

CowChips4Charity

Design Document

sdmay19-28

Client: Ken Johnson

Advisor: Lotfi Ben Othmane

Connor Rust - Backend Lead

Jack Boike - Backend / Scribe

Ben Meeder - Frontend Lead

Kenneth Ho - Frontend / UX / Meeting Scribe

Daniel Lev - Business Lead (PM)

Alex Lev - Administrative / UI

Team email: sdmay19-28@iastate.edu

Team website: <http://sdmay19-28.sd.ece.iastate.edu/>

Revised: 9/25/18 - V1.0

Table of Contents

1 Introduction	2
Acknowledgement	2
Problem and Project Statement	2
Operational Environment	3
Intended Users and Uses	3
Assumptions and Limitations	4
Expected End Product and Deliverables	4
2. Specifications and Analysis	6
Proposed Design	7
Design Analysis	9
Testing and Implementation	9
Interface Specifications	9
Hardware and software	9
Functional Testing	10
Non-Functional Testing	10
Process	10
Results	7
4 Closing Material	12
4.1 Conclusion	12
4.2 References	12
4.3 Appendices	12

List of Figures

Figure 1: Proposed Use Case Diagram

Figure 2: Proposed Component Diagram

Figure 3: Project Process Diagram

List of Tables

None at this time.

List of Symbols

None at this time.

List of Definitions

CowChips4Charity: A charitable game initiative that is based on bingo fundamentals of the Boo Radley Foundation to increase donations and outreach.

1 Introduction

1.1 ACKNOWLEDGEMENT

We would like to thank our client, Ken Johnson, for the opportunity to work with him and the Boo Radley Foundation for this project. Ken has provided significant assistance in the forms of financial aid and project context throughout the duration of this project. We would also like to thank Professor Lotfi Ben Othmane for volunteering his time to be the faculty advisor.

1.2 PROBLEM AND PROJECT STATEMENT

The Boo Radley Foundation aids in the research of diseases found in both humans and their animal companions. In order to raise funds and awareness, the organization hosts interactive events at college football games called CowChips4Charity. The CowChips4Charity event is a modern version of cow chip bingo. Currently, the CowChips4Charity event is operated

manually and is executed without leveraging any technologies. We believe the development of a web application interface for this event will increase the amount of participants and resulting donations that the Boo Radley Foundation will receive.

We will develop a cross platform web application for the CowChips4Charity event to be used during collegiate sporting events and possibly professional sporting events in the future. The web application will provide an efficient way for sporting event attendees to participate in the cow chip bingo game. Users will be able to sign in via Facebook, Google, or as a guest. Once the user is signed in, they will then be able to select the sporting event they are attending and the team they are supporting. The user will be able to acquire square(s) for the cow chip bingo game through a donation via credit card. The selection of square(s) can be done well in advance of the game. Users who select the winning square for their sporting event will be notified via text message and will then receive a prize in the mail.

1.3 OPERATIONAL ENVIRONMENT

The operating environment of our application will be from a laptop or mobile device. The web application is hosted on off-site servers, and thus there are no conditions or hazards from the outside world that we need to take into account. Users will be able to access the application during or prior to the sporting event. Due to the large amount of anticipated users and limited cellular coverage during sporting events, the web application needs to be quick and easy to use.

1.4 INTENDED USERS AND USES

The team has identified two types of intended users for this application: regular users and admin users. The regular user is an individual who is accessing the application with the purpose of participating in the CowChips4Charity game. The regular user will be able to sign up for a regular account, select bingo square(s), and donate money to the Boo Radley Foundation. This user will need a simple and quick user interface that can be used from a computer or mobile device. Regular users will be able to access the application prior to and during the sporting event.

The admin user is an individual who works for the Boo Radley Foundation and is running the CowChips4Charity event. This user have access to information on all of the CowChips4Charity participants. The admin user will have the ability to send out texts blasts and select the winning square for each occurrence of the CowChips4Charity event. We are anticipating that the admin user will not be someone with a software development background. Given this constraint, the application will need to be easy to use and maintain.

