

“Cy” Sophus

DESIGN DOCUMENT

SDDEC20-04

Client - Doug Jacobson

Samuel Christianson

Morgan Funk

Sean Gordon

Daniel Laracuenta

Aaron Lawrence

William Maston

sddec20-04@iastate.edu

<https://sddec20-04.sd.ece.iastate.edu>

Executive Summary

Development Standards & Practices Used

- Linux Development
- Unit Circle and Radians

Summary of Requirements

- Build a system that allows for user interaction with the table
- Build a system that can send live position drawing data
- Develop a multitude of different ways for interaction

Applicable Courses from Iowa State University Curriculum

- CPRE
 - 308: Operating Systems: Principles and Practices
 - 288: Embedded Systems I: Introduction
 - 281: Digital Logic
 - 488: Embedded Systems Design
- COM S
 - 309: Software Development Practices
- EE
 - 201: Electric Circuits
 - 230: Electronic Circuits and Systems

New Skills/Knowledge acquired that was not taught in courses

Table of Contents

1. Introduction	4
1.1 Acknowledgement	4
1.2 Problem and Project Statement	4
1.3 Operational Environment	4
1.4 Requirements	4
1.5 Intended Users and Uses	4
1.6 Assumptions and Limitations	5
1.7 Expected End Product and Deliverables	5
2. Specifications and Analysis	6
2.1 Proposed Approach	6
2.2 Design Analysis	6
2.3 Development Process	6-7
2.4 Conceptual Sketch	8
3. Statement of Work	9
3.1 Previous Work and Literature	9
3.2 Technology Considerations	9
3.3 Task Decomposition	9
3.4 Possible Risks and Risk Management	9-10
3.5 Project Proposed Milestones and Evaluation Criteria	10
3.6 Project Tracking Procedures	10
3.7 Expected results and Validation	10
4. Project Timeline, Estimated Resources and Challenges	11
4.1 Project Timeline	11-12
4.2 Feasibility Assessment	13
4.3 Personnel Effort Requirements	13
4.4 Other Resource Requirements	14
4.5 Financial Requirements	14
5. Testing and Implementation	15
5.1 Interface Specifications	15
5.2 Hardware and Software	15
5.3 Functional Testing	15-16

5.4 Non-Functional Testing	16
5.5 Process	16
5.6 Results	16
6. Closing Material	18
6.1 Conclusion	18
6.2 References	18
6.3 Appendices	18

1 Introduction

1.1 Acknowledgement

Our team would like to acknowledge our client Doug Jacobson for supplying us with the information needed to complete this project, as well as a Sisyphus table for testing.

1.2 Problem and Project Statement

Artist and engineer Bruce Shapiro has created the Sisyphus kinetic art table as a medium for artistic expression. Unfortunately, the process for creating tracks for the marble to follow is a fairly complicated process, with difficulty increasing exponentially the more intricate the design. This hurdle also limits the creativity of the artists using it, as much time must be spent planning a route in a polar coordinate file rather than spent creating new designs.

Our team was tasked with developing an interface for users to more easily use the Sisyphus table. The interface allows increased freedom of expression, eliminating much of the difficulty with creating tracks through applications for uploading images directly to the table, or simply for easier route planning.

1.3 Operational Environment

Our product exists in both physical and virtual space. Through physical components, we connect the table to our applications hosted on the internet, allowing ranged control of the device. Our interfaces are purely software so as to be displayed online for ease of access.

1.4 Requirements

The following are the requirements for the project:

- Self-explanatory user interface.
- Draw different kinds of shapes depending on how the user interacts with the table.
- Image scanning system.

1.4.1 Engineering Constraints and Non-functional Requirements

The following are the non functional requirements for the project:

- Have the table be able to support a series of images to be displayed as a playlist
- Fully functional companion website to go with table
- Integration with broadcast media (i.e Twitch)

Engineering Constraints

- We were unable to meet up in person after March of 2020 due to the ongoing pandemic
- Our team had to all work remotely and have a single team member be the main point of contact for interfacing with the table.

- Further time constraints were placed on each team member as our workloads from school and life increased in the recent months.

1.5 Intended Users and Uses

Our intended users are owners of a Sisyphus table, and artists with an interest in the Sisyphus table. In the event that a table owner streams video of their table online and allows input, the interface should also support those that are allowed input to that table.

Our single intended use is to more easily design tracks for the Sisyphus table.

1.6 Assumptions and Limitations

Assumptions:

- Users of the Sisyphus table have access to the internet.
- The Sisyphus table is a working prototype.
- The table operates in an area with heavy foot traffic.

Limitations:

- Web interfaces must be supported on all major browsers and some smartphones.
- The table requires extra hardware in order to connect to the IOT.
- Physical size of the finished table is determined by the implementation of the end product.
- Additional equipment needed is limited by the budget.

1.7 Expected End Product and Deliverables

Physical:

A setup for wireless communications between the table and our web interfaces.

Virtual:

An centralized web interface for easier track design including:

- An automated process for converting from cartesian coordinates to polar coordinates.
- A gui for creating designs.
- A system to connect the web interface to a particular table.
- An application for uploading images for conversion to a track.

