition.

The tool that will be used for integration testing is Selenium. Selenium is a popular open-source tool widely used for integration testing of web applications. It supports various programming languages, including Java, and can be used for testing applications developed in various web development frameworks, including Next.js. Additionally, Selenium supports cross-browser testing, making it a suitable choice for testing web applications across different browsers.

5.4 SYSTEM TESTING

The system-level testing verifies the overall behavior and performance of a software application as a whole and typically involves a mix of unit, interface, and integration tests before the additional tests that verify the functionality of the individual components together. Our system-level testing strategy will be closely tied to the requirements and focus on validating the system's behavior against the specified functional and non-functional requirements. These will include the following requirements that will demand most, if not all, of the system to be operational:

1. Updating and maintaining the stencils by the admins.
2. Preparation for the upcoming event involves tracking, selecting, printing, and cutting the stencils in the current year
3. Searching for and changing the pumpkin status from the admin's and the volunteer's side
4. Viewing the update submission logs and approving the statuses from the admin's side

The tools used for system testing include React Testing Library, Jest, and Selenium. For these tests to be deemed successful, they will have to end with the correct result without taking too much time.

5.5 REGRESSION TESTING

Regardless of the changes made to the application in the current development cycle, it is crucial that the website stays accessible for the visitors and the admins both during the event and throughout the year. So, to ensure that the application meets the required specifications and is stable and reliable after each cycle, we will re-execute the test cases executed in the previous cycles and add any additional cases that cover the new functionalities. The execution of the regression tests will be automated through CI/CD pipeline on Gitlab. A runner image will be generated to process the tests on our team's virtual machine on Iowa State Network.

As part of the Agile development process, these tests will mainly prioritize satisfying the functional requirements that are most important for the application to run. These functionalities include the ability for volunteers to update pumpkin/stencil information, the admin's control over selecting, printing, and cutting capabilities of the stencils, and approving or denying the volunteer's status submission.

5.6 ACCEPTANCE TESTING

Our acceptance testing is done bi-weekly with our client, with additional tests if needed. Client acceptance is very important in this deliverable and, thus, a large decision-making tool in our process. Each week leading up to the first week of October 2023 will consist of the tests with the client. Beginning this week, we plan to put our app into production and deep test the process and workflow. A trial run of the event will occur. Our team will be available on-site to solve any problems that arise. Based on feedback, and issues that occur, our team will work around the clock to ensure that our app can perform as expected in the event for years to come.

5.7 SECURITY TESTING (IF APPLICABLE)

Security testing does not quite apply to our project as our application does not rely on user accounts.

5.8 RESULTS

At this point, we do not have any test results to report.

6 Implementation

The project will be implemented as shown in the Gantt chart in section 3.4. The team will focus first on creating the application's volunteer side, then the admin side, and finally, the visitor side or stencil recognition. The team expects to finish the implementation before the Spirits of the Garden event in October so that the application can be tested and further improved.

7 Professionalism

7.1 AREAS OF RESPONSIBILITY

I. Work Competence

The SE code of ethics addresses this area in items (3), (4), (5), (6), and (8), which are mainly focused on work ethic standards.

Work competence is utilizing technical or soft skills relevant to the product's development.

The two versions differ in this area because the NSPE version discusses “high quality” work while the SE code of ethics pertains to working on a product that meets the “highest professional standards possible.” In the SE code of ethics, there is a standard set for work quality, while in the NSPE version, there was no standard mentioned aside from personal competence.

II. Financial Responsibility

The SE code of ethics addresses this area in item (4), which is mainly about a software engineer's professional judgment. Item (4) is exemplified in financial responsibility because products with “reasonable costs” require a judgment with integrity.

An engineer with financial responsibility can quantify the price of a product or service based on the effort and costs required to create or maintain it.

The two versions differ in this area because the SE code of ethics does not mention anything main point regarding finance. It might be the case that judgment with integrity is what a software engineer needs to make good financial decisions.

