

Cow Chips 4 Charity

DESIGN DOCUMENT

Group 8

Ken Johnson

Daniel Lev, Ben Meeder, Lotfi ben Othmane

Chloe Sabado - Front End

Clint Lamar - Back End

Helen Woldesenbet - Back End

Jake Liebman - Front End

Jared Schuckman - Front End

Swecha Ghimire - Front End / Coordinator

Xander Apponi - Back End/ CI/CD/ Scribe

sdmay21-08@iastate.edu

<https://sdmay21-08.sd.ece.iastate.edu/>

Revised: 10/1/20 Version 1

Executive Summary

Development Standards & Practices Used

- Agile Methodologies
- Daily Standups
- Biweekly Client Meetings

Summary of Requirements

- Update Administrator Screen
- Design animation for the Game Screen
- Improve Data Analytics
- Modernize User Interface

Applicable Courses from Iowa State University Curriculum

- COM S 363
- COM S 336
- COM S 319
- COM S 309

New Skills/Knowledge acquired that was not taught in courses

- VueJS
- Unity3D
- Blender

Table of Contents

1	Introduction	5
1.1	Acknowledgement	5
1.2	Problem and Project Statement	5
1.3	Operational Environment	5
1.4	Requirements	5
1.5	Intended Users and Uses	6
1.6	Assumptions and Limitations	6
1.7	Expected End Product and Deliverables	6
2	Project Plan	7
2.1	Task Decomposition	7
2.2	Risks And Risk Management/Mitigation	8
2.3	Project Proposed Milestones, Metrics, and Evaluation Criteria	8
2.4	Project Timeline/Schedule	9
2.5	Project Tracking Procedures	10
2.6	Personnel Effort Requirements	10
2.7	Other Resource Requirements	11
2.8	Financial Requirements	11
3	Design	11
3.1	Previous Work And Literature	11
3.2	Design Thinking	12
3.3	Proposed Design	12
3.4	Technology Considerations	12
3.5	Design Analysis	12
3.6	Development Process	12
3.7	Design Plan	13
4	Testing	13
4.1	Unit Testing	13
4.2	Interface Testing	13
4.3	Acceptance Testing	14
4.4	Results	14
5	Implementation	14

6 Closing Material	14
6.1 Conclusion	14
6.2 References	14
6.3 Appendices	14

List of figures

Figure 1: Project Timeline

Figure 2: Component Diagram

List of Tables

Table 1: Effort Requirements

1 Introduction

1.1 ACKNOWLEDGEMENT

We would like to thank Ken Johnson, Daniel Lev, and Ben Meeder for their help with this project. Because of their concise feedback, introductions, and guidance, we were able to get a great start with the design and workflow of this project. Our team also thanks Professor Lotfi Ben Othmane for volunteering his time to be the faculty advisor for this project.

1.2 PROBLEM AND PROJECT STATEMENT

The Boo Radley foundation is a non-profit organization created to promote research for diseases that are common to humans and animals. Being a non-profit organization, funding for the organization is acquired through donations and events. One of the most important events for funding is CowChips4Charity. CowChips4Charity is currently held as an in-person fundraising event, where participants select one of many squares in a pasture. If the cow roaming the field selects a participant's square, they win, and are given a prize. Any funds collected during this event contribute to The Boo Radley Foundation charity. Presently, while beloved as a state fair event, CowChips4Charity lacks the outreach it could have to acquire more funding for the charity. On top of this, the current need for drones (and other management equipment) ends up costing a sizable amount, lessening the total funds used for the charity.

To reach a wider audience and cut down on unnecessary expenditure, we seek to create a fully functional digital version of CowChips4Charity. To create this digital version, we plan on expanding the existing codebase with features such as an animated UI and accessible menus for a wider audience, as well as running a web service with virtually non-existent maintenance costs.