1.5 ASSUMPTIONS AND LIMITATIONS

Assumptions

- Users will have a cell phone with internet access
 - To receive text message and fill in the address form when the user's square wins
- Users will have a credit card
 - In order to complete square(s) donation
- Users will not be citizens of the European Union or residents of California
 - Due to the GDPR and CCPA privacy laws just passed
- Client will provide logo design and theme for application
- Admin users will not have deep software development knowledge
- Winning square selection and live stream will be handled by external party

Limitations

- The application cannot be used in California
 - Due to the CCPA privacy laws
- The application cannot be used by citizens of the European Union
 - Due to the GDPR privacy laws

1.6 EXPECTED END PRODUCT AND DELIVERABLES

The deliverables of our project will be documentation and a cross platform web application. Documentation will be used by future developers and admins of the web application to maintain and improve the application over time. Documentation deliverables include: use case diagram, component diagram, communication diagram, scope and requirements document, local database setup instructions, and instructions on how to set up and start the application.

Scope and Requirements Document (Delivery: 9/8/2018)

The Scope and Requirements Document was created after our initial meeting with the client. This document is to be used as a general guidance tool for the development of the CowChips4Charity web application. The document outlines the functional scope of the web application and how the product will be utilized in our client's Boo Radley Foundation. In addition, the document also specifies the usability requirements of the web application in order to fully accommodate for our client's needs. Once the client assumes full ownership of the web application and source code, this document will serve as a foundation for future maintenance and development.

Use Case Diagram (Delivery: 9/19/2018)

The use Case Diagram was created after the completion of the Scope and Requirements Document. This document describes how users will navigate through the flow of the

application. The document identifies two different user types for the web application and their respective capabilities. It is essential that this document is correct and approved by our client as it is one of the building blocks that determines how the application will be built.

Component Diagram (Delivery: 10/4/2018)

The Component Diagram was created after the completion of the Use Case Diagram. The diagram analyzes the flow of the screens to determine what individual pieces or components will be needed. This document is used by the developers and project management to break down the application into individual user stories in order to efficiently and correctly allocate work.

Communication Diagram (Delivery: 10/4/2018)

The Communication Diagram is used to provide a visualization of how different components within the web application interact with each other. This document will be used as the main reference for developers when they develop and connect different components throughout the development lifecycle. Future developers of the CowChips4Charity application can reference this document when performing maintenance or creating updates for the app.

Local Database Setup Instructions (Delivery: 9/9/2018)

The Local Database Setup Instructions are a set of technical instructions to be used by developers who will be working on the application once our current engagement has concluded. During future development iterations of this application, it is important for developers to not clutter up the production database with false/test data. This instruction document will provide easily interpretable directions on how to setup a local database to prevent the previously mentioned issue.

Application Startup and Setup Instructions (Delivery: 5/1/2019)

The Application Startup and Setup Instructions will be delivered alongside our final application. This document will give step by step instructions to admins and future developers on how to setup and start the application correctly. This document can be useful for the cases when our client wishes to save money and turn off the application when it is not being used, or if the client decides to move to a different set of servers. The document will be simple enough for people not from a software development background to use.

Final Version Web Application (Delivery: 5/1/2019)

