

CowChips4Charity

Project Plan

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: 12/2/18 - V3

Table of Contents

1. Introductory Material	1
1.1 Acknowledgement	1
1.2 Problem Statement	1
1.3 Operating Environment	1
1.4 Intended Users and Intended Uses	1
1.5 Assumptions and Limitations	2
1.6 Expected End Product and Other Deliverables	2
2. Proposed Approach and Statement of Work	4
2.1 Objective of the Task	4
2.3 Constraints Considerations	5
2.4 Previous Work And Literature	6
2.5 Proposed Design	7
2.6 Technology Considerations	14
2.7 Safety Considerations	14
2.8 Task Approach	15
2.9 Possible Risks And Risk Management	17
2.10 Project Proposed Milestones and Evaluation Criteria	17
2.11 Project Tracking Procedures	18
2.12 Expected Results and Validation	18
2.13 Test Plan	19
3. Project Timeline, Estimated Resources, and Challenges	20
3.1 Project Timeline	20
3.2 Feasibility Assessment	20
3.3 Personnel Effort Requirements	20
3.4 Other Resource Requirements	23
3.5 Financial Requirements	23
4 Closure Materials	23
4.1 Conclusion	24
4.2 References	24
4.3 Appendices	24

List of Figures

Figure 1: Proposed Use Case Diagram

Figure 2: Proposed Component Diagram

Figure 3: Wire Frames based off Use Case Diagram (Regular User)

Figure 4: Wire Frames based off Use Case Diagram (Admin User)

Figure 5: Work Breakdown Structure

Figure 6: Project Process Diagram

Figure 7: Diagram of Testing Procedure

List of Tables

Table 1: Use Case Diagram Table - Use Case and Actors

Table 2: Use Case Diagram Table - Adjacent System / Components

Table 3: Personal Effort Tasks and Description

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. Introductory Material

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 (Johnson, Ken). We would also like to thank Professor Lotfi Ben Othmane for volunteering his time to be the faculty advisor.

1.2 PROBLEM 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 (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, through their profiles credentials, 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 OPERATING 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 INTENDED 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 has access to information on all of the CowChips4Charity participants. The admin user will have the ability to send out text 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 OTHER 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. Proposed Approach and Statement of Work

2.1 OBJECTIVE OF THE TASK

Develop a cross platform web application for the CowChips4Charity event during collegiate and professional sporting events. The web application will provide a way for attendees of a sporting event to interface with the Cow Chip Bingo game sponsored by CowChips4Charity. This application will consist of a user being able to login via Facebook and google or they can make an account (Facebook for Developers). Users will also be able to access the app as a guest. The application will let users pick what event they're attending and fill out all of their personal information (phone number, address, team, credit card). This application will let users provide credit card information to acquire a square on the bingo board through a donation to the client's Boo Radley Foundation. Once the Cow Chip activity is finished the admin will be able to select the winning square and users will be notified via text message.

2.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Functional Requirements:

- Ability to log in via Facebook, Google or as guest

Fit Criteria: The user shall be able to login via any of the above options 100% and within 3 seconds shall be redirected to the webapp main page.

- System shall verify credit cards before payment is verified

Fit Criteria: The credit card API shall be able to verify credit cards within 5 seconds of user entering their information.

- Users shall be able to select multiple squares from a 6 x 6 grid

Fit Criteria: The user shall be able to select multiple squares and it be registered in their selected square section within 3 seconds. The user shall be able to select multiple squares 100% of the time if they choose to do so.

- Admin shall be able to select winning square

Fit Criteria: The admin shall be able to select the winning square and have a text blast sent within 1 minute to all of the users with the winning squares.

- Admin shall be able to add users, ban users, and remove users

Fit Criteria: The admin shall be able to complete these functionalities 100% of the time and see it updated in their management portal within 10 seconds of completing above actions.

- Admin shall be able to view a list of users using filters

Fit Criteria: The admin shall be able to view their management 100% of the time and have it be updated to their respective filters within 5 seconds.

- System shall store square purchases for users

Fit Criteria: The system shall updated the users respective squares within 10 seconds.
- Multiple users can purchase the same square

Fit Criteria: The user shall be able to purchase the same square and it be updated in the portal with the amount of squares they own within 10 seconds.
- System shall store school selections for user purchases

Fit Criteria: The system shall store the school selections for the users and update every time a user selects a school and be shown in the management portal within 10 seconds.
- System shall be capable of aggregating and filtering purchase data

Fit Criteria: The management portal shall be updated with new data every 10 minutes and the admin will be able to filter the purchase data within 10 seconds.
- System shall notify winners

Fit Criteria: After the admin chooses the winners the user shall be notified they won via text in 1 minute.

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)
5. The entire credit card transaction information will be encrypted (security)
6. At least 1,000 users will be able to use the web app as it is designed to work (reliability)
7. The user shall be able to login through an existing Facebook or Google account in less than 15 seconds (performance)
8. The web app shall be able to calculate the entire cost of the donation once the user is “checking-out” within 3 seconds (performance)
9. The winning user shall receive a notification within 12 seconds from when the admin selects the winning square (performance)
10. The web application will be compliant with all policies and regulations set forth by the Boo Radley Foundation (operational)

2.3 CONSTRAINTS CONSIDERATIONS

One of the constraints that we will have is our budget. Our client wants us to make this project scalable because it will be taken over once we are done and he will try expanding the reach of the users. We have to be cautious of our budget because our client is paying out of pocket for this from his foundation. Another non-technical requirement is the reliability of the product during the main uses of it which will be during football games. The constraint for this is that wireless networks are often worse during college football game days therefore our server reliability will be integral to the success of the project. The last main non-technical requirement will be the usability of our page, our client wants to have the quickest way to be able to donate and not confuse the clients.

The standard protocols the we will follow while developing our project will follow agile protocol. We have multiple members of our team who are agile certified therefore we will use their expertise to maintain the agile process. The agile process is taught in school in many of our classes therefore they meet IEEE and ABET criteria.