2 Specifications and Analysis

2.1 Proposed Approach

- To ensure maximum efficiency in the Sisyphus table's performance, we are employing a well-documented approach to executing a plan to deliver the table on-time and on-budget.
- The initial phase of our project will comprise of conceptual engineering in which the budget and schedule is properly assorted to deliver the desired end-result.
- Documentation begins when the table is acquired.
- Development on the first phase of the project focuses on establishing a complete understanding on manipulating information outputted by the table.
- Communication between user and table will be tested using basic means designed by the team. The communication is declared enough when we can give multiple purposes to our method of user interaction with the table.
- A robustly tested and developed first approach to interacting with the table allows for further development for other means of interactions between table and user.

2.2 Design Analysis

Progress

- Met with client and developed a roadmap for the project
- Created prototype UI elements for Web and App development
- Created documentation and organization for the project to ensure development is on track and deliver a professional product

Successes

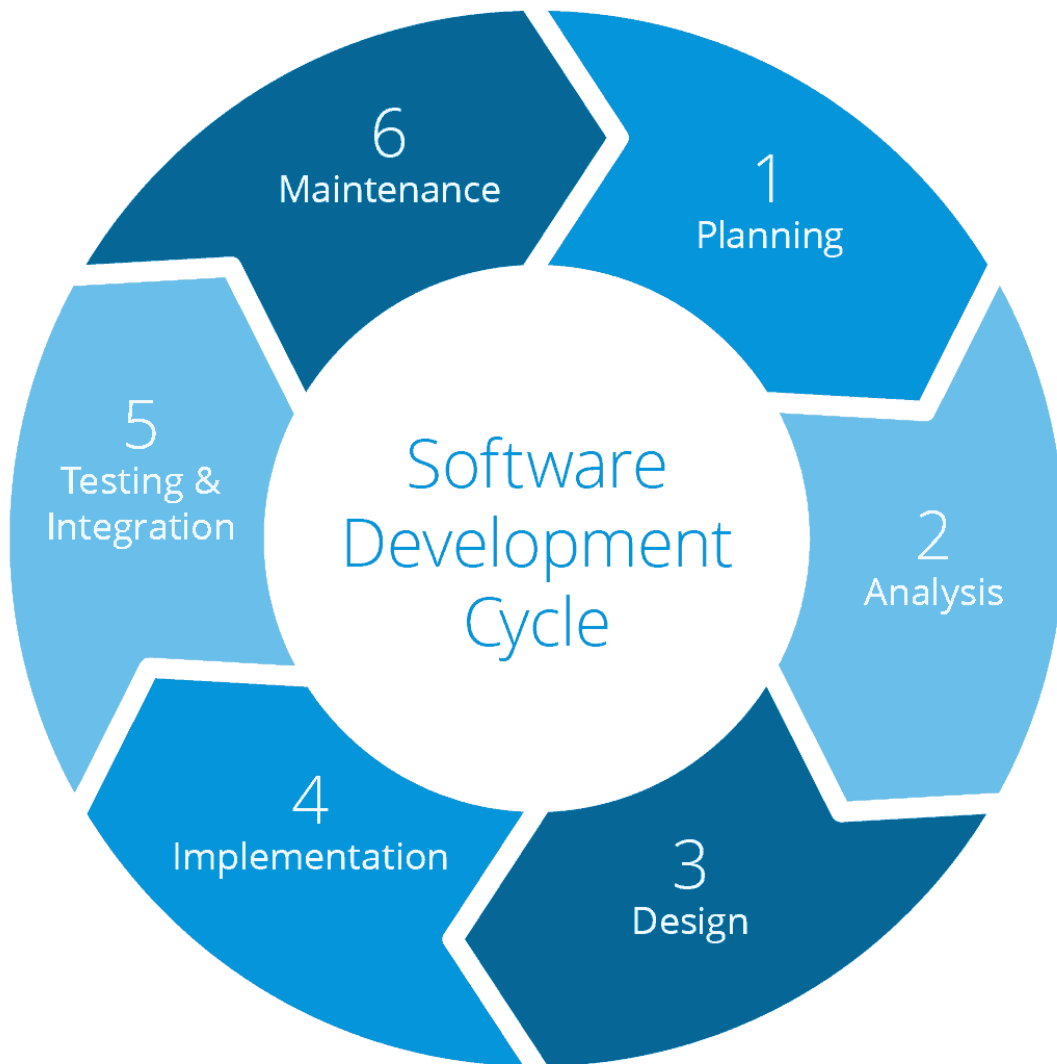
- Team communication is consistent and deadlines are apparent and respected
- Proposal is clear to our client and meets their needs
- Skills of team members cover project needs and development process is clear

