CowChips4Charity

DESIGN DOCUMENT

Team Number 16
Client: Ken Johnson
Advisor: Lotfi ben Othmane
Members:
Justin Lee: Admin Panel Team Lead
Dustin Schultz: Test Engineer
Tyler Bartleson: Game Team Lead
Elizabeth Li: Game Developer
Meghna Vaidya: Project Manager
Brandon Bui: Admin Panel Developer
sdmay20-16@iastate.edu
https://sdmay20-16.sd.ece.iastate.edu/

Revised: 12/03/19/Version 2.0

Executive Summary

Development Standards & Practices Used

List all standard circuits, hardware, software practices used in this project. List all the Engineering standards that apply to this project that were considered.

- Version Control: 3 main branches: master, dev, QA

- Code Review: Pull Requests will be made before committing to main branches

- Standard Practices: conform to existing standards within partially-completed project we're receiving

- Specialization: members have focus areas to reduce redundancy

- Use Cases: Use Cases will be organized on Trello

Summary of Requirements

Functional Requirements

- The User shall be able to choose a square(s) depending on how many squares they bought
- The User shall be able to choose which game to watch
- The User shall be able to watch the game live
- The Game shall be able to choose an unbiased square for every game
- The Game shall inform which square won
- Multiple games should be able to be run at once
- Game time should last between 5-10 minutes
- Game should have a limit to how late a user can choose a square(s)
- The Admin shall be able to see various data in the data panel (donations per game, the number of people who donated, donations per team, etc.)

Nonfunctional Requirements

- The Website shall look aesthetically pleasing
- The Website should be able to work on any device in any browser
- Security
    - The entire credit card transaction information will be encrypted
- Usability
    - The user shall be able to complete the transaction in one page of the web app
- Reliability
    - The application shall have 100% runtime during football games
    - At least 1000 users will be able to use the web app as it is designed to work
- Operational
    - The web app will be compliant with all policies and regulations set forth by the Boo Radley Foundation

- Performance
    - The application shall be able to support scalability for multiple football game users
    - The user shall be able to login to his/her account in less than 15 seconds
    - The winning user shall receive a notification within 12 seconds of the cow finishing
    - The web app shall be able to calculate the entire cost of the donation once the user is checking out within 3 seconds

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

- COM S 309
- SE 319
- COM S 228
- COM S 311
- COM S 363

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

- Vue.js
- Node.js
- Express
- Heroku
- mLab
- AWS
- Stripe

Table of Contents

List of figures/tables/symbols/definitions

Figure 1. Design Plan

Figure 2. Project Timeline

Figure 3. Process

Table 1. Personal Effort Requirements

1 Introduction

1.1 ACKNOWLEDGEMENT

Our team would like to thank the Boo Radley Foundation and our client, Ken Johnson, for allowing us to work with them for our senior design project. Ken has been very helpful for the entire creation of the project. Ken has also provided financial aid for the product whenever it was necessary. Additionally, we would also like to thank our faculty advisor, Professor Lotfi Ben Othmane for volunteering his time.

1.2 PROBLEM AND PROJECT STATEMENT

The Boo Radley Foundation funds the research of diseases that are common between humans and household pets. They believe research on these animals is a better way to find cures and treatments for animals and humans with the disease. To fundraise for their cause, the Boo Radley Foundation hosts an event called CowChips4Charity. This event is held at halftime of college football games between two participating schools. The event is a bigger scaled version of cow chip bingo. Participants will select a square on a 36x36 grid and if the cow defecates on your square, you win. Currently, the foundation does this in a real field with a real cow and presents the outcome via video feed on YouTube. The cost for the cow and time to make the grid are too high for the foundation. Our goal is to convert the cow chip bingo process to a digital version. We believe this will considerably decrease the costs for the cow chip bingo game and increase the participants and donations of the event.

Our team will develop a cross-platform web application for the CowChips4Charity event. The Boo Radley Foundation will use the web application to enable people to participate in the event. Participants will sign in to the application, select which football game they want to play at, and then select square(s) on the grid to buy. Once the user has at least one square selected they will then pay via credit card. This will occur all the way up to halftime during the football game. During halftime, participants will then go to our app to watch the outcome of the CowChips4Charity event. Our web app will then show an on-screen animation of a cow that the defecates on a certain square in the grid. People who picked the winning square will then be notified that they won and to claim their prize.