1.3 OPERATIONAL ENVIRONMENT

Our application will be run on the cloud, so there are no physical risks. While there are still possible outages, they will be minimal due to the use of a major company's servers. As for the client application, little processing power will be needed, as the game itself can easily be handled by modern smartphones. As well as limited graphical use, network usage is also minimal, limited to sending account and guess information. This is especially important, given the unpredictability of client connections.

1.4 REQUIREMENTS

- Use of Predefined Languages:
 - MongoDB (Database)
 - Vue.js (Visuals)
 - Node.js (Website)
- The Administrator Panel must be upgraded to include data analytics
- The UI of the game must be easy to understand and use
- Menial tasks should be more automated (winner selection, prize email sending)
- The game should be fully functional in a virtual environment
- Analyzed data should be highly customizable
- Network connections/usage should be brief
- Application visuals should follow the given template

1.5 INTENDED USERS AND USES

For use of this application, there are two core groups of users. The first group are the clients. Clients will access the application through the website, playing the game after creating an account, registering payment methods, and finally selecting an active game. The other core group is the administrators. The administrators consist of Boo Radley Foundation members, and as such will be able to create games, change game settings, and look at a comprehensive set of data analytics. By dividing these two groups we can separate the systems completely, into an administration website, and a client website, simplifying the development process.

1.6 ASSUMPTIONS AND LIMITATIONS

Assumptions:

- Users will have a smart phone with internet access
- Users will have a valid payment method
- Users will be residents of the United States (excluding California)
- Clients and administrators may not have extensive technical knowledge
- Creation and management of events will be handled by the administrators

Limitations:

- This application cannot be used outside of the United States (or in California) due to privacy laws

1.7 EXPECTED END PRODUCT AND DELIVERABLES

Due to use of an agile workflow, as well as a pre-existing codebase, design deliverables are not a core aspect of our project. Rather, our deliverables generally take the form of implementations of functional requirements.

Tech Stack Documentation (Due: 10/25/20)

Information regarding the technologies to be used on the various ends of the project will be detailed here. By this point in the project, all members will be familiar with the existing code, and will be able to select/justify use of further technologies. New technologies will particularly be used for animations on the client, data aggregation on the server, and card verifications on the server.

First Game Prototype (Due: 1/1/21)

The first prototype of the Game will include a fully functional (playable) game, as well as an initial implementation of the visual style requested by the client. The playable game will allow control features from the administrator panel, simplistic menus, and contain a rough look of the final game to be shortly tested.

First Administrator Panel Prototype (Due: 1/1/21)

The first iteration of the admin panel will include all required functions for administrators, with rough visuals to focus more on functional elements. Requirements, mainly revolving around data aggregation will be in place, with customization options in place for future testing.

Beta Testing (Due: 2/1/21)

Beta testing will begin shortly after the completion of the first prototypes. This beta testing will involve the use of a plethora of testers from a variety of backgrounds to test both usability and reliability. By recording the feedback gained from these testers, we will be able to fix any oversights as to the usability of our system to an average user. On top of this, we can get closer insight to the power needed by this application to handle users in the real world

Final Game Prototype (Due: 4/10/21)

The final prototype for the game will commence directly after the combined beta testing taking place until February. This final prototype will build on top of the previous prototype, using the data gathered from the beta to offer a more understandable and tested system. To achieve those objectives, minor focus will be placed on animation and design, while the core focus will be on testing, ensuring the product will be consistent upon completion of this project.

Final Administrator Panel Prototype

The final prototype of the administrator panel will also begin directly after the beta testing. While the core functionalities of data aggregation will have already been completed, based on the feedback from the beta, changes will be made. These changes include minor design alterations to be more understandable, as well as more aggregation options to offer better control to administrators.

2 Project Plan

2.1 TASK DECOMPOSITION

We will have 4 main subtasks each with their own requirements:

Design an CowChips Animation to be embedded into the website: This task can be further broken down into Creating the environment for the animation, Animating the environment to the specifications of the client, and Embedding the animation at the correct place in the website. This task is dependent on the UI sending a signal to start the animation and the Results screen to distribute rewards effectively.