2.4 PREVIOUS WORK AND LITERATURE

The product we are developing is the second software product that has been commissioned by our client's organization. The first software product is a mobile application currently in development by a team at Florida State University. The client rolled out this application in early November of this year. Our web application will follow the same narrative of the mobile application in order to promote the CowChips4Charity event. In addition, a link to our web application will be embedded in the mobile application. The Florida team has a prototype of their game available which will use the same color scheme as our application and have the same general advertising conditions as well. There is also currently the website which our client has and is fully functioning and developed. The site has some of the functional requirements that are necessary for our project as well. Therefore, we have been talking with the web developer who created this website and will also be taking our website over once our final product is delivered. Therefore, we will be basing our application of previous work that the client has already done and making the design off of said work.

In regards to the technicalities of our application and previous work like this we have a fairly typical single page application. Therefore, we are following business best practices for making a single page application. Going forward we are also using API's that are business best practices such as facebook for logging in which many other charities and organizations use to create a simple and easy login for users. Using API's like this will help with security aspect and make our design easier in that regard. It is also better in terms of scheduling because we don't have to independently develop all of that software.

All of us are also familiar with the software that we are using which is vue.js and node server on that backend. The application we are making is a web application with functionalities of fairly common web applications like: logging in, creating an account, payment option, small game functionality. Therefore, we believe that past class material and are knowledge of using the software that is being used can be considered as previous work and literature. More literature can be found on vue.js website about different applications and then also for node there are several different types of documents on their websites. Going along with node there are also many different NPM packages that have documentation which is also previous work and literature that we can reference for our own project.

2.5 PROPOSED DESIGN

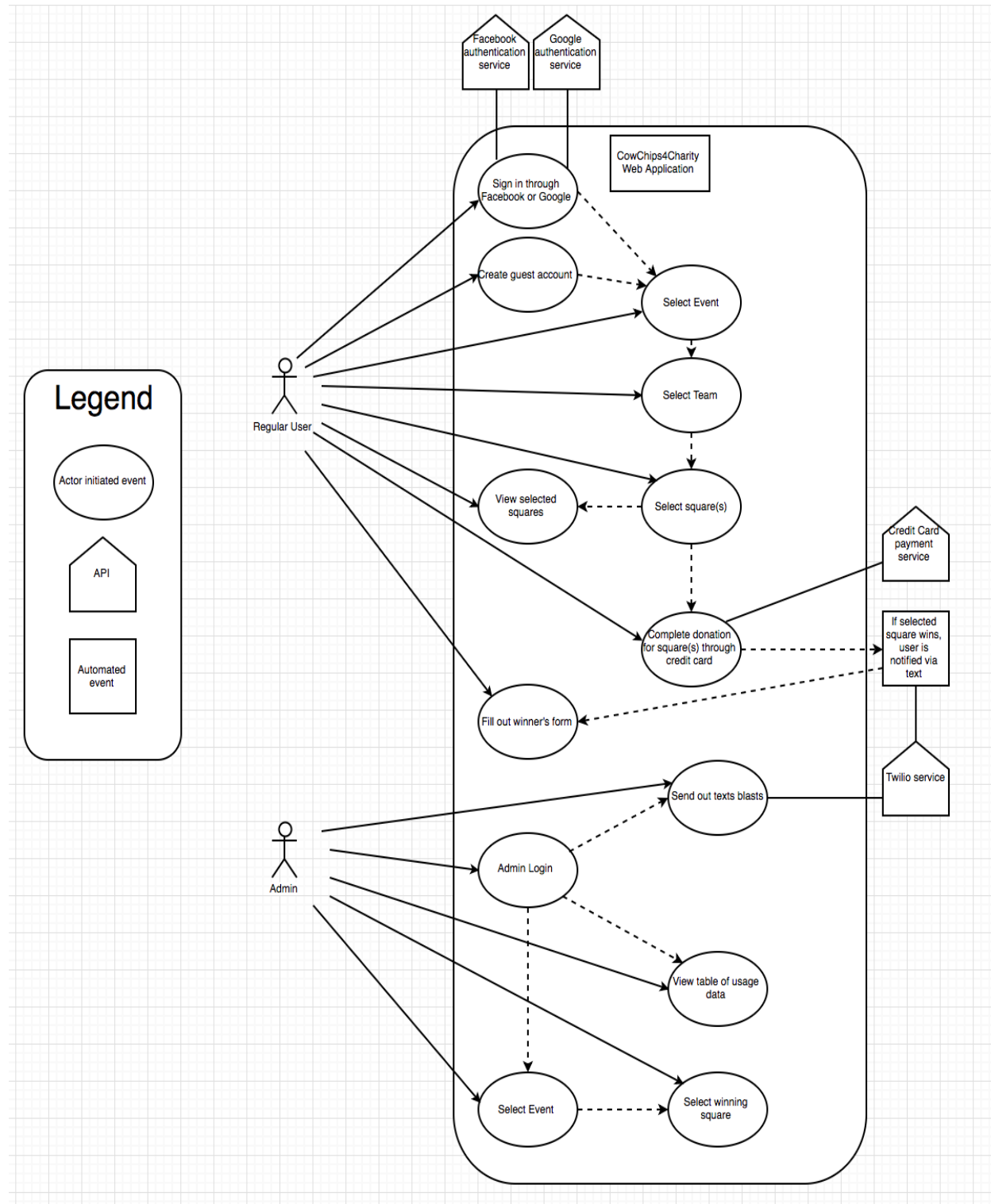


Figure 1: Proposed Use Case Diagram

We decided to use a Use Case Diagram to show how our users would interact with the application. In the diagram (Figure 1) it can be seen that there are two types of users that will have separate interactions. The regular users are people who will actually be playing the game, while admin users are the employees of the non-profit actually administering the game. Those users will initiate actions which will either trigger further actions (dotted arrows), or will trigger an automated action (house like pentagon shape) that initiates an external action such as text message blast. The different components of the application are the adjacent system and users that are a part of the application. We have many different components of the application the first being the User component which has all of the functionalities listed in the Use Case diagram. Another main component is the Admin which has different functionalities and also access to the admin database which gives information regarding the Users and donations. The last major component is the API's that are being used these are all individual components for example one API is the login through facebook which uses the facebook API. We also leverage an API to validate credit card and other online payment options so there are a variety of ways for users to be able to donate.