Failures

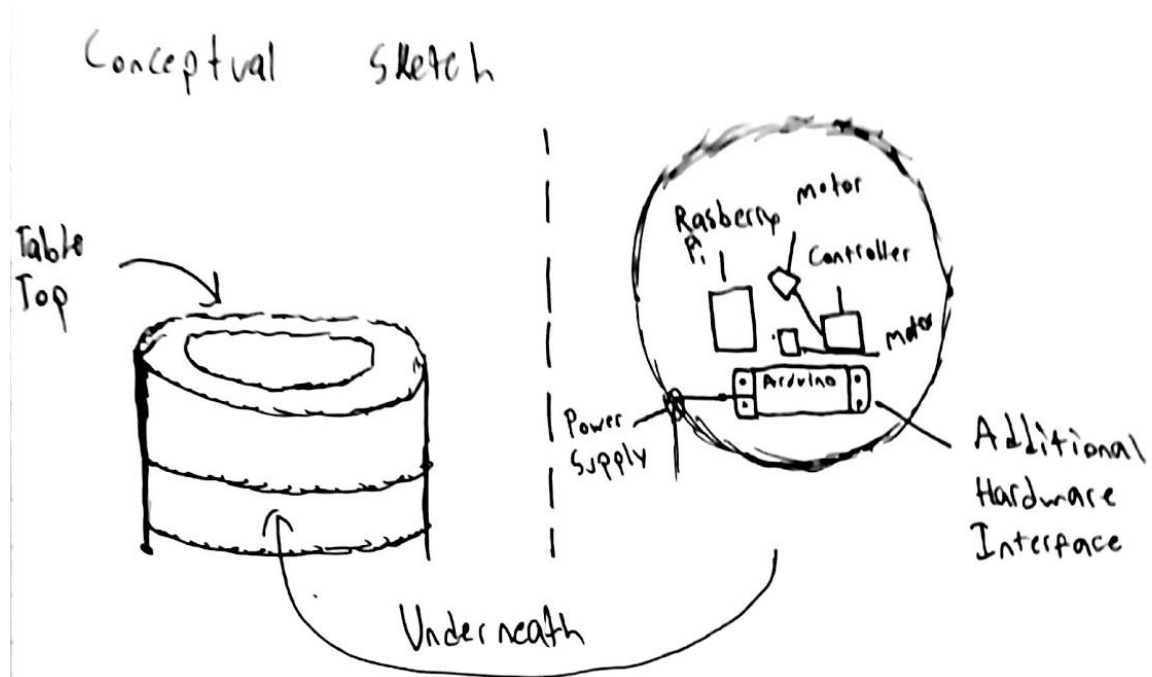
- Instances of meetings in person falling through
- Deliberation of concrete project design still prevalent
- Understanding and undertaking project roles and responsibilities

2.3 Development Process

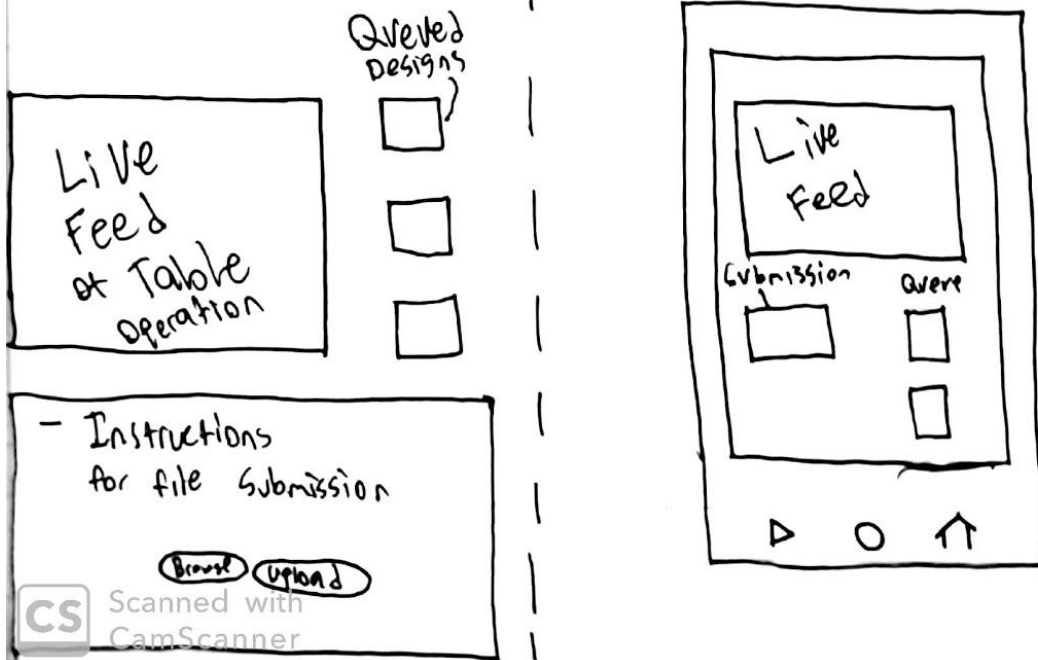
For this process, the Agile Development Cycle appears as the most prevalent choice as our development focuses on a heavy use of programming and hardware testing in order to adapt a given product. This product is then developed further with developments and testing in order for us to deliver a UI and User interactable version of this product.



2.4 Conceptual Sketch



Web and Application



3. Statement of Work

3.1 PREVIOUS WORK AND LITERATURE

The Sisyphus kinetic art table is a fairly unique art installation, Bruce Shapiro did a great job creating a unique and fascinating system. It is not wholly original however. Sand art is not a new phenomenon, but specifically robot made sand art is not completely unique either. In 2018 an independent engineer and artist, Ivan Miranda, created a [SAND DRAWING ROBOT](#). His design was a system that would draw using sand as a canvas by driving over sand and having electronically controlled brush turn down, creating a divet in the sand. The build worked effectively like a printer, going over the sand single inch wide row at a time.