1.3 OPERATIONAL ENVIRONMENT

Due to the fact that our end project will only be software and not involve any hardware it will not be subjected to any form of physical hazard such as weather. However, the project is designed to be used during the half-time of college football games both by users watching from home and users in the stadium. This environmental factor does cause us to have to account for the spotty cellular and internet connection that is often found in college football stadiums when creating the project so that even the users in the stadiums can still use the product.

1.4 REQUIREMENTS

Functional Requirements

- The User shall be able to choose a square(s) depending on how many squares they bought
- The User shall be able to choose which game to watch
- The User shall be able to watch the game live
- The Game shall be able to choose an unbiased square for every game
- The Game shall inform which square won
- Multiple games should be able to be run at once
- Game time should last between 5-10 minutes
- Game should have a limit to how late a user can choose a square(s)
- The Admin shall be able to see various data in the data panel (donations per game, the number of people who donated, donations per team, etc.)

Nonfunctional Requirements

- The Website shall look aesthetically pleasing
- The Website should be able to work on any device in any browser
- Security
    The entire credit card transaction information will be encrypted
- Usability
    The user shall be able to complete the transaction in one page of the web app
- Reliability
    The application shall have 100% runtime during football games
    At least 1000 users will be able to use the web app as it is designed to work
- Operational
    The web app will be compliant with all policies and regulations set forth by the Boo Radley Foundation
- Performance
    The application shall be able to support scalability for multiple football game users
    The user shall be able to login to his/her account in less than 15 seconds
    The winning user shall receive a notification within 12 seconds of the cow finishing
    The web app shall be able to calculate the entire cost of the donation once the user is checking out within 3 seconds

1.5 INTENDED USERS AND USES

This product is intended for two colleges to go against each other to raise money for the Boo Radley Foundation. The end product should allow for users to donate through the website to support their college and the Boo Radley Foundation. The end product should also allow for the user to see the game on their device.

1.6 Assumptions and Limitations

Assumptions

Users will have a device from which they can access can access the internet
- In order to be able to get to the website to play the game, users will need an internet connection

Users will have a credit or debit card
- In order to bet on a square or squares in the game

The product shall comply with all laws and regulations
- States have different privacy laws

Admin panel users will not have an in-depth software development knowledge

Limitations

Internet connection in the college football stadiums tends to be spotty.

1.7 Expected End Product and Deliverables

The deliverables we will produce will be amendments to existing documentation, and an expanded final project cross platform web application. A large portion of the documentation for the project already exists, but the additional changes we're making will need to be reflected in updated documents. The new/adjusted deliverables will be: a new use case diagram for the analytics portion, an updated communications diagram, and a scope and requirements document for the amendments we will be making.

Analytics Use Case Diagram
One of our major changes is a large expansion of the analytics portion of the admin dashboard. This will describe the interactions that an admin will have available to them, and the flow of events of those actions. This document solidifies the client and team's understanding of the expansion's expected flow.

Updated Communications Diagram
The existing project already contains a well-documented communications diagram. Our expansion to the project will entail new communications between components of the application. To ensure understanding of these additions, the existing communication diagram will be updated to reflect our new features.

Scope and Requirements Document
This document will serve as a high-level guide for the desired output of our work on the application. It will be frequently evolving as time allows additional expansions. Despite this, an initial version will be approved by the client to ensure a full understanding of the expected work.

Final Version Web Application
A fully functional version of the expanded web application will be the ultimate deliverable. It will include all our new features, including game generation, data visualization, and analytics

2. Specifications and Analysis

2.1 PROPOSED DESIGN

Our main components that we'll implement are the game animation, and admin panel analytics.

For the game animation, we've begun discussion on potential tools and libraries that fit into our existing framework that adhere to the requirements of being lightweight for spotty mobile connections. A conclusion hasn't been reached on which tool to use, the decision is awaiting confirmation with client of an idealized style for the animation.

For the admin panel analytics, we're currently investigation data visualization libraries that fit into our framework. Also under our consideration, analytical tools that will provide output to for use in our selected data visualization techniques. While specific analytical desires are still being refined, we've halted selection of pre-built analytics, or even if we'll use them at all, depending on what level of analysis is desired be available.

We will strive to follow the IEEE Standard Std 829-1998 within our test planning.

2.2 DESIGN ANALYSIS

Our components fit into the existing application, with somewhat defined "openings" in the code ready for development. We've examined these openings, and while nothing has been committed and tested, discussion with our technical contact leads us to believe this is the correct approach.