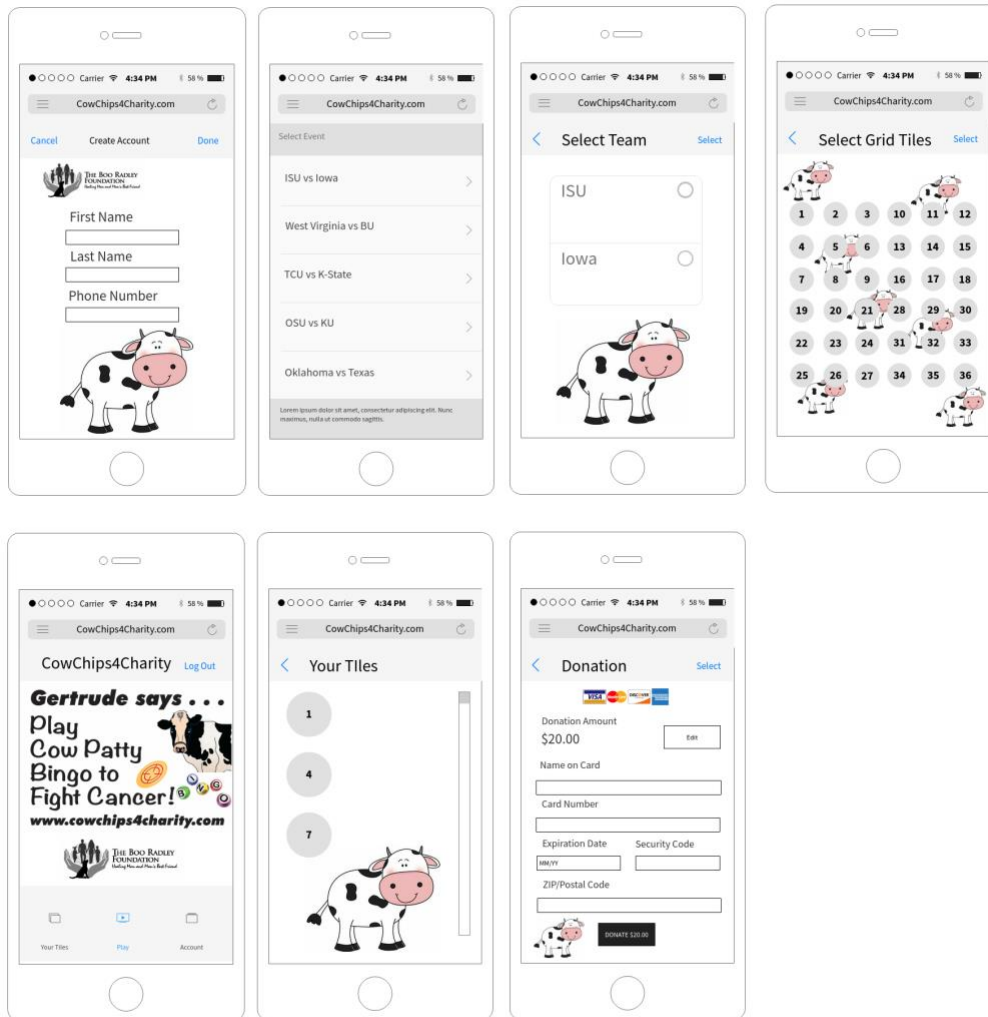
The web application deliverable will be a production scale web application capable of handling hundreds of users at any point in time. The application will be intuitive and easy to use, be aesthetically pleasing, and be secure. Additionally, the database will be secure and redundant.

2. Specifications and Analysis

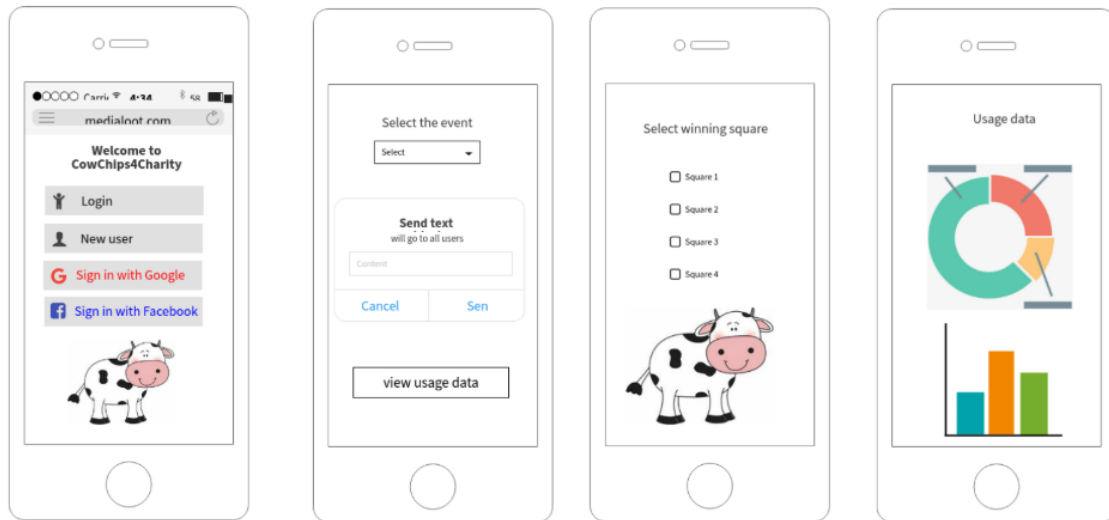
2.1 PROPOSED DESIGN

We have been coming up with the requirements between our group and having our client approve them as we go. We have created various documents so the client can visualize what the application will look like. We have also setup all of the infrastructure for our application, therefore when all of our design is complete we are able to start developing without any holdup. The requirements are listed below.