Our system behaves fairly differently, with the Sisyphus being able to handle more fine detail, and can work in any location. The Sisyphus is also just more fascinating to watch work, but that subjective fact is hard to quantify.

3.2 TECHNOLOGY CONSIDERATIONS

This project does not require many new technologies on our end. Our main focus will be on the website itself. We will need to make sure that we pick a website hosting service that is reliable and can handle the amount of people that we expect to make use of our service, while also remaining affordable.

3.3 TASK DECOMPOSITION

- Research how to create files that the table can read
- Implement a UI for creating these files for a general user base
 - Allow users to create files using draw tools supported by our website
 - Allow users to upload pictures to our website
- Put up a website that can be accessed by the wider world
 - Make it so the website is not only accessible from the Iowa state network
- Create a server that can create files that receives requests from the website and sends them to the table
- Coordinate a place to display the table on Campus
 - Art installations, larger gathering areas, and library display rooms

3.4 POSSIBLE RISKS AND RISK MANAGEMENT

Our day to day lives have changed a fair bit since this report was last written, and it would be wrong of us to not at least acknowledge that. As coronavirus has spread throughout the world, we here at Iowa state have reacted swiftly to help with the effort to keep everyone safe. As of writing this (3/29/2020), all members of this group have started working remotely from home. All of our meetings are online and no in person events of any kind have been scheduled. We hope that by the start of the next semester this will change, and we can once again work unhindered, but for now we are all taking an abundance of caution to help slow the spread of covid-19.

As for other major risks and concerns we do have to worry about website security. Making sure that no one can shut down our website, or use our website as a vector to harm our users will be one of our top concerns. All information and data passed from user to server will need to be properly screened and secured.

3.5 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

1. Getting the table from Sisyphus Industries.
2. Initial testing of table, Formulation of our teams plan of attack, Setup of tools like drawing software and coding platforms that we all decide on.
3. User-table interaction up and running, Environment processing.
4. Environment processing and live data processing.
5. Ending table demo and setup in a public area.

3.6 PROJECT TRACKING PROCEDURES

For this project we will be making use of Trello to track project progress and facilitate work that needs to be done and who is working on that specific work. This program is great for setting milestones and letting us know who specifically is working on what milestones.

3.7 EXPECTED RESULTS AND VALIDATION

In the end of this project we intend to have a Sisyphus table that we can set up in a public area and have any type of passer-byer interact with and effect. We also intend to have a idle design processing where when the table is not in use it takes in sensor data to create meaningful patterns from, this will all be done live and we will be working alongside Sisyphus Industries to enable our table to take in live data and drawing patterns rather than only being able to process static programs.

We will confirm that our table is working at a high level by giving regular users access to the table beforehand and seeing how specifically they interact with the table and see if what we have created is at all user friendly and if not going back and changing some issues we see.

WBS NUMBER	TASK TITLE	May																															August																														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	Project Beginning																																																														
1.1	Class Beginning/Acclimation																																																														
1.2	Obtain Requirements																																																														
1.2.1	Refine Requirements																																																														
1.3	Research																																																														
1.4	Estimate Timeframe																																																														
2	Project Planning																																																														
2.1	Determine User Base																																																														
2.2	Define Scope																																																														
2.3	Communication Plan																																																														
2.4	Brainstorming																																																														
2.5	Decide on User Interfaces																																																														
3	Project Construction																																																														
3.1	Create UI Mockups																																																														
3.2.1	Begin Back-End Construction																																																														
3.2.2	Begin Front-End Construction																																																														
3.4	Connect Table to Internet																																																														
3.5	Connect Table to Interfaces																																																														
4	Project Monitoring																																																														
4.1.1	Debug Interfaces																																																														
4.1.2	Debug Table Connection																																																														
4.2	Effort and Cost Tracking																																																														
4.3	Project Performance																																																														

WBS NUMBER	TASK TITLE	September																															October																														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1	Project Beginning																																																														
1.1	Class Beginning/Acclimation																																																														
1.2	Obtain Requirements																																																														
1.2.1	Refine Requirements																																																														
1.3	Research																																																														
1.4	Estimate Timeframe																																																														
2	Project Planning																																																														
2.1	Determine User Base																																																														
2.2	Define Scope																																																														
2.3	Communication Plan																																																														
2.4	Brainstorming																																																														
2.5	Decide on User Interfaces																																																														
3	Project Construction																																																														
3.1	Create UI Mockups																																																														
3.2.1	Begin Back-End Construction																																																														
3.2.2	Begin Front-End Construction																																																														
3.4	Connect Table to Internet																																																														
3.5	Connect Table to Interfaces																																																														
4	Project Monitoring																																																														
4.1.1	Debug Interfaces																																																														
4.1.2	Debug Table Connection																																																														
4.2	Effort and Cost Tracking																																																														
4.3	Project Performance																																																														