We believe that implementing new libraries for data visualization, analytics, and animation will be beneficial to both development time, and maintainability of the product. This will conform to the standards already existing in the project, which already uses libraries where possible to avoid redundant development.

Selecting the correct tools mentioned above will be a time consuming activity, but the benefits gained by faster development and maintainability gains will outweigh this upfront cost.

2.3 DEVELOPMENT PROCESS

For this project we are following the Agile development process. We settled on using the Agile development process because we thought that it would be the easiest and most efficient way to have meetings, work together, and organize the tasks required to complete our project. More efficient meetings means that we will be able to make the most of whatever time we are able to have all of our team members together; allowing us to come together and discuss what we have all gotten done, what we all still have to do, and what problems any of us are having. Agile will help work together easier because we will know the current progress of our project. Finally, the Agile process will help us organize the tasks required because it allows us to know who has been assigned which task and the approximate time of when each task should be complete.

2.4 DESIGN PLAN

This project is a charitable bingo game with an admin panel feature. Therefore, we will have the standard client, server, and database setup. We do not know exactly which tools would be used for the server and database, but since this is a web-based project, it is assumed that a fair amount of HTML, JS, and CSS will be used. As we get more information, we will be able to create more specific diagrams that go along with the principles of UML.
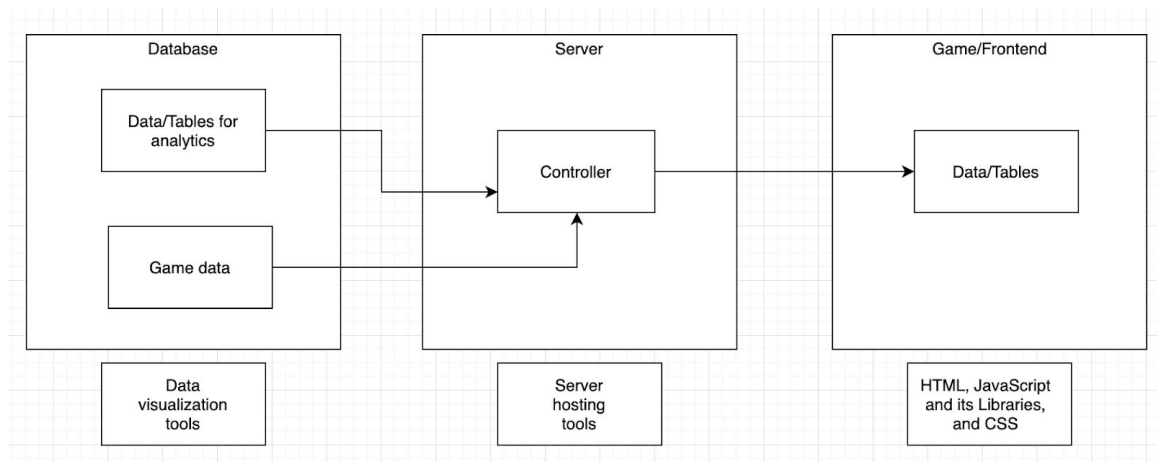

Figure 1: Design Plan

3. Statement of Work

3.1 PREVIOUS WORK AND LITERATURE

This project is basically a bingo game that involves betting on some squares, which are two basic ideas that have probably been done many a time.
We have access to the previous teams work on this project, and it can be used as a basis for our work this time around.

Advantages
- The access to the previous teams work, gives us more of an insight as to what our client's requirements actually means.
- It gives us a visual on what our end product should look like and on what parts we can build upon and make better

Shortcomings
- We are limited to the technologies we are using, we have to use what the previous team used as to keep everything on the same page.

3.2 TECHNOLOGY CONSIDERATIONS

We will be using the previous group's technology stack (Vue.js, Node.js, etc.). We also will be using any data-visualization tools that would be compatible with Vue.js and Node.js, such as Chart.js. This

will make it easier the transition relatively easier, as we will not need to worry about making sure the previous group's project would be well-suited to work with our version of the updated project.