III. Communication Honesty

The SE code of ethics addresses this area in items (1), (2), and (7), which are mainly about maintaining integrity in the relations a software engineer has with the public, the client, and colleagues.

Communication honesty is building trust and integrity in relationships with stakeholders.

The two versions differ in this area because the SE code of ethics mentions the “stakeholders” associated with a software engineer—the public, the client, and colleagues.

IV. Health, Safety, and Well-Being

The SE code of ethics addresses this area in items (1), (2), and (6), which are mainly about satisfying the needs of the public and the client.

Responsibility towards the health and safety of the public is preventing potential hazards or risks associated with direct or indirect consequences of one's work.

The two versions differ because the SE code of ethics implies that the health and safety of the stakeholders go hand in hand with the "public interest."

V. Property Ownership

The SE code of ethics addresses this area in item (7), which pertains to the profession's integrity. Respecting others' work is a form of showing integrity in the profession.

Responsibility towards property ownership is not crediting others' work as your own.

The two versions differ because the SE code of ethics summarizes all the practices involving integrity into item (6), whereas the NSPE version has a dedicated area for property ownership.

VI. Sustainability

The SE code of ethics addresses this area in item (1), which pertains to the public interest. Environmental concerns are encapsulated within the public's interest.

Responsibility towards sustainability is being mindful of one's work's environmental consequences and thus preventing further damage.

The two versions differ because the SE code of ethics assumes public interest as a responsibility towards the environment, while the NSPE version specifically mentions this area.

VII. Social Responsibility

The SE code of ethics correlates this professional responsibility with item (6). The SE code of ethics tells us that we should always advance in projects and work with integrity and reputation and always in the public's interest, much like the responsibility area that tells us to make products and services that benefit society and communities.

The SE code of ethics differs by always focusing on the client or the public. The NSPE is mainly focused on the profession without regard to the public. Here, they are worried about giving the profession a tainted name, whereas the SE code tells us to do the right thing for the people.

7.2 PROJECT SPECIFIC PROFESSIONAL RESPONSIBILITY AREAS

I. Work Competence

This area applies to our project's professional context because our skill competence directly influences the quality of our output. The type of work we will exhibit while accomplishing the web application project will stay within the range of the team's software development skills.

The team performance in this area is high. The team must use technologies that are within the member's technical competence. For example, the project would not be successful if the team does not use a framework that the members can (or learn to) use and implement.

II. Financial Responsibility

This area does not apply too much to the team. Our project does not require us to incur any cost for implementation aside from setting up the server.

The team performance in this area is low because we are not given any budget as our project does not require us to buy anything big or physical where we need to exhibit financial responsibility.

III. Communication Honesty

This area applies to our project's professional context because the team has to be open to communicating work expectations and difficulties to other team members and stakeholders. Honesty in communication allows the team and stakeholders to understand what software requirements can be (or makes sense to) implement.

The team performance in this area is high. The team must communicate to each other any difficulties or misunderstandings so that assistance can be provided. This also helps the team to stay grounded and understand conflicts so that they can be better resolved.

IV. Health, Safety, and Well-Being

This area applies to our project's professional context because the team needs to ensure that our app is safe for public use on the internet.

The team performance in this area is medium. We need to be mindful of the data we will get from our users and not violate user privacy rights.

V. Property Ownership

This area applies to our project's professional context because our project involves handling stencils that were from the internet. Our team has to ensure that we will not show any of these stencils to the public because of possible copyright issues.

The team performance in this area is medium. This is because we only need to be careful about what information we show on our app's public side.

VI. Sustainability

This area does not apply too much to the team's project because there is no direct environmental impact we need to consider, as this is only a software-based project.

The team performance in this area is low. We do not need to account for any direct impact on environmental and natural resources.