WBS NUMBER	TASK TITLE	November																															December																														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1	Project Beginning																																																														
1.1	Class Beginning/Acclimation																																																														
1.2	Obtain Requirements																																																														
1.2.1	Refine Requirements																																																														
1.3	Research																																																														
1.4	Estimate Timeframe																																																														
2	Project Planning																																																														
2.1	Determine User Base																																																														
2.2	Define Scope																																																														
2.3	Communication Plan																																																														
2.4	Brainstorming																																																														
2.5	Decide on User Interfaces																																																														
3	Project Construction																																																														
3.1	Create UI Mockups																																																														
3.2.1	Begin Back-End Construction																																																														
3.2.2	Begin Front-End Construction																																																														
3.4	Connect Table to Internet																																																														
3.5	Connect Table to Interfaces																																																														
4	Project Monitoring																																																														
4.1.1	Debug Interfaces																																																														
4.1.2	Debug Table Connection																																																														
4.2	Effort and Cost Tracking																																																														
4.3	Project Performance																																																														

4.2 FEASIBILITY ASSESSMENT

This project is at risk for the typical issues that come with full stack development, and is otherwise a very safe venture. The most problematic point of the project will be working with pre-existing hardware to connect the table to our interfaces on the internet.

4.3 PERSONNEL EFFORT REQUIREMENTS

Task	Text reference/Explanation	Estimate of effort
Parsing an image in order for table to draw it	This will be the first milestone that we will be aiming for in order to get familiar with making the table draw objects.	This will be highly difficult as it will be the first target we will tackle and will probably help us learn what is necessary to draw using the table.
Parsing the image data of a user's silhouette using a camera	This will likely be a big part of the project as we have two components that compose of users interacting with the camera (with the help of a Raspberry Pi and Arduino Board) which will have a solid image detection system in place.	This task will be difficult as the image detection will not be a hassle for getting the general parts of a human's body, but getting a more detailed version will require more data points.
Table drawing the face of a user using a camera	This is going to be similar to the previous component, but more difficult as there will be more details to be captured.	Assuming that the last task is enough to get a strong hold on using the camera then it should come slightly easier but still hard since we will capture plenty of detail on a user's face.
Make user interface for our program on a tablet	This will be the core of our project and will be on the top priority to get working at its best as it will be the main way users will likely interact with the table.	This task will be of medium difficulty as we will be building an interface similar to the one that the creators of this table have available on phones. However, we will have our own version so the user's can have something drawn live from whatever is drawn on the tablet.

4.4 OTHER RESOURCE REQUIREMENTS

(Tentative) Parts/Resources Required:

- Raspberry Pi
- Arduino Board
- On-Campus 3D Printing/CNC Resources

4.5 FINANCIAL REQUIREMENTS

The Financial requirements for this project are fairly minimal.

The first and largest expenditure is the purchasing of the table itself. This purchase in particular however has been handled by Doug Jacobson through funding from the school, so we are directly tied to this purchase.

The second hardware purchase we will make is for a standard video recorder. One of our required features is a video stream of the table in motion. A proper video recorder could range anywhere from \$40 to \$60.

The last purchase we will make is for a subscription to a website hosting service. If we want to have our website application be usable by others, it will need to be run openly across the internet. The price of a service like this is around \$10 a month.

5. Testing and Implementation

5.1 INTERFACE SPECIFICATIONS

For hardware-related components regarding our project, we will be dealing with a Raspberry Pi, Arduino Board and Sisyphus table. We will make thorough tests of each component that we will work on in order to make sure that the next one can be done confidently. In terms of software interfacing, it will be running in Unix and Windows systems as we will want both types of tables available to our disposal for testing.

5.2 HARDWARE AND SOFTWARE

The majority of software method testing was done mainly using print testing, alongside the JUnit library. JUnit is a unit testing framework for the Java programming language, providing tools like annotations and assertions to compile tests for methods. These tests were done immediately after software completion to ensure the software worked as intended.

Web endpoint testing was done through postman. Postman is an API development platform, capable of sending full HTTP requests for testing endpoints and functions without requiring extensive code. We used postman during software development as well as afterward for final testing.

UI testing was done using Jest. Jest is a universal framework for creating, running, and structuring tests. It is very simple to set up thanks to its zero-config dedication. This was used to black-box test our javascript UI functions.

5.3 FUNCTIONAL TESTING

Testing over the course of the project was done using the 'Given, when, then' formula:

- (Given) some context.
- (When) some action is carried out.
- (Then) a particular set of consequences should be observed.

This formula was applied to each developed requirement in turn, ensuring the desired behavior had been achieved.

Below is a set of example tests:

Requirement	Test Scenario	Formula
Users can upload tracks to their table.	Create several sample tracks and verify that they are uploaded successfully.	Given a track, When the track is uploaded to the table, Then the file is successfully transferred.
Tracks can be created from images, preserving original features.	Take several images of varying contrast and makeup, and apply the image -> track algorithm. Ensure the result is adequately similar to the initial image.	Given an image, When the image -> track algorithm is applied, The resulting track is adequately similar to the initial image and preserves the main features.
Users can share custom tracks with others.	Take several sample tracks and attempt to upload them to the web applications. Ensure they are uploaded successfully and can be accessed from other accounts.	Given a track and an account, When the track is uploaded to the web applications from that account, The track is accessible by other accounts through the web applications.