3.3 TASK DECOMPOSITION

There are three main aspects to this project: the game itself, an admin panel with analytic data from the games that have been played, and the scalability of the project itself. In order to complete all of these aspects of the project it will help to break them down into multiple tasks as follows:

Admin Panel Tasks:
- Analytics/Statistics
  - Specific Statistics
    - Per Game
      - Donations per game
      - Number of people who donated
      - Donations per team
      - Amount of winners per school
    - Per University
      - Lifetime total
      - Personal best game (donations, players, winners)
    - Admin Only
      - Live total money
      - Bets per tile (coordinate as well as value)
      - Donation frequency vs time
  - Adjust data collection to track
    - Keep tile coordinates
    - Donation: get game ID with team association
- Data Visualization
  - Research viable framework/library (has to happen first)
  - Add subsection to admin panel
  - Support for the specific analytics
    - Pie charts
    - Bar charts (Timeline with hits)
    - Heatmap
- Live Stuff
  - Setup event-driven (upon donation received) data sending to web app

Game Tasks:
- Update website UI
  - Update home page
    - Update logo
    - Change buttons
    - Add link to Facebook page

- Update donation page / credit card pages
- Update about page
- Update account page
- Update register page
- Update select org page
- Update select game
- Update select team
- Update your tiles page
- Create look of game
  - 6 x 6 grid
    - Randomly numbered squares
  - Animated cow
  - Digital field & patty
  - Add a game timer
  - Design winning messages
  - Mark squares selected by users
- Add game functionality
  - Make cow move randomly
  - Make game last between 5-10 minutes
  - Make cow move over winning tile as the cow patty is generated
  - Display winning messages when game is finished

Scalability:
- Make able to appear correctly on different sized devices
- Allow multiple games to be run at once
- Allow multiple users to play a game
- Make able to appear correctly on different internet browsers

3.4 POSSIBLE RISKS AND RISK MANAGEMENT

A part of this project that slow us down would our collective teams knowledge of the technologies we are using. We are going to be using technologies that most of us on the team are not knowledgeable about, including Vue.js, Node.js Express, Heroku, mLab, AWS, and Stripe. While some of us may have had some exposure to these, nobody is proficient in them, and it will require all of taking the time to really learn what each one of these do, and how they play in to our overall project.

3.5 PROJECT PROPOSED MILESTONES and EVALUATION CRITERIA

The milestones of our project will largely revolve around the two different aspects of our project. Some milestones of the Game aspect of our project will be: creating working visuals for the game, creating backend so that the game can be run from start to finish, one game is able to be played at a time, multiple games are able to be played at a time, the game can scale to multiple sized devices, the game can be run on multiple browsers, and the game can run with little internet connection. Some milestones for the Admin Panel aspect of our project will be: able to collect data from the game, able

to store data from the game, able to analyze / calculate additional data from the data collected from the game, and being able to show that data as graphs and diagrams. The final milestone for the project will be our final presentation to the client at the end of the year.

3.6 PROJECT TRACKING PROCEDURES

Our team will github to track our progress. We will be using a storyboard with use case cards to keep track of what needs to be done, who is working on which card, which card needs to be tested, and which card is done.

3.7 EXPECTED RESULTS AND VALIDATION

A desired outcome is to have a working simulation game of Cow Patty Bingo. This includes an animated cow randomly roaming a digital Bingo card that looks like a pasture. Each square will be numbered and be able to be bet on by a user. The cow will then randomly deposit a cow patty indicating which square wins. Another desired outcome is a working admin panel that can analyze and display various data desired by the client in the form of easy to read graphs and charts.

4. Project Timeline, Estimated Resources, and Challenges
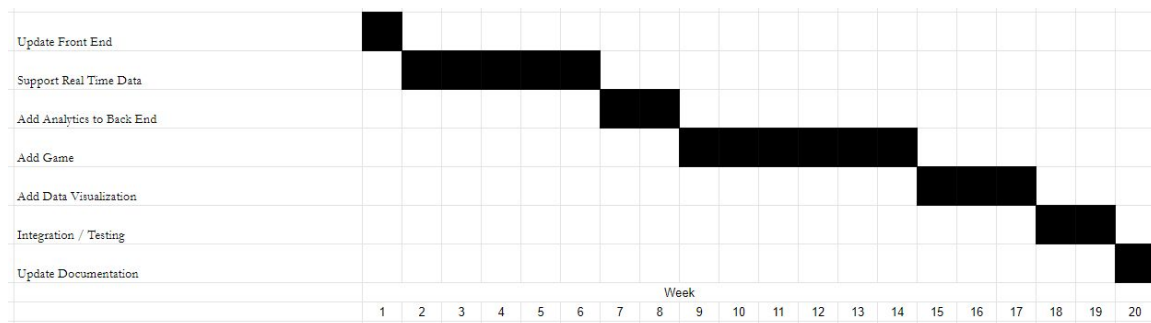
4.1 PROJECT TIMELINE

Figure 2: Project Timeline

We're proposing to split the development up into two main stages: game animation, and admin panel tasks. We'll begin with game-related tasks, and move to admin panel analytics after.