VII. Social Responsibility

This area applies to our project's professional context because the team should be mindful of how the project will benefit its users. This software project focuses on improving the efficiency of the stencil-tracking process, so it becomes essential to us that we fulfill our users' needs.

The team performance in this area is high. This is because the design and features of our project app are stirred by ensuring that the final implementation of our app adheres to the requirements set by our client. The team must also work responsibly to have a final product that benefits the stakeholders.

7.3 MOST APPLICABLE PROFESSIONAL RESPONSIBILITY AREA

Social Responsibility

This area applies strongly to our project's professional context because the team must be mindful of how the project will benefit its users. This software project focuses on improving the efficiency of the stencil-tracking process, so it becomes essential to us that we fulfill our users' needs. If we fail our social responsibility by not benefiting our users, a massive roadblock could be put in the way of the event. Without improving efficiency, thousands of hours could be wasted, volunteers could become discouraged, and the event could shut down.

Our application will simplify the tracking process by bringing in the help of regular volunteers to lessen the load and burden of tracking. The lead volunteer will only be responsible for doing a final check of each pumpkin.

Our application can save lead volunteers time and energy on boring, repetitive tasks. This saved time could be reinvested into the event to improve the overall quality, giving the visitors an even better experience at the Spirits of the Gardens event.

8 Closing Material

8.1 DISCUSSION

The main objective of the project, as proposed by our client Nathan at the start of the semester, was to design an application that the organizers can use to make the tracking of the stencils and pumpkins less exhausting. As we proceeded through the design process, we not only met the requirements set by the client but also came up with more features to help manage the Spirits in the Gardens event. We extended the app from just admins to volunteers using it to submit their carving statuses to make things even easier for the staff, as described in Chapter 3 of this document. We also enriched the admin side of the application to manage the existing stencils and the selection, printing, and cutting of the stencils, all done in preparation for the event throughout the year.

The other aspect of the application that the client wants as an additional feature is Stencil Recognition from pumpkin images for admins and stencil images for volunteers using an AI model. Although we are still researching our options for the model, we considered switching to QR code as a backup plan if the accuracy of the AI model ends up being suboptimal.

8.2 CONCLUSION

For the first semester, our requirements were met, and we were able to get started on the implementation of the volunteers' side of the application. We successfully set up the database and the backend and containerized it with Docker on our Virtual Machine on the Iowa State network. Moving forward, we need to work on the admin side and the stencil recognition module of the application.

Completed Work Summary

- Volunteer side working prototype
 - Volunteer Screens
 - Stencil Search
 - Status Update
 - Database Implementation
 - Containerization with Docker
- Project Design
 - Requirements
 - User needs
 - Risk
- Schedule
 - Milestones
 - Gantt Chart
 - Testing Plan

One issue we've faced in this project has been the lack of well-defined design details for implementation towards the beginning of the semester. Everyone had different ideas about the project's appearance, and we needed to reach a consensus. However, after several team meetings

with the client and our advisor, we developed our designs and are now ready to implement them accordingly.

8.3 REFERENCES

"ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes," in *ISO/IEC/IEEE 12207:2017(E) First edition 2017-11* , vol., no., pp.1-157, 15 Nov. 2017, doi: 10.1109/IEEESTD.2017.8100771.

"ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," in *ISO/IEC/IEEE 29148:2018(E)* , vol., no., pp.1-104, 30 Nov. 2018, doi: 10.1109/IEEESTD.2018.8559686.

"ISO/IEC/IEEE International Standard - Software and systems engineering - Software testing -- Part 2: Test processes," in *ISO/IEC/IEEE 29119-2:2021(E)* , vol., no., pp.1-64, 28 Oct. 2021, doi: 10.1109/IEEESTD.2021.9591508.