Update the Administrator Panel for hierarchical privileges and increased data: This task can be broken down into Creating a User Hierarchy, implementing more usage statistics, and redesigning the look of the existing screen for these new features. This task is also dependent on the UI as well as the new Data Analytics framework.

Modernize the UI for Cleanness and Conciseness: This task can be broken down into updating the start screen UI, updating the admin panel UI and updating the result screen UI. This task is dependent on all of the existing elements as each new and old feature will need to be accessible to certain tiers of users.

Create a framework for Data Analytics: This task can be broken down into Gathering and Storing appropriate data, Analyzing the newly grabbed data, and Displaying the data to the appropriate users. This task is dependent on the UI, the Admin panel and the Results screen as all of these pages have worthwhile data to collect.

2.2 RISKS AND RISK MANAGEMENT/MITIGATION

Design an CowChips Animation to be embedded into the website: None of us have experience with animation. This presents a risk of meeting our deadlines, because we may have a larger learning curve than expected. Risk Factor: 0.4. We can buy pre-built models with animation rigging set up off sites for \$100-\$400 if we are unable to become competent in the required skills on time.

Update the Administrator Panel for hierarchical privileges and increased data: The only risk for this is the interdependencies of this with other parts of the project, such as the backend systems, and data analysis. We have competent programmers on our team who have built similar things before. Risk Factor: 0.1.

Modernize the UI for Cleanness and Conciseness: This task should be simple as the elements already exist, and just need to be updated. This will require competency in Vue.js framework, which none of have past experience with. Risk Factor: 0.3

Create a framework for Data Analytics: Similarly, to the admin panel, this task has a heavy reliance on the backend data systems to function properly. Also, there is not much test data to create visualizations from and perform analysis. Risk Factor: 0.5. Our risk mitigation plan is to generate our own synthetic test data to perform analysis on.

2.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

Based on the task decomposition above, we have proposed the following milestones, metrics and evaluation criterias

Design CowChips Animation to be embedded into the website

This milestone is going to be measured by the Usability metrics for effectiveness which will be calculated by completion rate. Meaning, how long a user will take to complete the task when navigating the animation that is embedded into the website. we will use 80% completion rate as our goal metrics. Which is Number of tasks completed divided by Total number of tasks undertaken.

Update the Administrator Panel for hierarchical privileges and increased data

For updating the Administrator Panel for hierarchical privileges and increasing data, we will be measuring the response time as metrics and evaluation. Since, this part of the project is the core backend system of the project metrics and evaluation will depend on overall efficiency and usability of the Administrator Panel. Which shouldn't be longer than 10 ms per request (This is a guess).

Modernize the UI for Cleanness and Conciseness

For modernizing the UI, we will be measuring this milestone by using Usability metrics for the time it takes to finish a task. Meaning, how long a user takes to navigate to a specific dashboard or link. This can be calculated by subtracting the End time from the start time. Finally, we'll average out the data based on how long each task is supposed to take. We will aim for an 80% completion rate.

Create a framework for Data Analytics:

For the Data Analytics, since our core functionalities are Gathering and Storing appropriate data, and Analyzing the newly grabbed data, and Displaying the data to the appropriate users. we will be using the search result success rate as our metrics and evaluations criteria. Meaning, we will look at the

usefulness and quality of our search result as it tells us the quality of the data we gathered. We'll use a 90% search result success rate as our metrics and evaluation. We will update it as we go into our next agile development process.