Table 1: Use Case Diagram Table - Use Case and Actors (Continued on next page)

Use Case	Actor(s)	Input and Output
Sign in through Facebook or Google	User	The user will input their credentials through a redirect from the respective social media site. The output will be the users credentials to the webapp so they can sign in and have their info synced.
Create Guest account	User	The user will be able to create a guest account which takes input as their credentials. The output will be the webapp storing their information and getting a successful response from the database.
Select Event	User/Admin	The input will be the event they want to select and donate for. The output will be the confirmation of the event selected.
Select Team	User	The input will be the team they want to select and donate for. The output will be the confirmation of the team selected.

Use Case	Actor(s)	Input and Output
Select Squares	User	The user will be able to select squares on the bingo board they want to choose and be able to donate to. The output will be that the squares they select can be viewed in their selected squares section.
Admin Login	Admin	The admin will be able to login using their credentials. The output will be a redirect to the admin main page.
Send out Text Blasts	Admin	The admin will be able to send text blasts to a select demographic of users. This can be by event or by team or all of the signed up members. The output of this will be a text is sent out to the respective users.
View table usage data	Admin	The admin will be able to see usage of different things. The different things can be something like donations per team, donations per event, how many users have signed up. The request will be what the admin wants to view and the output will be fulfilling that request.
Select winning square	Admin	The admin will be able to select the winning square to the event. The output is going to be users who won recieving texts to fill out information so they can receive their prizes.

Table 2: Use Case Diagram Table - Adjacent System / Components

Adjacent System/Components	Description
Social Media authentication	This will be social media platforms like facebook,twitter, etc. The user will be able to login with their respective accounts and not have to make their account or remember their information. These API's will be connected to our web application.
Credit card payment service	We will have an API connected that is a secure source, so a user will be able to enter their credentials and pay for their selected squares.
Twilio	The twilio service will send texts to users that the admin wishes to send texts to. This will sometimes include another link where the user can enter their info for their prizes.
User	The user will be able to do all of the main functionalities of the application that are documented in the requirements section of the project Plan.
Admin	The Admin will be able to do all of the main functionalities of the application that are documented in the requirements section of the project Plan.

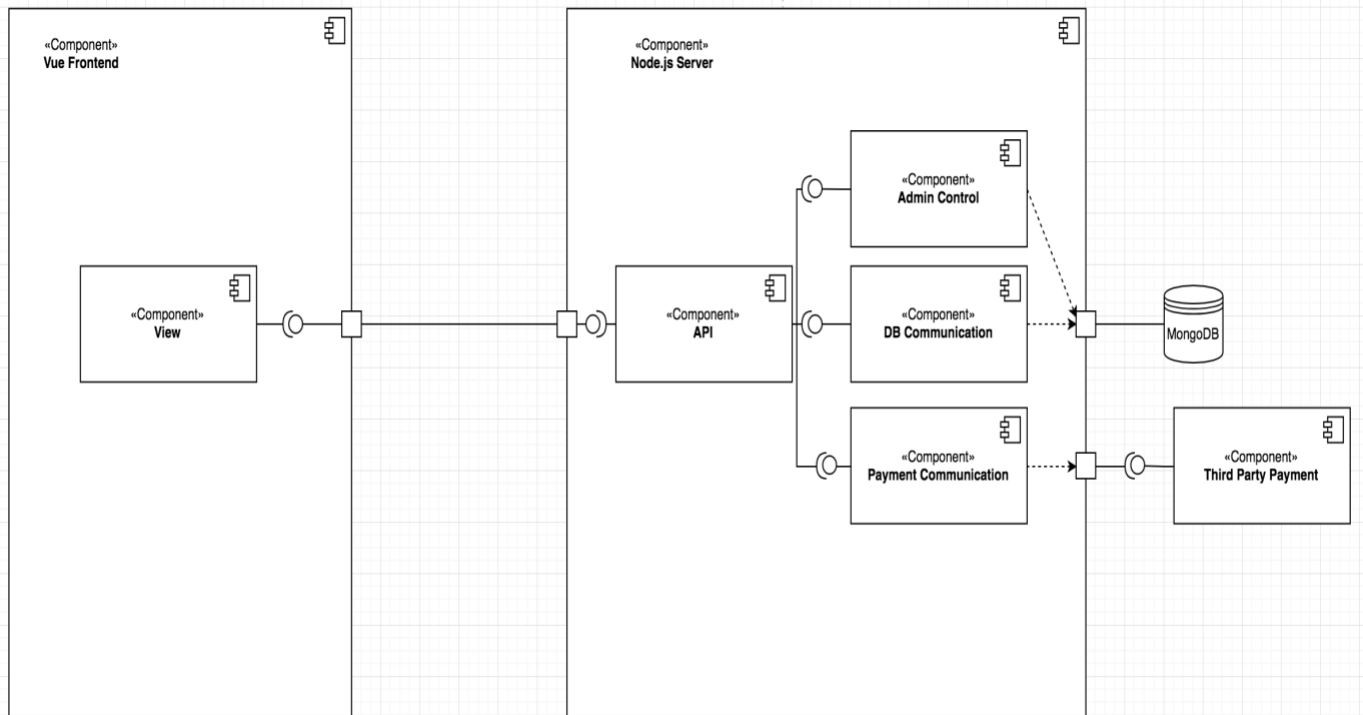


Figure 2: Proposed Component Diagram

Description of Component:

Admin Control component

The admin will have different functionalities and features than a regular User. The admin will leverage a Node JS server to accomplish all of the admin functionalities. The admin component will also communicate with the MongoDB. The reason the admin component will communicate with the MongoDB server is to be able to view usage data. The Admin component will also communicate with the database to select the winning user and notify them that they have won.

View component

The view component will house much of the frontend aspects of our project including the entire user interface of our web app. We will leverage vue and vuetify for all of our frontend design components including all for the User and Admin to ensure that we have a consistent design for all features and aspects of our project.

Payment Communication component

The payment communication component will be used by Users to confirm their donation as well as to purchase squares to play the Cow Chips Bingo game. The payment communication component will communicate with the third party payment component including stripe to validate all payments and ensure a secure transaction.