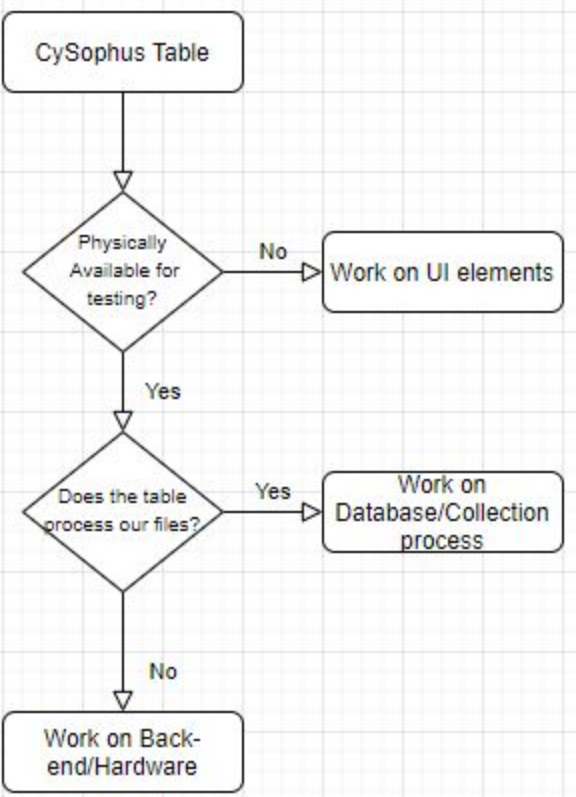
5.4 NON-FUNCTIONAL TESTING

Testing for performance, security, usability, compatibility

Requirement	Test Scenario	Formula
Table is easy to pick up and learn by any regular user.	The table and the corresponding device to control it is given to someone who has not used the table before and then given a task to complete.	Given a user with no prior experience When the user is given control of the table, They are able to intuitively pick up and use the table in various ways.
Table has measures in place to deter users from breaking into	Attempt to use the table and or its connecting app in various	Given the table and the app, When the app or table is

the tables software	unusual and malicious ways to attempt to break in.	tampered with in some way, Then the attack is properly halted by security measures.
Table instructions are easily understood for new users	Develop different ways to demonstrate the use of the table i.e infographic, website instructions and poll from users their comfortability or level of understanding	Given a website and new users. We are able to poll various different infographics that display the step by step process. Afterwards, the test group is polled for which is more desirable and understood properly.

5.5 PROCESS



Meetings/Communication:

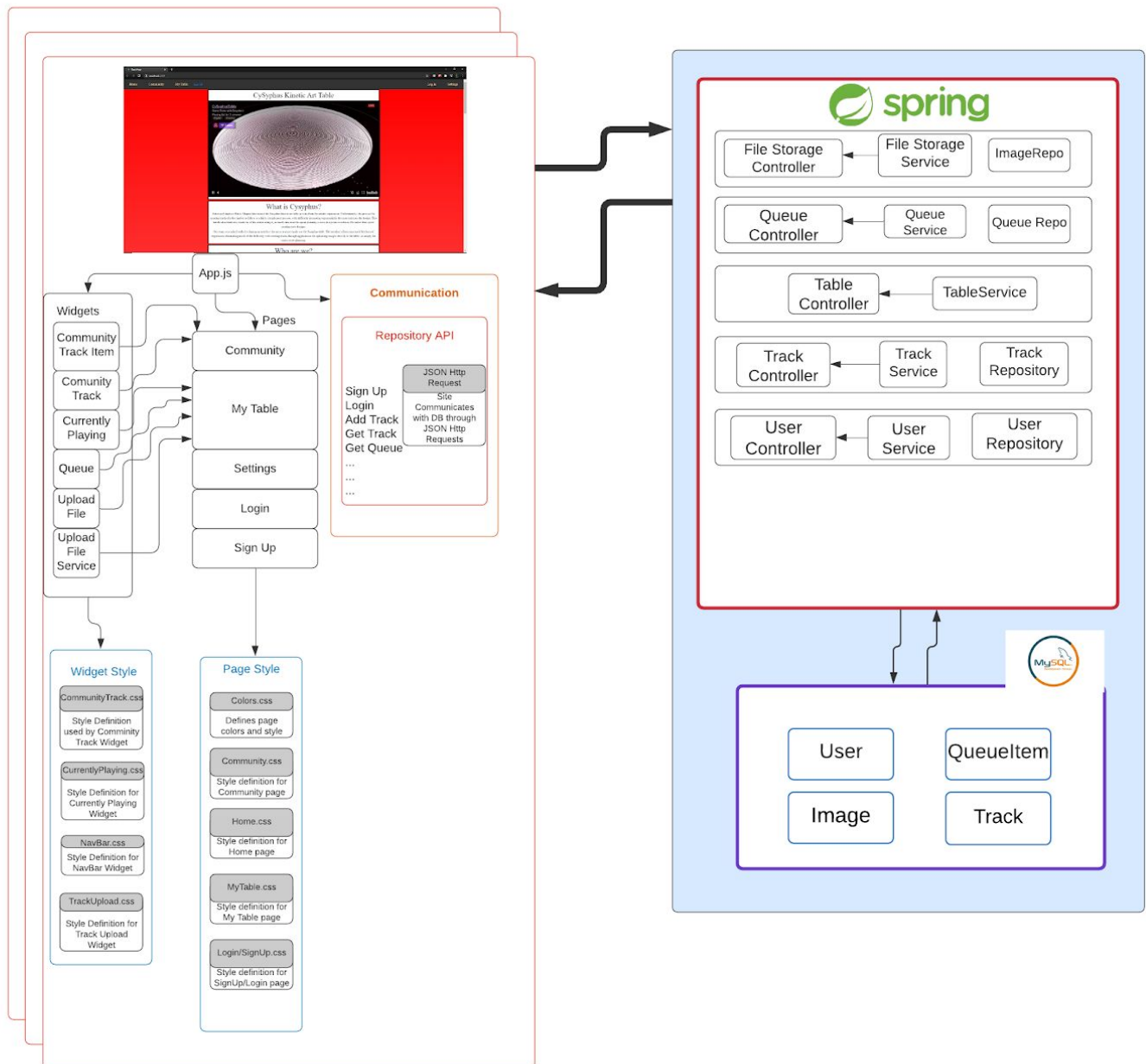
We are currently keeping in communication with our client and each other as this semester continues.