2.4 PROJECT TIMELINE/SCHEDULE

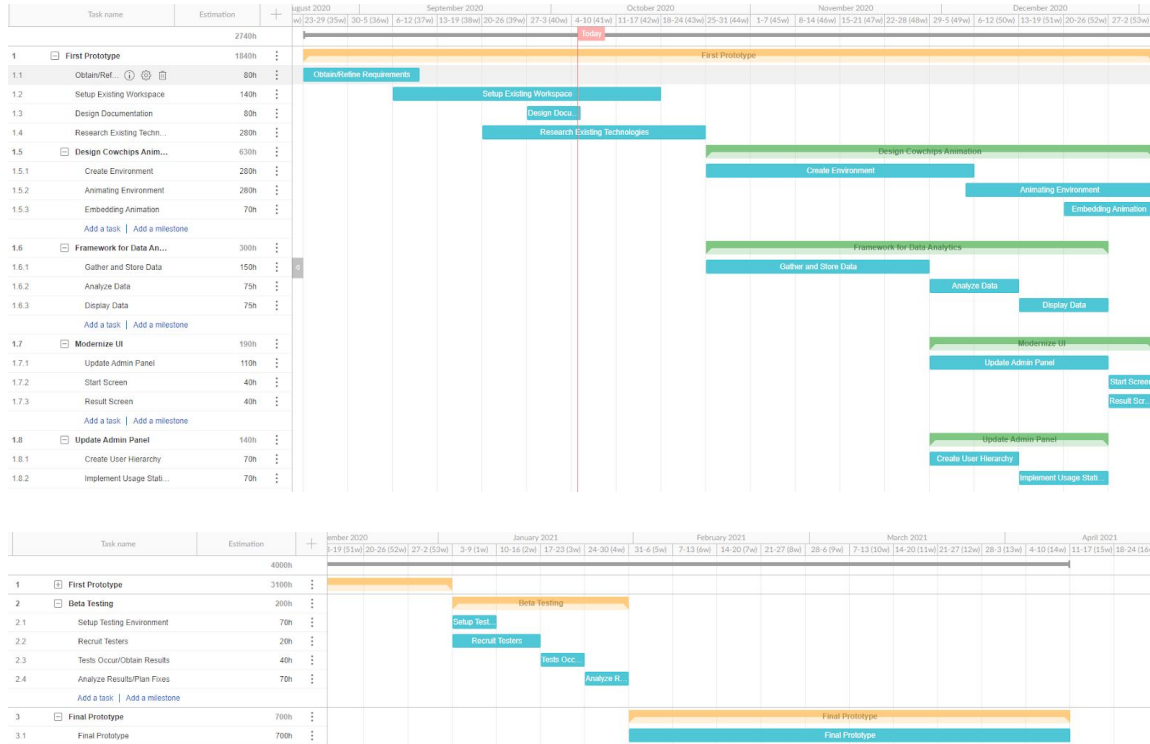


Figure 1: Project Timeline

2.5 PROJECT TRACKING PROCEDURES

Our group will be using Jira, Github and Miro to keep track of tasks. We will keep the repository in Github, we will keep track of stories and required tasks on Jira, and lastly, we will use Miro during team meetings as a group whiteboard style. Progress will be assessed upon completed stories and successful implementation of features in the repository.

2.6 PERSONNEL EFFORT REQUIREMENTS

Main Task	Sub tasks	Project Effort (person-hours)
Design Cowchips Animation	Create Environment	280
	Animating Environment	280
	Embedding Animation	70
Total		630

Update Admin Panel	Creating User Hierarchy	70
	Implement usage statistics	70
Total		140
Modernize UI	Update Start Screen	40
	Update Admin Panel	110
	Update Result Screen	40
Total		190
Framework for Data Analytics	Gather and Store Data	150
	Analyze data	75
	Display data	75
Total		300

Table 1: Effort Requirements

We have 13 weeks till 1/1/2021 which is our deadline for having our first game prototype complete. Assume each person works at least 10 hours per week for 13 weeks that is 130 hours per person, for 7 people that is 910 hours. We are also assuming during the 5 weeks of no school until the deadline each person works at least 20 hours per week or an additional 10 hours per person per week or 50 hours per person total. So total hours until deadline is $910 + 50 * 7 = 1260$. We figure that 50 percent of effort should be put into designing cow chips animation due to it being the most complex. 35 percent should be put into updating the admin panel and the framework for data analytics. And the last 15 percent should be spent on modernizing UI.