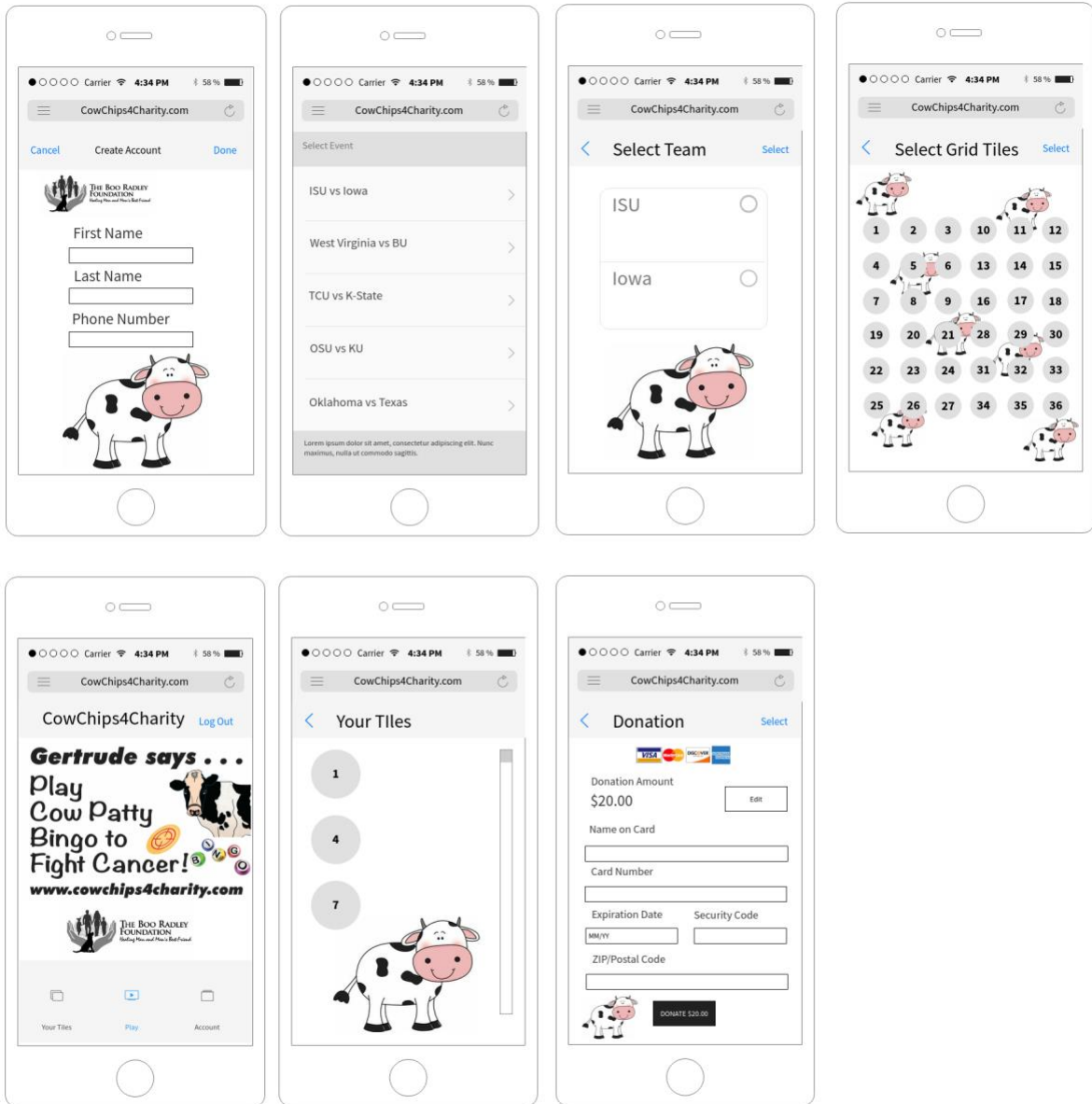


Figure 3: Wire Frames based off Use Case Diagram (Regular User)

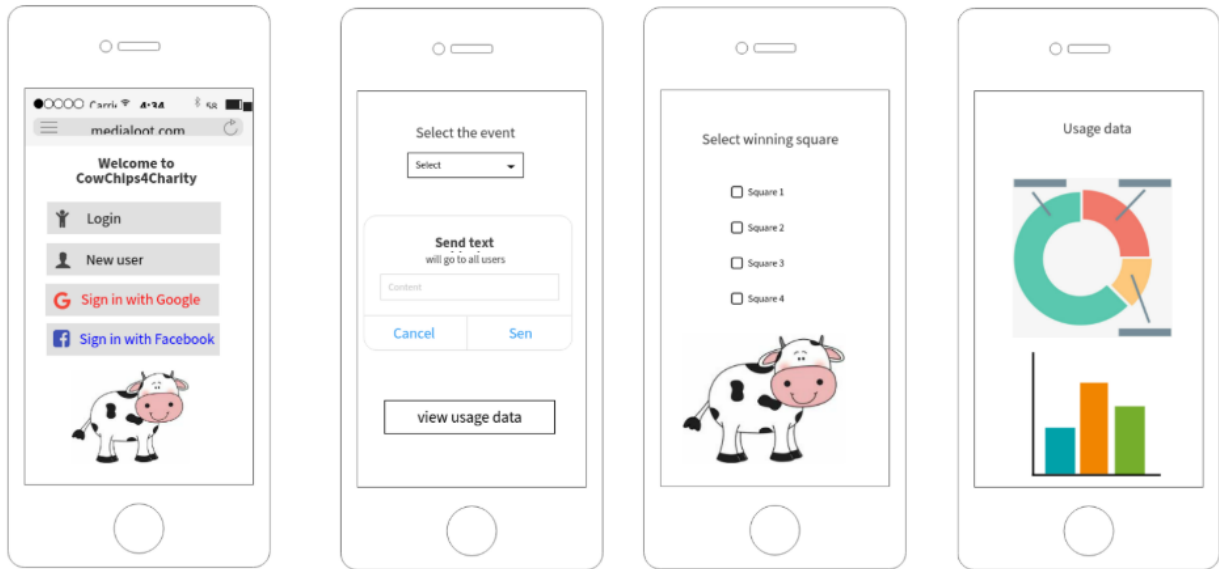


Figure 4: Wire Frames based off Use Case Diagram (Admin User)

Assessment:

When considering the design we could use a different type of architecture for example we could have a microservice architecture based off of Java Spring framework. Although we believe any other architecture would be over complicating the scope and requirements of our project. During server selection we considered three different hosting options: Heroku, Amazon Web Service, and MongoDB. The team and our client chose to move forward with MongoDB due to its competitive pricing options and our team's experience with the technology. This process is used because we have used it in other projects learning from our prerequisites from other courses. We have also taken into consideration best industry practices which we applied when making the design of this project. Our design also reflects the type of project management and implementation we will be doing which is going to be agile. The requirements and screens are separated so we can execute them through different types of sprints. After talking with our client other resources (i.e. professor, and advisor) we decided these design documents and process would best fit our project. We extensively focused on all of the main functionalities because that is our clients main concern. Our client isn't as concerned about a lot of the non functional requirements, but wants the application to be simple and to have all of the main things working reliably.