We believe that a larger portion of time will need to be dedicated to the admin panel activities, since it'll be using existing libraries, and those still need investigated as to which we want to use (if any).

4.2 FEASIBILITY ASSESSMENT

The realistic projection of our project will be that the website will be able to run multiple games at the same time while the admin panel will be able to see various data on the games live. Some challenges we thought of was adapting and understanding the previous group's project. This project's feasibility will also be affected if the users are not able to connect to the internet and donate and play the game. Another factor that could affect the project's feasibility is getting a fanbase to

play. Because the game is niche, not everyone might want to play this game. It is also hard to start a fanbase and keep it growing.

4.3 PERSONNEL EFFORT REQUIREMENTS

Task	Description	Approximate Hours
Update Front End	The current frontend will be updated to look more modern, and more aesthetically pleasing.	20 hours
Support Real Time Data	For the analytics, real time data needs to be implemented to get a more accurate understanding of the current live games.	100 hours
Add Analytics to Back End	Various statistics will be useful to get an understanding of how users interact with the game. These will be usable by administrator users for promotion and marketing.	50 hours
Add Game	The game itself is the most significant addition. It will be the simulated cow portion that enables the whole operation to function in a digital environment. There will be animations that respond to incoming data for positioning.	150 hours
Add Data Visualization	Once analytic are generated, visualizing them will be the next step. These visualizations will work for promotion, as well as getting an understanding at a glance.	75 hours
Integration/Testing	Final testing and integration will be done to ensure whole system functionality. Bugs discovered in this stage will be patched..	50 hours
Update Documentation	We will enhance and expand current documentation to reflect our changes, as well as clarify portions that we found challenging to understand as an initially uninformed third party. This way future maintainers will benefit from our initial misunderstandings.	15 hours

Table 1: Personal Effort Requirements

4.4 OTHER RESOURCE REQUIREMENTS

Due to the fact that this project will be entirely software the only resources we may need are different softwares and frameworks.

4.5 FINANCIAL REQUIREMENTS

As of now it does not seem like our project will have any financial requirements unless we need to purchase professional versions of software.

5. Testing and Implementation

We will need to integrity test our data, playtest our game, and conduct user-studies to see how the UI reacts to each scenario. It will be essential to test the entire bingo game with every rule, the ability to choose squares, watching the game live, determining the winning square, the ability to run multiple games at once, the game time, user limits, and the admin privileges that include donations per game, those who donated, donations per team, and other data.

Our test cases should be based off of the game rules- the ability to choose a square by the player based off the squares purchased, user being able to choose which game to spectate, watching the game live, unbiased square, determine winning square selection, having multiple games run at once, the limited time frame for each game, limits to how late a user can select a square, and having Admin be able to accurately see the data such as donations per game, the number of those who donated, and donations per team.

The anticipated test results for the actual game should be based off of the rules of how Bingo works and having the constraints properly set for one winning square per round, having more than one game run at once, having the game only lasts between 5 and 10 minutes, limiting how late to how a user can select a square, and accurate data should be exact to the predicted data that is calculated by the tester.

We will be using Test Driven Development practices, with test cases being written before the story's code. This ensures that the story's requirements are fulfilled, and that test cases are created that can be ran again later to test for regression.

Once these features are implemented, we can perform the actual tests and evaluate the results to improve our product and retest, after documenting our entire process and results. We can run those tests and a lot of our results are based off of what is simulated in the game. We believe that the biggest challenge would be integrating the frontend and backend and allowing the user to watch the games live and having multiple games go off at once.

5.1 INTERFACE SPECIFICATIONS

Our expansion of the game functionality will include new endpoints for data transfer

Game Endpoints
- GET /event?id=<number>/game - Retrieve updates for the current game (cow sprite movement, end declaration)
- GET /event?id=<number>/stats - Retrieve updates for the current game's (active users, team percentage, etc)

5.2 HARDWARE AND SOFTWARE

We will be using a CI/CD pipeline ran primarily by Travis CI, with Jest and Cypress ran as plugins. Travis CI is the driver of the pipeline, with Jest and Cypress test cases being ran as part of the build process ran by Travis. Every commit/merge to a branch will kick off a build within Travis, allowing errors caused by the new commit to be found immediately.

Jest serves as our backend testing of components. It will execute test cases, similar to JUnit like the team is already familiar with. All existing Jest test cases will be run every build to ensure no regression.