The design cow chips animation main task is divided into three subtasks create environment, animate environment, and embed animation. Creating the environment and animating the environment we expect to be of equal complexity, so we have allotted them both about 45% of the time allotted to designing the animation. And embedding the animation should be a smaller task so we allotted about 10% of the allotted design animation time.

The update admin panel main task is divided into two subtasks creating user hierarchy and implement usage statistics. Both are of about the same difficulty, so we allotted 50% of the time allotted to update admin panel for each.

The modernizing UI main task is divided into three subtasks start screen, admin panel and results screen. The start screen is one of the simpler screens so modernizing should take the least amount of time, so we are assuming about 20% of time that is allotted for modernizing UI. Result screen is also a fairly simple screen, so we also allotted about 20% of time that is allotted for modernizing the UI. And the admin panel is one of the more complex screens, so we allotted 60% of time to update admin panel screen.

The framework for data analytics main task is divided into three subtasks gather and store data, analyze data, and display data. We expect gathering and storing data to be the most complex requiring 50% of the allotted time for the framework for data analytics. And analyze data, and display data to be of equal complexity requiring 25% of the allotted time each.

2.7 OTHER RESOURCE REQUIREMENTS

We do not require external hardware or other resources to complete our project.

2.8 FINANCIAL REQUIREMENTS

The only financial requirements we may run into are for the animation. We need either a 3D artist to make the models or we need to purchase models from outside sources.

3 Design

3.1 Previous Work And Literature

Our Project is to be built on two previous years of senior design work. As such, there is a pre-existing website that will be built upon for this project. This website incorporates many of the necessary features to make the game 'CowChips4Charity' functional. For the frontend this includes a hosted website, login/verification, a CowChips board, and communication with the backend. The backend currently has a hosted website, administrator login/verification, database storage, and communication with the frontend. To make all of this functionality possible, the website currently uses technologies such as Javascript, Vue.js, Node.js, MongoDB, CoreUI, and Heroku. Combined, these technologies offer a robust modern system, allowing for a good looking (and effective) frontend, while also simplifying the backend communication and data storage. The largest downfall of these technologies is our groups lack of knowledge surrounding them. In order to prevent this negative, our group has designated a period of time to understand the current technology in use (see section 2.4).

Source Citation:

<https://www.cowchips4charity.com/>

3.2 Design Thinking

Our project, as a loosely constrained, highly client driven project, requires the use of at least two 'define' points. The first of these points is done before the first iterations of the project. Using data gathered from actual past users of this application, we are better able to define what these users need (such as a superior UI). With this information we can then ideate solutions to those needs. Key ideations thus far include the use of google analytics and power bi for data aggregation, or the embedding of Unreal Engine animations in the UI. In order to test these ideations, we plan to

finish the first iterations, and immediately beta test the application with real users. From here, with the results of the testing, we repeat the design thinking methods (mainly define, ideate, and prototype). We believe this layered approach will allow us to correct any disproven ideations, and result in a superior final project.

3.3 Proposed Design

Currently, our team has spent time to learn the existing technologies and frameworks that were used in the past for this project. The biggest aspect of this was learning vue.js, as the entire front-end is built on this. Also, we spent time learning mongodb, CoreUI, and other aspects.

One of the big challenges of this project regards the animation aspect. We have been in contact with a recent master graduate who is an expert in video game animation and design. Leveraging her experience and knowledge, we are working on a plan to make sure that the animation aspect of the project is done professionally and on time. Moving forward, we will be testing using Blender and Unreal Engine to create and embed the animations needed into our project.

Regarding the requirements of the project, our design will work well to satisfy all items listed in section 2. Since this project has already been in development for 2 years, we plan on leveraging the existing tools and technologies as much as possible, whilst working within our constraints to achieve the requirements. This means that the default design for most of the project is dependent on the frameworks that are already in place.

We have decided to change out the CI/CD tool from Travis CI to GitHub Actions, as it is cleaner and more fully functional. In addition, we have decided to move forward on the animation design with the tools previously mentioned.