One thing we really wanted to focus is being able to utilize outside API's the reason for this is because it is best practice when making software in industry. We are utilizing API's also because this will save us time in development and testing. Also we will be transitioning this project when we are finished and the API's that we are using are continuously updated therefore the person who we transition will not have to worry about updating their application because the API will asynchronously do that for them. Some of the API's we are using are the facebook API for logging in, not only does this save the user time but it also is beneficial in terms of non-functional

requirements for security purposes. The same thing goes for our Stripe API which we are using for card processing, therefore our client isn't liable for having stand alone code that does credit card processing the API (Stripe) is liable and if anything goes wrong they will help solve the issue for our client. Another API that we used is twilio, this is for sending texts to our users, the reason we utilized this instead of making our own code is because for scalability and reliability purposes the testing is already done for us and conclusive that it works.

Process details:

- Are design consists of two actors (regular user and admin)
- We leverage several APIs for logging in including Facebook and Gmail
- Our design is broken up into different events where each event/game has different entities
- To pay and donate we leverage a credit card payment service
- All usage information and events are broken down for each specific game
- Regular users are able to view their overall history and their donations for each specific game

2.6 TECHNOLOGY CONSIDERATIONS

The reason our team chose the technology that is selected is because we have presented all of the possible options for technology that could be used and discussed which one should be chosen. The criteria we focused on was usability and reliability because our client identified these as his main concerns. After finding out the main constraints for technology and the priorities we were able to do research and figure out industry best practice for each of possible technologies. The only weakness of some technology that was selected is that a lot of the team hasn't previously used them, but because the research we did the team concluded that these would yield the best results for our client. The biggest trade-off would be in terms of time because we will have to learn the new technologies. Our team hasn't identified any negative trade-offs in regards to functionality or any of our non technical requirements. The solutions we have selected are presented in the above section which demonstrate some of the design aspects.

2.7 SAFETY CONSIDERATIONS

Our team does not have many safety concerns due to having limited hardware components needed to use our project. The only safety concerns we have is for our users to be operating machinery or vehicles while using our project. One thing our team did note is that we want to keep all of our user information in a secure database and encrypt their passwords. We will also be using a financial services API that is trusted and secured to make sure we can minimize any loss of our user's financial information.