5.6 RESULTS

By the end of the semester we were able to produce a fully functional table, backend, and website data stream system. Users can post images to the website, the website sends the images to be converted at the backend, the backend sends the converted files to the table, and the table has a stream going on Twitch to observe the result of their input.

492 Block Diagram

Morgan Funk | November 15, 2020



Frontend Image



Backend File



Table Drawing Result



6. Closing Material

6.1 CONCLUSION

For hardware-related components regarding our project, we will be dealing with a Raspberry Pi, Arduino Board and Sisyphus table. We will make thorough tests of each component that we will work on in order to make sure that the next one can be done confidently. In terms of software interfacing, it will be running in Unix and Windows systems as we will want both types of tables available to our disposal for testing.

6.2 REFERENCES

This will likely be different than in the project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

- [1] Sisyphus Industries. "Composing 101 Episode 1: Making Your First Track", *Youtube*, Feb 24, 2020 [Video File]. Available: https://www.youtube.com/watch?v=2D9c1b_MX7I. [Accessed: Mar. 2, 2020].
- [2] Sisyphus Industries. Sisyphus Community Forums. <https://sisyphus-industries.com/community/> [Accessed: Mar. 2, 2020].
- [3] Sisyphus Industries. Sisyphus Setup. <https://sisyphus-industries.com/set-up/> [Accessed: Mar. 3, 2020].
- [4] Github, sisyphus. Dithermaster sisyphus github repository. <https://github.com/Dithermaster/sisyphus> [Accessed: Mar. 10, 2020].

6.3 APPENDICES

Sisyphus table metal side table user manual:

<https://drive.google.com/file/d/1ZqvjU8xLEoJYnkJ3Z6rAmOa7L9vDJRjW/view>

Sisyphus RGBW converter kit user manual:

<https://sisyphus-industries.zendesk.com/hc/en-us/articles/360039204691-Metal-Table-RGBW-converter-kit-manual>

APPENDIX 1 : OPERATION MANUAL

To enable the use of our project, information about the table and some code configuration will be needed to properly get information forwarded from the website.

The table:

- To start we will need to connect the table to a wireless network.
 - This is done by downloading the sisyphus table app and then connecting to the table's personal wireless network, once the app connects to the table the table can be configured to connect to the wireless network.
- We will need to access this network and get the ip address of the table
 - This will be done through a router's configuration software.
- In addition to the ip address, a port of 3002 will need to be forwarded to the above table ip.
 - This will allow for any postman packets sent from the website to reach and control the table.

The backend code:

- This table uses postman http requests to receive and send data to and from the web server. These requests connect directly through the ip and port number of the table. Each postman request needs to be configured to use the new updated ip address once the table is connected to a network.
 - This can be done by changing the baseUrl in the Table Service in the backend code