Wire Frames based of Use Case Diagram (Regular User):



Wire Frames based of Use Case Diagram (Admin User):



Functional:

Front end:

- 1) Login via facebook, google, or guest
- 2) Credit card verification for payment
- 3) Be able to select multiple squares from a 6 x 6 grid
- 4) Admin page
 - a) Select Winning square
 - b) Add users,ban users, delete users
 - c) View previous users from database with filtering

Back end:

- 1) Store square purchases to specific user profiles
 - a) A user profile may have multiple square purchases
 - b) Users acquire squares through donations
 - c) Multiple users can own the same square
- 2) Store school selections to specific user profiles
 - a) Record total amount of money donated from specific schools
 - b) School info should be stored and easily accessed for future marketing activities
- 3) Twilio implementation for prize disbursement
 - a) Winners will receive a Twilio text message form for address and contact information
 - b) Prizes will be mailed out to participants and will NOT be monetary
- 4) Winning square(s) is selected through admin input

Non-Functional Requirements:

- 1) Only the Admin shall be able to select the winning squares (security)
- 2) The application shall 100% runtime during football games (reliability)
- 3) The application shall be able to support scalability for multiple football game users (performance)

- 4) The user shall be able to complete the transaction in one page of the web app (usability)

2.2 DESIGN ANALYSIS

We have developed requirements and design documents that allows us to visualize how this application will look and perform. All of these requirements and design documents have been approved by our client and will give us a template for how the application will perform.

We have also created wireframes for the screens that the user and admin will see when they are using the application. These wireframes are in the process of being approved by our client. Once we receive feedback from our client and build the final versions of the wireframes we will know how our application should look.

Our project has very strong set of requirements and scope, we also have through documentation and wireframes that will aid us in making sure that we meet all of the clients requirements.

Moving forward we will update our wireframes based on the client feedback. Once we are completed with the wireframes we will be able to start the development of our project.

3 Testing and Implementation

3.1 INTERFACE SPECIFICATIONS

We will have 2 different API's one that the single page frontend web application will connect to, and one that the admin panel will connect to. The first API will be for the frontend single page application that is implemented using Vue.js. The Endpoints that need to be defined are as follows:

- Payment Information
- User Information
- Game Information
- Authentication

The admin backend will need to have the following endpoints:

- Cowchips Games CRUD
- Users CRUD
- Events CRUD

3.2 HARDWARE AND SOFTWARE

- Indicate any hardware and/or software used in the testing phase
- Provide brief, sample introductions for each to explain the usefulness of each

Jest - Javascript Automated Test Runner. We will use this to run JS unit and integration tests. The main advantage of Jest is its powerful watcher that allows for queries based on filename, test name, and suite name. We plan to take full advantage of this test runner and fully implement Test Driven Development within our team.

Cypress - Cypress is an end to end browser automation test runner that will run tests in a Google Chrome or Headless Electron browser. This allows the team to create tests that hit every portion of the application simulating real user input, and verifying that features are working as expected.

3.3 FUNCTIONAL TESTING

The main functional testing we will use will be unit testing, integration testing, and end to end system testing. These will manifest themselves in ways that are slightly different depending on the project.

For the front end we will implement unit testing that mounts the components to a virtual DOM and then simulate actions and verify internal states of all components. We will also mock out the Axios HTTP client allowing us to verify that the correct calls are being made to the backend API.