2.8 TASK APPROACH

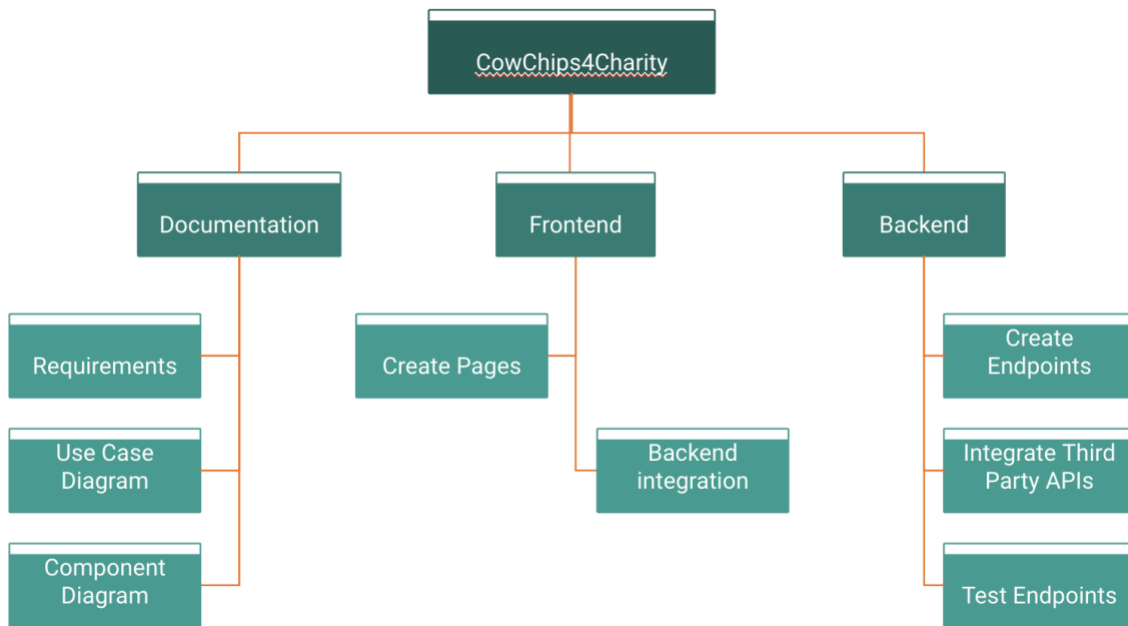


Figure 5: Work Breakdown Structure

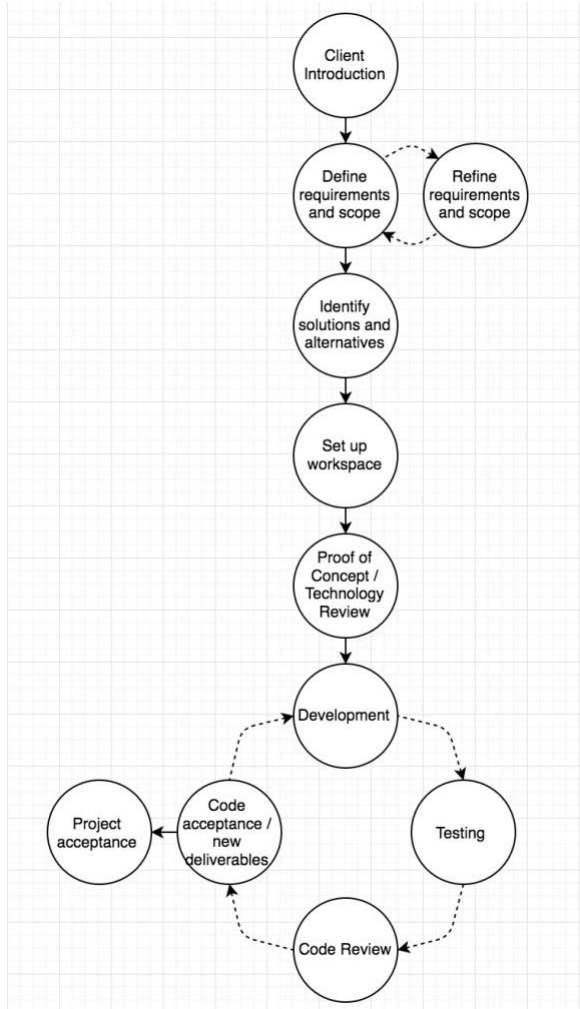


Figure 6: Project Process Diagram

1. **Client Introduction:**
 - a. Introduce client to the team
 - b. Learn more about the client and their organization
 - c. Talk about project background
 - d. Create member profiles for team website
2. **Define requirements and scope:**
 - a. Based on initial client meeting, define requirements and scope with team
 - b. Generate initial use case document
3. **Refine requirements and scope:**
 - a. Meet with client and discuss requirements and scope document
 - b. Confirm use case document with client
4. **Identify solutions and alternatives:**
 - a. Identify server pricing for client
 - b. Generate communication diagram
 - c. Generate component diagram
 - d. Identify APIs
5. **Set up workspace:**

- a. Database Initialization
- b. Node.js server backbone
- 6. Proof of concept / technology review:**
 - a. Database connection
 - b. Testing of Facebook Auth API
 - c. Testing of Google Auth API
 - d. Testing of Twilio API
- 7. Development:**
 - a. Begin frontend and backend development
 - b. Establish CI CD Pipeline
- 8. Testing:**
 - a. Individuals test their own code
 - b. Code is pushed and automatically tested against every test the team has written
- 9. Code Review:**
 - a. Team member reviews project code
- 10. Code acceptance / new deliverables:**
 - a. Team accepts code after it is reviewed and establishes new deliverables that may be needed for project progression
- 11. Project acceptance:**
 - a. Team delivers final project to the client
 - b. Client accepts project and makes the product live

2.9 POSSIBLE RISKS AND RISK MANAGEMENT

Some concerns that the team has identified regarding our project process and development plan include:

- Project transfer after our team is rolled off of the engagement in May 2019
- Privacy laws that may prevent user data storage in some states
- Service reliability caused by network lag during sporting events
- Server host pricing as web application expands and matures
- Learning curve when utilizing new APIs
- Instructing admin users on how to execute admin capabilities

2.10 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

Our main key milestone is to have all of our requirements and scope satisfied with enough time to make full adjustments to the application. We want to be able to provide our client with an application early into 2019 which will be version 1.0 so we can see how everything works technically. Once we have that done, we will begin work on version 2.0 which will mainly prioritize making sure the UI and UX are exceptional. The last deliverable we will have is compiling all of our documentation and have transition meetings with the team that will be taking the project over after we are done working with it. For a more accurate look of the proposed schedule look at the gantt chart which is referenced in the appendix.

2.11 PROJECT TRACKING PROCEDURES

We are using multiple project management tools these tools include: Trello (task list), gitlab (feedback and milestones), and github (code review). Our team will update all of these and the Project Manager will make sure they're kept up to date and that all team members are fulfilling their responsibilities.

2.12 EXPECTED RESULTS AND VALIDATION

Our desired outcome is to have a fully working product that meets all of our functional and non technical requirements. We believe that if we meet all of our requirements and scope which have been refined by our team multiple times and recently approved by our client the product will be satisfactory. The client has requested a web application that will be able to let users donate funds and pick a CowChipBingo square during football games (both collegiate and professional). The application will be able to let users know if they have won and notify them of how they will be receiving their prizes. The application will also allow an admin user to select when each games will start and end as well as deciding the winner of each respective game. Lastly, the admin will be able to view usage data and amount of funds donated for each game. We want all of this process to be completely automated and also scalable to large masses of people (~1,000 users). The last thing we are considering is making sure our documentation throughout our development process is very detailed because we will be transitioning to another team once our Senior Design course comes to an end. Our team has prioritized documentation due the needing to transition our project to a our client after completion of the Senior Design course. Our client is not well versed in technology and development that is why it is crucial we create documentation that ensures a successful transition process.

Standards

In terms of standards when choosing the final design, our team placed a high emphasis on performance, scalability, reliability, and security. As such our design incorporates load balancing to ensure high performance and scalability of the web application as the CowChips4Charity event becomes more popular. In addition, we chose to leverage the Stripe API for payments as it is better suited for handling credit card data. The security features of Stripe far outway what we will be able to develop in the given timeframe and we will not need to needlessly save sensitive financial information on our servers.

A major concern of our client was also being able to transition the code to another team which he plans on doing once we are done. The version of our project is the first phase of what our client wants to get done there will be a lot more evolution that needs to happen. As referenced above by utilizing all of these API's evolution of our software will be a lot easier because these services are continuously updated automatically for us and will evolve as technology evolves. The next standard we want to have addressed is scalability because we will be having large amounts of users during the use of our application. This is also addressed by using heroku to automatically scale for us and let our users have reliable service no matter how many are using it simultaneously.

2.13 TEST PLAN

Our teams plan is to have our project broken down into different deliverables. We will have each individual developer test the code that they have written. All initial testing will be completed by the owner of the deliverable no later than 10 days after completing that specific deliverable. Once the developer has tested their code they will also have another team member review their code and once that is complete we will add it to our repository. When adding the code to the repository we will include a summary of the testing that was done on that deliverable and members that were involved with testing. At the end of each deliverable we will have System Integration Testing where we will go and test the web app to make sure it satisfies all of the requirements and use cases we have defined for said deliverable. After SIT we will have User Acceptance Testing by having our client test the system and make sure it is up to his standards and that all of the requirements and scope of the deliverable is fulfilled.