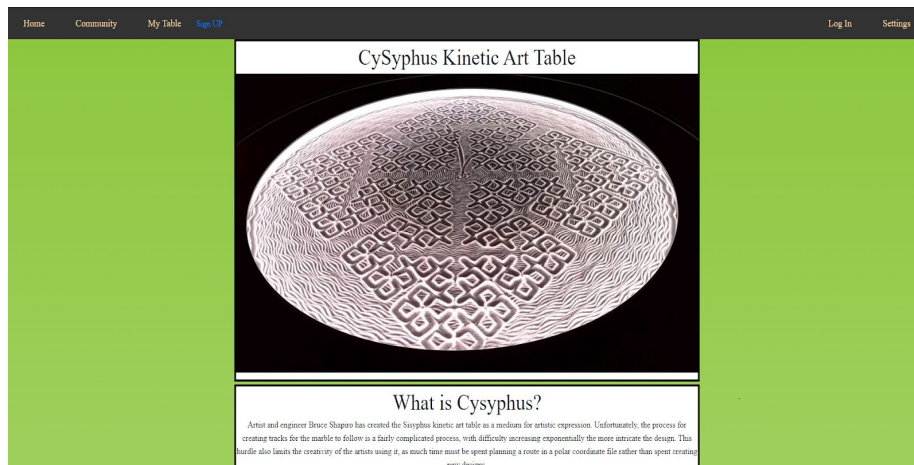
```

1 package com.sisyphusWeb.webService.controller;
2
3 import java.io.IOException;
4
5 @CrossOrigin
6 @RestController
7 public class FileController {
8
9     private static final Logger logger = LoggerFactory.getLogger(FileController.class);
10
11     @Autowired
12     private FileStorageService fileStorageService;
13
14     @Autowired
15     private TrackService trackService;
16
17     @PostMapping("/uploadFile")
18     public UploadFileResponse uploadFile(@RequestParam int userId, @RequestParam("file") MultipartFile file) {
19         // if(!userService.exists(name)) return new UploadFileResponse("This user does not exist", "", "", 0);
20
21         String fileName = fileStorageService.storeFile(file);
22
23         String fileDownloadUri = ServletUriComponentsBuilder.fromCurrentContextPath()
24             .path("/{fileName}")
25             .path(fileName)
26             .toUriString();
27
28         List<String> directories = fileStorageService.convertToTrack(fileName);
29         trackService.addTrack(directories, fileName, userId);
30
31         return new UploadFileResponse(fileName, fileDownloadUri,
32             file.getContentType(), file.getSize());
33     }
34
35     @PostMapping("/uploadMultipleFiles")
36     public List<UploadFileResponse> uploadMultipleFiles(@RequestParam int userId, @RequestParam("files") MultipartFile[] files) {
37         // if(!userService.exists(name)) return new ArrayList<UploadFileResponse>(Arrays.asList(new UploadFileResponse("This user does not exist", "", "", 0)));
38         return Arrays.asList(files)
39             .stream()
40             .map(file -> uploadFile(userId, file))
41             .collect(Collectors.toList());
42     }
43
44     @GetMapping("/downloadFile/{fileName:.+}")
45     public ResponseEntity<Resource> downloadFile(@PathVariable String fileName, HttpServletRequest request) {
46         // Load file as Resource
47         Resource resource = fileStorageService.loadFileAsResource(fileName);
48     }
49 }

```

Sisyphus table metal side table user manual: (Website)

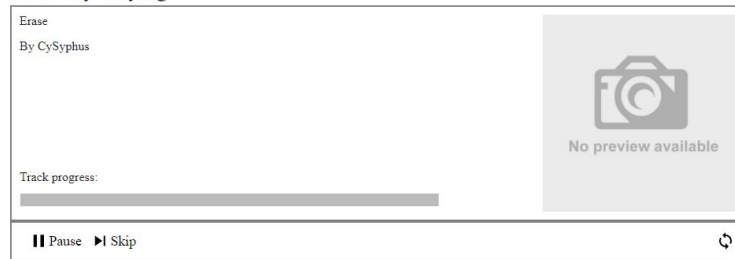
- To reach the website go to <http://localhost:8080>
 - We have developed a website platform for ease of use for users to interact with the table. This domain will lead to the home page which displays a current livestream of the table and information regarding the project and development team.



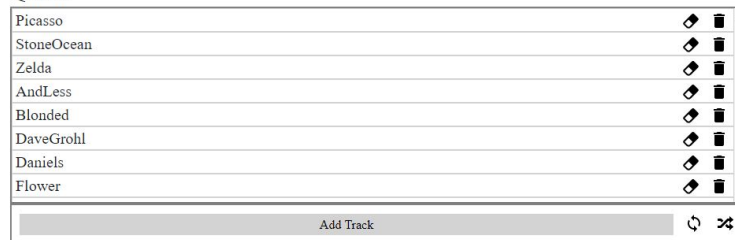
- The Navigation Bar at the top has links to several pages the User can Visit
 - Community: Users can view and download other users' tracks to use on their table.
 - My Table: Shows tracks the user has uploaded and is able to upload new tracks to add to the queue.
 - Sign Up: Sign up page allows users to make an account so that they may save and upload images to the My Table Page
 - Log in: Users may use this with the credentials they used for their account to access their My Table page.

Sam's Table

Currently Playing



Queue



APPENDIX 2 : ALTERNATIVE / OTHER INITIAL VERSIONS OF THE DESIGN

- Android App
 - We initially intended to build an android app as a counterpart to our web interface. This app would have identical functionality to the web interface, but would be more portable.
 - This was scrapped due to low demand and high time requirement.
- Draw Image to track
 - One of our early ideas was to include an application on our website that allowed users to use drawing tools to manually create and submit images to our table.
 - We had to abandon the idea after we ran into some trouble with PAINT api's, and decided to focus our efforts on more pressing matters.

APPENDIX 3 : OTHER CONSIDERATIONS

- We were tasked with working with the codebase given to us by the original creator. It was up to us to decipher and work around the constraints given by its creator.
- Utilizing Github and Communication programs we were able to test and move our project entirely online. It was surprising how much our project could adapt to online given the current circumstances.