For the entirety of the development and testing of this project, we will be following all the tenants laid out by the IEEE Software Engineering Code of Ethics. In particular, the foundation's goal is to help connect veterinary medicine to human sciences. Section 4.01 of the SE COE states: "Temperall technical judgements by the need to support and maintain human values." This is consistent with the goals of the foundation. Our team strives to find new value impact to help the foundation in its mission.

3.4 Technology Considerations

There are a lot of technologies that could be used so it was tough to decide what to use. We opted to use Vue.js, MongoDB and NodeJS as the three main languages and frameworks. A large part of this decision was that these were the languages the existing code base consisted of. The strengths of these tools are that they are very fast, modifiable and lightweight. We wanted to make sure our website could run on any machine to provide increased accessibility. The main weakness of these technologies is the team's lack of proficiency going in. Most of the team had little to no experience with these frameworks which makes it more difficult to develop with. However, as a team we decided it was well worth the learning curve to be able to continue using these frameworks. If we wanted to do things differently than the existing code base, we discussed other

javascript frameworks such as React. However, since we also had limited React experience we decided it was safer and smarter to just use the existing frameworks. One last technological consideration was deciding what tools to use for the animation. Our options were to use a 3d modelling software such as Blender or Maya, use a game engine such as Unity or Unreal, or use Adobe AfterEffects or Acrobat. We decided to use a combination of 3d Modeling software and a game engine as we liked the freedom and scalability of that combination.

3.5 Design Analysis

We are currently in development of our design from 3.3. As we work to implement the design such as the animation component for the game and the data to be gathered from the application we will constantly reflect on the design at least every 2 weeks during our bi weekly meetings with our client in order to make adjustments. This is also when we will share our plans for the next two week's worth of work in order to receive feedback.

3.6 Development Process

The process we use for this project is the Agile, but more specifically SCRUM. After initially conferring with our client/mentors, we determined that a waterfall approach would lead to an overly loaded second semester. Along with that, having a pre-established codebase to build off of requires less overall design work, and more implementation. This fact works perfectly with SCRUM. By compiling a list of necessary components and breaking them into bi-weekly segments (including their respective planning), we can immediately make progress on this project, and continue it until the final day. Another benefit to using a SCRUM model is the risk aversion. Simplifying our tasks into smaller iterations allows us to verify each task with our clients/mentors at meetings, ensuring we are creating exactly what they want, while also ensuring the task is possible (in its current design) before dedicating a large amount of time to it. Overall, our decision to use SCRUM is to provide the flexibility a client-facing project such as this requires.