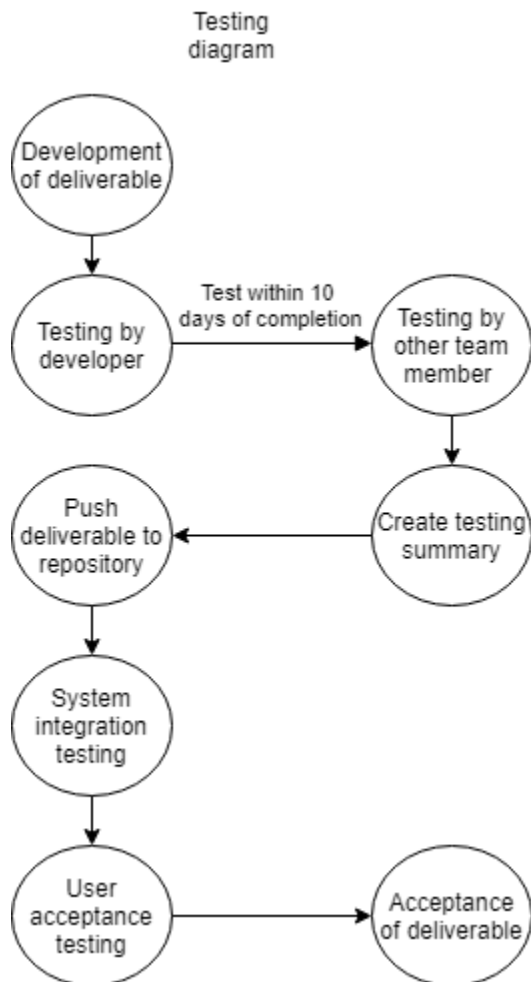


Figure 7: Diagram of Testing Procedure

3. Project Timeline, Estimated Resources, and Challenges

3.1 PROJECT TIMELINE

Over the course of the first semester our team must complete a variety of objectives. First we must communicate with the client. Over the course of the next few weeks we will obtain the requirements, refine the requirements, and approval of the refined requirements. Next we will brainstorm possible solutions, research technologies, and decide which technologies we will use. We will then create project documentation. This documentation will include: Use case diagram, component diagrams, as well as communication diagrams and wireframes. While this is happening we will set up our repository, the database, and the development environment. Finally, we will begin development of our project. The development that will be completed in the first semester is as follows: database communication, the log-in page, the donation page, and the supporting backend API endpoints.

During the second semester, we plan on continuing our development of the frontend and backend development. The frontend and backend will be developed in tandem to allow for maximum efficiency and quick completion. During this semester, we will finish the Admin page, tile selection pages, and the support backend API endpoints. After all functionalities have been developed, we will ensure that our application is safe and secure and add any necessary countermeasures that are not already present in order to secure the app. We plan on performing full system integration testing and finishing the project during the first week of March.

Please see appendix item number 1 for the Gantt chart.

3.2 FEASIBILITY ASSESSMENT

We expect to have a fully functional, production ready, auto scaling, load balanced, web application by the end of second semester. Our system will employ the use of a frontend backend architecture allowing our team to make changes to either system without the need to redeploy the other. We expect to have all of the user experience requirements met, making donations and signup simple and easy. We also expect to have a administration panel attached to the backend system that will allow for easy administration of the system by a system administrator.

The main unexpected challenges will really come as a result of having a team of programmers working with technologies and systems that they are unfamiliar with. This can have the impact of slowing down the development of new features, and would thus throw the project off schedule. The next major impact that could come from our lack of knowledge would be the possibility of major server outages as a result of deployments. Any outage of the system, especially during a major event, would spell disaster for the project. We hope to eliminate these risk through the use of automation to prevent any critical faults ending up in production that could lead to a crash of the servers.

3.3 PERSONNEL EFFORT REQUIREMENTS

Table 3: Personal Effort Tasks and Description (Continued on Next Two Pages)

Task	Effort	Description
CD/CI Setup Frontend	8	This initial setup of the CI/CD to a live server carries significant unknowns especially with the team not having used Jenkins or Heroku before. Therefore we think that the complexity and ranking of this item should be higher than most other tasks
CD/CI Setup Backend	8	This initial setup of the CI/CD to a live server carries significant unknowns especially with the team not having used Jenkins or Heroku before. Therefore we think that the complexity and ranking of this item should be higher than most other tasks
User authentication API	5	The user authentication API implementation and design will be a key functionality of our project this item requires significant thought to functionality and efficiency.
User authentication UI	5	The user authentication UI is th entry point for the user into the application, thus it is extremely important that this item is completely successfully and that the UI components are highly performant and look appealing to the users.
Selection API	5	The selection API is vital to the project as it is the main interaction that the user is actually using the application for, this API needs to be extremely reliable as well as secure to prevent unauthorized selections.
Selection UI	5	Again because this is the main interaction that the user is using the app for it is extremely importan that the components are responsive and more importantly accurate to prevent mistakes and errors to the user that can cause inaccurate payment issues

Task	Effort	Description
Stripe Integration	3	The stripe integration is the main payment service we will connect to to process credit card transactions. Because this is a well documented and designed API connection we don't expect much complexity or unknowns when dealing with it.
Payments API	5	The payments API will be of the utmost importance in terms of security. This API is vital to the application as it literally is the butter that will make the entire application work.
Payments UI	5	The payments UI will again need to be highly responsive to give the user a experience, because if the sale is lost at this point then the rest of the application is irrelevant
Administration panel UI	8	The administration panel will have the largest number of features and data access points. Thus the api will be larger than all of the other API's
Administration panel API	8	The administration panel UI will be more complex than any other screens requiring CRUD operations and tables.
Twilio Integration	3	The twilio integration is again a well documented and simply designed API thus making this integration less complex
Production Servers Setup	5	The setup of the larger production servers will require the accumulated knowledge that the team has taken from server deployment up to the project thus far and will focus that knowledge to make a much larger and more reliable system.

Task	Effort	Description
CD/CI Prod setup	8	The CD/CI setup of the production servers needs to be highly reliable, and complex to catch any errors that appear during the deployment process and work to mitigate and prevent any regressions.
Scale Production	8	The scaling of production servers will be a nerve wracking experience as it will be hard to test and measure high loads on the servers. We think that this step will be complex in the diagnostics and data.