W3C. (n.d.). Retrieved April 23, 2023, from <https://www.w3.org/standards/>

8.4 APPENDICES

Our work in the class throughout the semester can be found through the following Figma links:

Volunteer side screens:

<https://www.figma.com/file/ESoXYzD4mev57sEgmc2xKQ/Volunteer-Logging?t=5SKHyyN1FuLUQS8u-1>

Admin side screens:

<https://www.figma.com/file/jZgLhc1r33jZkQvs7WeJfk/Admin-side?t=5SKHyyN1FuLUQS8u-1>

Discover, Empathize, Research:

<https://www.figma.com/file/3DDZIRJ3VAZJ6FAzhqsbIk/Discover-%7C-Empathize-%7C-Research-%5BTeam-01%5D?t=5SKHyyN1FuLUQS8u-1>

8.4.1 Team Contract

Team Name

sddec23-01

Team Members

- 1) Kyle Goben
- 2) Phuoc (Johnny) Nguyen
- 3) Kiara Sta Maria
- 4) Omar Muhammetkulyyev

Team Procedure

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
 - a. Tuesdays after class face-to-face
 - i. Alternating between client meetings and regular meetings
 - b. Our advisor meeting is every other Wednesday at 1 pm
 - c. We will also meet after both our advisor and client meetings for a brief time
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
 - a. Discord
 - b. Email for client and advisor communication
 - c. Face-to-face for most meetings
 - d. Online discord meetings when needed
3. Decision-making policy (e.g., consensus, majority vote):
 - a. *Establish Clear Goals and Objectives.* Before making any decisions, the team should establish clear goals and objectives for the project. This will help guide decision-making and ensure that decisions align with the project goals.
 - b. *Use Data-Driven Approaches.* The team should use data-driven approaches to support their decision-making. This includes gathering and analyzing relevant data, such as client/advisor feedback and project progress, to inform decisions.
 - c. *Encourage Open Communication.* The team should encourage open communication and feedback from all members. Each team member should have an opportunity to share their ideas and perspectives, and all team members should be encouraged to ask questions and provide constructive feedback.
 - d. *Implement Consensus Decision-Making.* Use a consensus decision-making process when making decisions that impact the entire team. This involves agreeing that everyone can support rather than a simple majority rule.
 - e. *Consider the Risks.* Before making any decision, the team should consider the potential risks and benefits. The team should carefully weigh the risks and benefits of each decision and determine if the potential benefits outweigh the potential risks.
 - f. *Document Decision.* Document all decisions made by the team, including the reasons behind them. This will help ensure that decisions are transparent and can be reviewed and revised.
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
 - a. The team will save meeting notes in a Google doc in our drive folder.

- b. The team website will update all completed documents, notes, and work weekly.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
 - a. All team members will attend meetings on time to the best of their ability
 - b. If an absence occurs, it should be communicated to the team with notice
 - c. All team members will voice their opinions and contribute in all meetings with the team, advisor, and client.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
 - a. Team members should plan their work carefully and prioritize tasks based on importance and deadline.
3. Expected level of communication with other team members:
 - a. Team members should communicate regularly to ensure everyone is aware of project status, upcoming deadlines, and any issues or obstacles that may impact the project timeline.
 - b. Suppose a team member encounters an issue or obstacle that may impact the project timeline. In that case, they should proactively address it and communicate it to the team.
4. Expected level of commitment to team decisions and tasks:
 - a. Team members should take ownership of their assigned tasks and be accountable for their work. This includes ensuring work is completed on time and to the best of their abilities.
 - b. Team members should collaborate and support their colleagues as needed. This may include helping to troubleshoot issues, providing feedback and suggestions, and working together to overcome obstacles.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
 - a. Kyle: Team Lead, Client and Advisor Interaction, Project Organization
 - b. Kiara Sta Maria - Individual Component Design
 - c. Phuoc (Johnny) Nguyen - Documentation Lead
 - d. Omar Muhammetkulyyev - Testing Lead
2. Strategies for supporting and guiding the work of all team members:
 - a. Provide clear instructions and expectations for the project to ensure everyone is on the same page. This includes clearly defining each team member's roles and responsibilities, project goals and objectives, and project scope.
 - b. Schedule regular check-ins and meetings to discuss project progress, upcoming deadlines, and any issues or obstacles that may arise.
 - c. Encourage collaboration and communication among team members by providing opportunities for team members to work together and share ideas.
 - d. Provide team members with the resources and support they need to complete their work successfully.
3. Strategies for recognizing the contributions of all team members:

- a. Recognize and reward team members for their hard work and successes throughout the project through verbal recognition or team celebrations.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
 - a. Kyle:
 - i. Front and backend dev experience
 - ii. React/docker experience
 - iii. Ability to learn and adapt on the fly to new situations and challenges
 - iv. Excellent communication skills to communicate with advisor and client
 - b. Omar:
 - i. Back-end experience with NodeJS
 - ii. Familiar with both MySQL and MongoDB
 - iii. Moderate algorithmic background
 - c. Johnny:
 - i. Frontend and backend experience
 - ii. Familiar with both SQL and No-SQL Database
 - iii. Ability to work under high-pressure
 - d. Kiara:
 - i. Frontend experience with HTML and JavaScript
 - ii. Backend experience with Java and SQL
 - iii. Ability to work with component design and documentation
2. Strategies for encouraging and supporting contributions and ideas from all team members:
 - a. Create a safe and supportive environment that encourages open communication, active listening, and constructive feedback.
 - b. Encourage all team members to actively participate in team meetings and discussions.
 - c. Set clear expectations for team members regarding their roles and responsibilities, and communicate the importance of each member's contributions to the project's success.
 - d. Provide opportunities for team members to collaborate and work together on tasks and projects.
 - e. Celebrate successes and accomplishments throughout the project, and recognize the contributions of all team members.
3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment obstructs their opportunity or ability to contribute?)
 - a. Schedule regular check-ins and meetings to discuss project progress, upcoming deadlines, and any issues or obstacles that may arise.
 - b. Encourage clear and open communication among team members.
 - c. Develop a conflict resolution process to identify the issue, gather information from all parties involved, and collaboratively work towards a resolution.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
 - a. Be nice to each other and uphold the team's integrity

- b. Deliver a decent design document at the end of the term that satisfies both the client and our advisor
 - c. Get experience with big cloud providers like AWS or GCP
 - d. Meet deadlines for team assignments with ample time
2. Strategies for planning and assigning individual and teamwork:
 - a. Clearly define each team member's roles and responsibilities.
 - b. Assess each team member's skills and abilities to determine the best way to assign tasks.
 - c. Encourage collaboration among team members by assigning tasks that require multiple team members to work together.
 - d. Establish clear deadlines for each task and ensure everyone understands the importance of meeting these deadlines.
 - e. Monitor project progress regularly and adjust assignments as necessary.
 3. Strategies for keeping on task:
 - a. Identify and prioritize tasks based on their importance and urgency.
 - b. Develop a detailed schedule that outlines when tasks need to be completed.
 - c. Schedule regular check-ins with team members to discuss progress and address any issues or concerns that may arise.
 - d. Minimize distractions by establishing an effective working environment.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
 - a. Identify the infraction and determine the severity of the issue.
 - b. Discuss the infraction with the team member(s) involved.
 - c. Take action to address the issue.
 - d. Follow up with the team member(s) to ensure the issue is resolved.
2. What will your team do if the infractions continue?
 - a. If the infraction is particularly serious or is not resolved through discussion and action, it may be necessary to seek additional support. This could involve consulting with the instructor, advisor, or other stakeholders.

- 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) Omar Muhammetkulyyev | DATE 02/19/2023 |
| 2) Phuoc (Johnny) Nguyen | DATE 02/19/2023 |
| 3) Kyle Goben | DATE 02/19/2023 |
| 4) Kiara Sta. Maria | DATE 02/19/2023 |