3.7 Design Plan

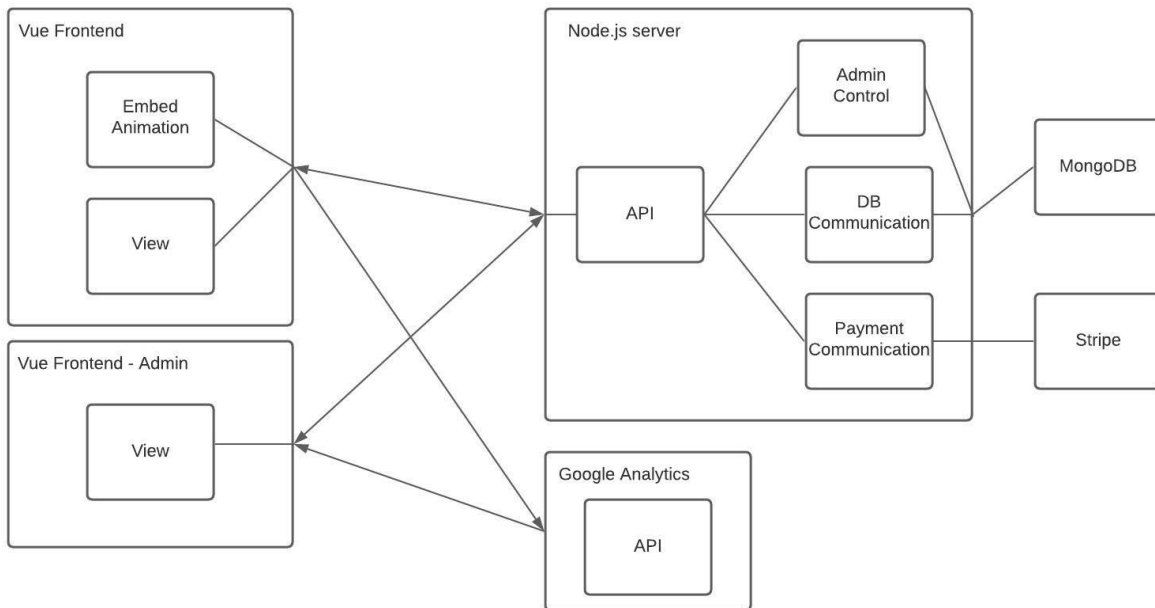


Figure 2: Component Diagram

Description of Components:

Vue Frontend

This is the frontend of the application which is located at <https://www.cowchips4charity.com/> . This module is the interface of the app that is visible to the users. It leverages vue.js components for a consistent design and seamless experience due to the site being loaded once. This component consists of modules such as the embed animation which is where we will embed the animation for the cow chip bingo game. And view which is all of our frontend components. This will communicate with the node.js backend api in order to complete tasks such as login and retrieving and sending data about games and payment.

Vue Frontend Admin

This is the frontend of the application for administrators. This provides the interface in order to create games and choose winning squares. This also will provide graphs and information about the number of users, the amount of donations per game, total donations, what the users are interacting with on the site. This will communicate with the node.js backend in order to retrieve information about the games and donation values and will send data in order to update games. And this will communicate with the google analytics api in order to retrieve data about user traffic.

Node.js Backend

This is the backend component of the application which communicates with the admin and user frontend. This provides an api for the frontend to communicate with. The backend also takes care of payments through stripe. And communicates with the database in order to update information about games, users, and donations. It will also be able to retrieve data about donations for the admin panel.

4 Testing

4.1 Unit Testing

We will need to test the software units in isolation. For both frontend and backend we will need to adjust unit tests for the new code. Since our application is built on a renovation of an existing application, we will need to make sure that our code works with the old unit tests, but we also need to make sure that the new code has separate unit tests that also are working. For new changes created in the frontend and backend, the tests need to reflect that the changes are fully functional.

4.2 Interface Testing

We will test the combined composition of our units by testing the performance of our application. This will require the whole team to use the application both as an admin user and as a normal user. The relevant interfaces in our design would be our user interface, the graphical game display, the backend API, and our database management system. We will be testing these interfaces through Postman and Fiddler.

4.3 Acceptance Testing

For our acceptance testing, we'll need to see how the application works in a mock real life setting. If our application was played by a user, would there be any detrimental issues that can affect how the user can play on the application? Acceptance testing, and especially testing that involves beta users and user study would be beneficial to the performance understanding of our application. Since our client has an understanding of how they want this application to perform, and how this application has performed in the past before our new changes and use cases, we will utilize the client during the testing process. Utilizing our client would ensure that we can guarantee that the application meets their expectations.

4.4 Results

For the testing component, it is extremely important that we have results that reflect accurate information and user friendliness. It's important that we can see a defined product that meets our expectations. Throughout the design process, it's critical that we maintain a working application that can be sent out in production. Failures of this application are having any environment errors

when starting a game.. We've learned a lot about testing, and the different forms of testing we need to dive into to ensure a fully functional almost-bug free application. We are planning a large addition of user testing. As we've been developing the testing process we've realized a need for user testing, and with that our plan is to organize testing from beta users for the game so we can ensure our game is fully working. Our results should display that these beta users didn't have difficulty playing the game and that the application was working smoothly and at an efficient rate.