Cypress serves as our frontend testing. It executes test cases by simulating user input to a browser, testing the functionality of our site from the user perspective. These test cases will also be run on every build for regression.

5.3 FUNCTIONAL TESTING

We will be using unit testing and acceptance testing.

Unit testing will be the primary focus while developing. Tests will be developed before code is written to ensure proper testing of individual parts of the code is complete. These tests will revolve around the main objectives of the code, as well as any internal components of more complex code as determined by a case-by-case basis. These test cases will be included in the CI/CD pipeline and automatically run upon every new commit to prevent regression.

Acceptance testing when significant components are completed. These tests will be done by hand, with a developer (or client, when doing final testing) acting as a user, navigating the site to test the flow, usability, and functionality of the application.

Smaller sub-portions of acceptance testing will be possible using Cypress. These will be considered part of the unit testing since they focus on a more micro level.

5.4 NON-FUNCTIONAL TESTING

We will be using integrated systems testing and acceptance testing.

Integrated systems testing will occur to test the application under a more realistic environment. This includes a high amount of concurrent traffic, ability to function with imperfect connection, time to load, etc. We will also measure time time taken to perform basic functionalities, under normal traffic and high amounts to ensure the systems are fast enough to support the users.

Acceptance testing will occur similarly to integrated systems testing, with a focus more on normal conditions rather than stress testing ones. Usability will be emphasized, with individual users (ideally some unconnected to the development of the project) performing the tests, to gather feedback on

the usage flow, and intuitiveness of the UI. Feedback will be discussed with the client (who also functions as an acceptance tester), and implemented as deemed appropriate.
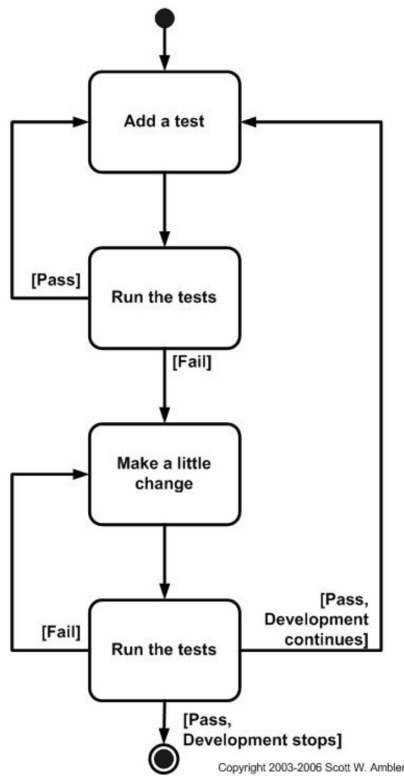
5.5 PROCESS

Figure 3: Process

Our process should start as soon as the feature we want to test is implemented. We will go through the process of creating the test cases, run the tests, and if tests fail we can go ahead and implement the necessary changes to get them to pass and rerun the tests. The cycle shall repeat until there are no more features to test.

5.6 RESULTS

As of now, there has not been any implementation for the project that can be tested. Therefore, there are no failures or successes during a nonexistent testing phase. Currently we will have to anticipate how testing will go and learn how to test and improve based off of the results. It is assumed that all team members should know how to run unit tests, playtests for the game, and debug accordingly. Any form of errors are possible when testing, so it is best to get our project up to speed as best as possible and resolve any errors as soon as possible.

Since our project involves a majority of UI, a lot of the testing is visual and is dependent on the results of the game actions. A large amount of the testing would be a result of playing the game and seeing the UI interact with it and based off of the tester's analysis of the visuals.

This project will truly test the knowledge and capabilities of every team member. The biggest challenges would consist of getting the frontend and backend working together and implementing the analytics.

6. Closing Material

6.1 CONCLUSION

Our goal for this project is to be able to provide the technological solutions our client needs for the CowChips4Charity initiative to raise money for the Boo Radley Foundation. In order to do this we will be creating interactive web application where users can bet on square to play the CowChips4Charity game. We will also create an easy and organized admin panel for the admin user to be able to see analytical data about any of the games that are played. The intention of the project to be able to raise awareness and money for the Boo Radley Foundation.

6.2 REFERENCES

We have been extensively using the documentation created by the previous group for understanding the existing implementation. It has been critical for our success. Similarly, our points of contact Ken Johnson (business side) and Daniel Lev (previous team member, technical side), have provided us immense support in our work.

6.3 APPENDICES

http://cowchips4charity.com/ : Our clients website which he currently has.