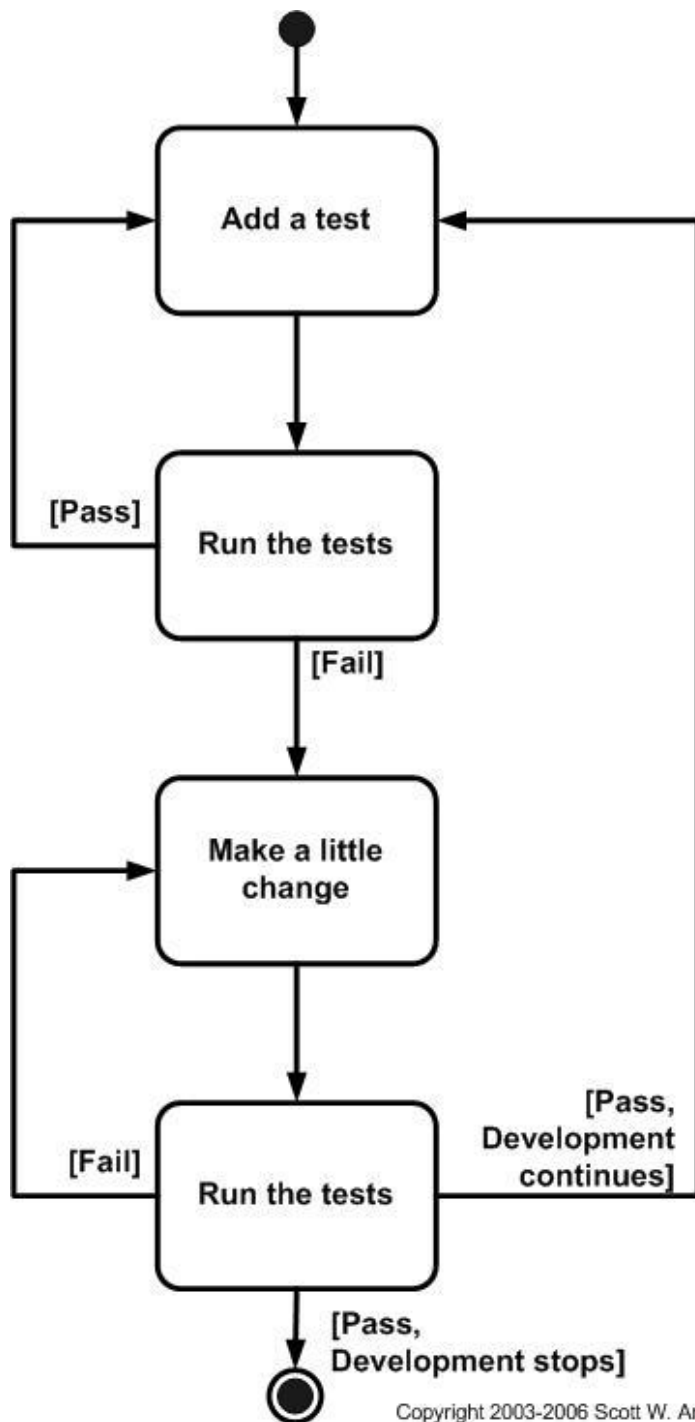
For the back end we will implement unit testing for individual components and methods that do not interact with the database. Additionally, we will implement integration testing to test and verify that API calls are fully implemented to the spec. The combination of these tests will make it so that we have a high confidence in the API's correctness.

3.4 NON-FUNCTIONAL TESTING

For non functional testing we will do this during our IST process and UAT process. For IST our team will test all of the non functional and functional requirements. During our UAT we will provide the fully functioning application to our client and he will test the application along with the other users who will be using the application. One of the main concerns are reliability and performance during a football game when many users will utilize the application; therefore our team has decided to do load testing for this.

3.5 PROCESS

We will test everything during development by utilizing Test driven development. Once all of our main functionalities are pushed and approved by the team then we will move into IST and then when IST is complete we will solve all the bugs that arise. Once all of the bugs are solved in IST we will continue to go to UAT and if the client or Users report any bugs we will fix them. Once all of these processes are done we will then submit the client with a final product.



3.6 RESULTS

We have set up our environment and infrastructure for development and have not had any issues thus far. We will most likely run into more issues once development starts. Our design and documentation portion of

the project is going smoothly, and the client is happy with our results.

4 Closing Material

4.1 CONCLUSION

We have completed all of the requirements, scope, documentation, and are in the processes of getting our wireframes approved by the client. Once we have a final version of the wireframes we will be able to start the development of our project.

4.2 REFERENCES

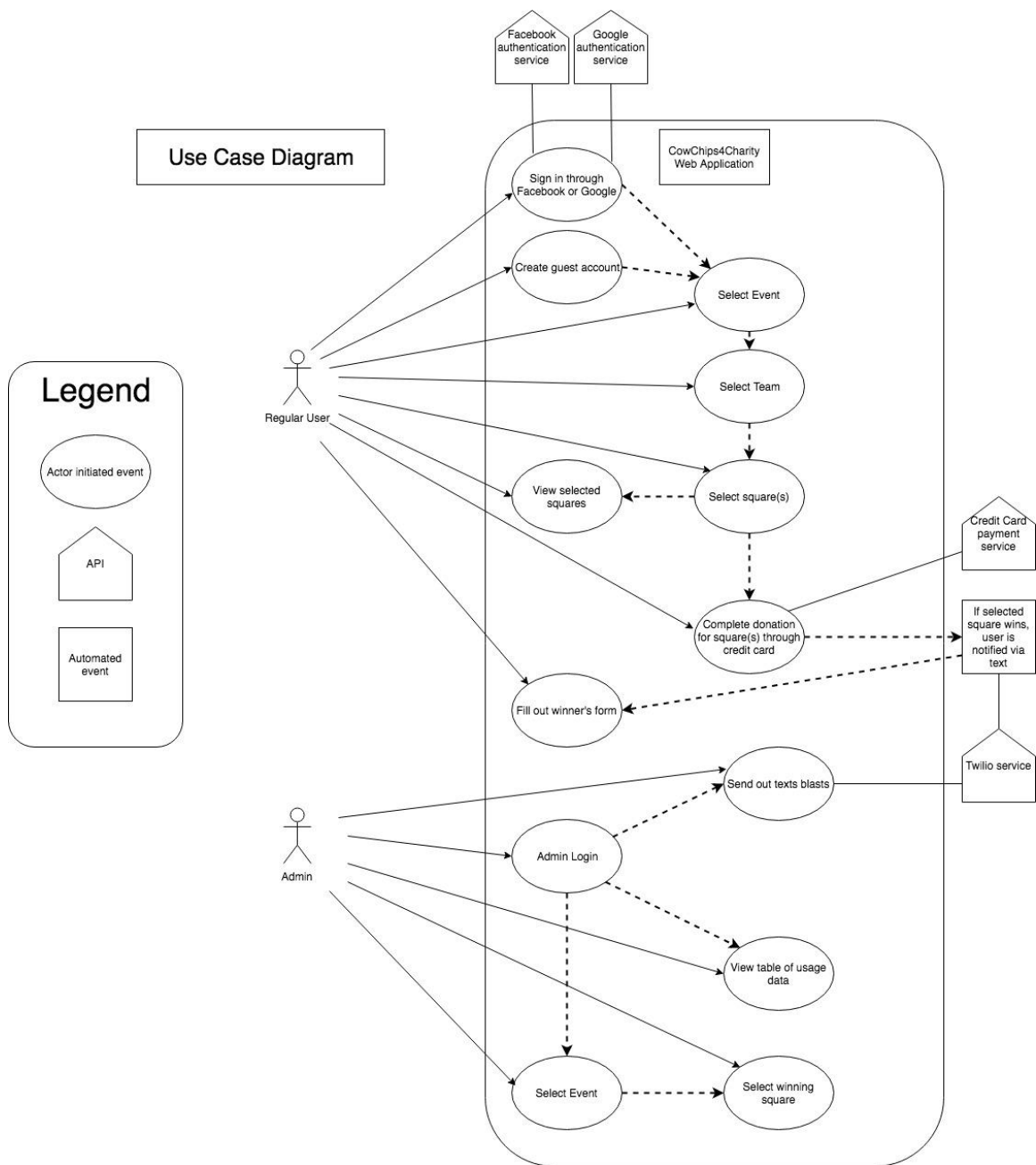
We referred to material we learned in other classes to come up with the design documents we have created. We also interviewed our client thoroughly to come up with design criteria. The website he has currently made has helped us decide the layout and style of our application.

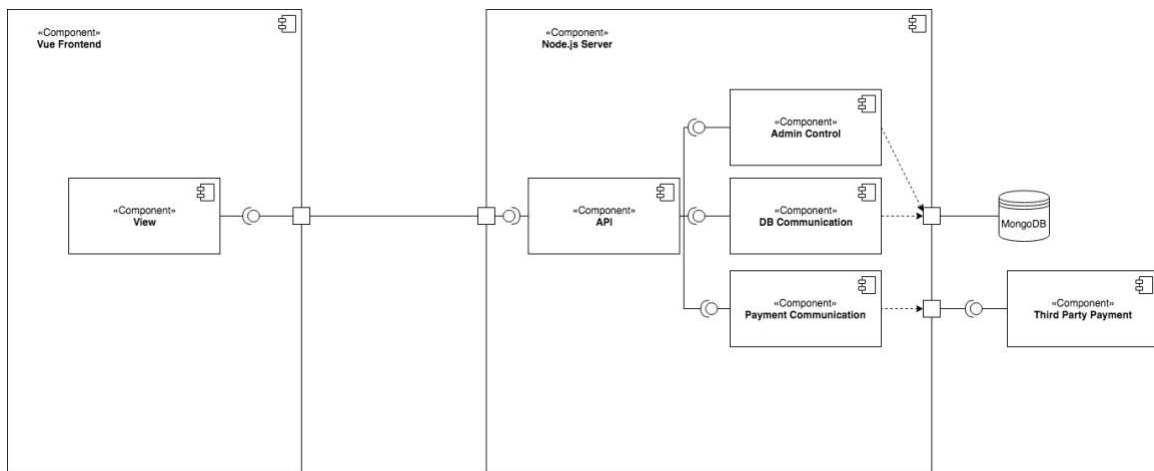
<http://cowchips4charity.com/>

4.3 APPENDICES

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

Use Case:





Component Diagram