The scaling of complexity we decided upon is that of the SCRUM fibonacci sequence which ranks items on their relative complexities to each other. This method doesn't provide an exact amount of time that it takes to complete a requirement thus giving us more flexibility. Additionally, since the scaling is relative to each of the other requirements that allows us several advantages:

1. The completion of one task will give a good estimate for the approximate completion time of remaining tasks.
2. The scaling of items can be changed as we move through the project to provide a more realistic expectation.

3.4 OTHER RESOURCE REQUIREMENTS

We have no additional resources outside of our own laptops and experience to complete this project.

3.5 FINANCIAL REQUIREMENTS

The main financial resources needed to complete this project are fairly minimal during the development cycle with our team only needing one server to host both our frontend and backend. That server is currently priced at \$7 per month. Additionally we will be using a database hosting service called mlab to host our MongoDB database which will charge \$15 per GB which we do not expect originally to come even close to, yet we will budget \$5 per month for that database.

Once the system moves to production, the costs go up significantly due to the introduction of auto scaling and much larger servers to handle large amounts of user traffic.

4. Closure Materials

4.1 CONCLUSION

Our goal for this project is to assist our client and the Boo Radley Foundation in creating a technological solution for the CowChips4Charity initiative. To fulfill our solution we will be creating an interactive web application where users can virtually buy squares and play the CowChips4Charity game. Our solution will not only create an easy means of intaking, but also retaining and engaging supports for the Boo Radley Foundation. Our web application will also create an easy and organized means for the admin to have access to the game and properly select the user. A goal our team has is to make sure that our solution is sustainable and easy to maintain for our client after we are done with our scope of work. This project is intended to increase outreach and participation in both the CowChips4Charity event as well as other Boo Radley Foundation events and initiatives. Our team is excited to help a non-profit organization grow and help make the world a better place.

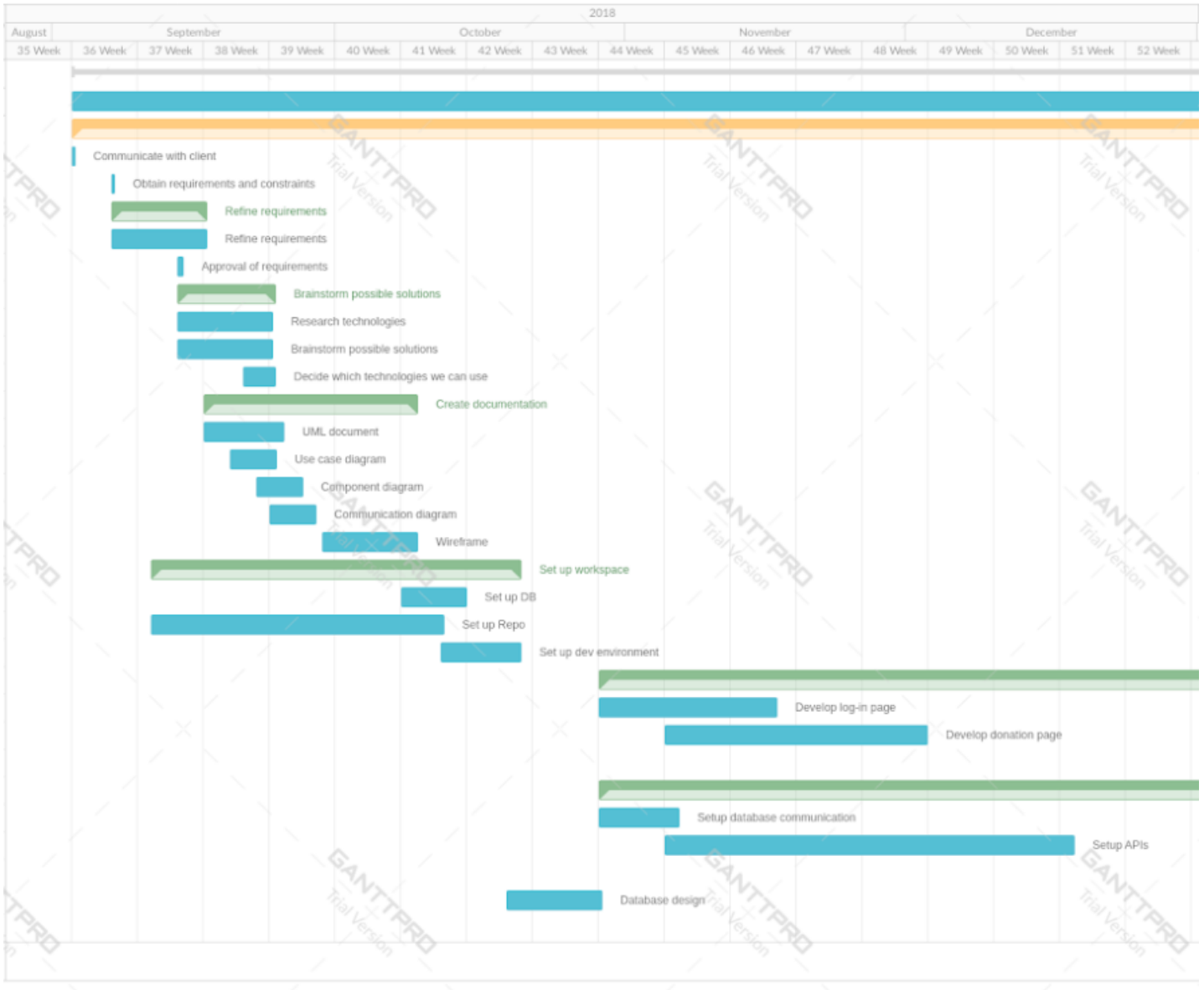
4.2 REFERENCES

To understand our project statement, our team participated in several meetings with our client to make sure we understand the statement of work and create the proper goals and initiatives. Our client provided us with several resources to develop an understanding of the CowChips4 Charity (<http://cowchips4charity.com/>) initiative and the Boo Radley Foundation (<http://booradleyfoundation.org/>). We also took reference from other nonprofits to understand how they create outreach and increase sponsorship. These references come from the The Make-A-Wish foundation (<http://wish.org>) and from the Special Olympics event of Polar bear plunge (<http://soiowa.org>).

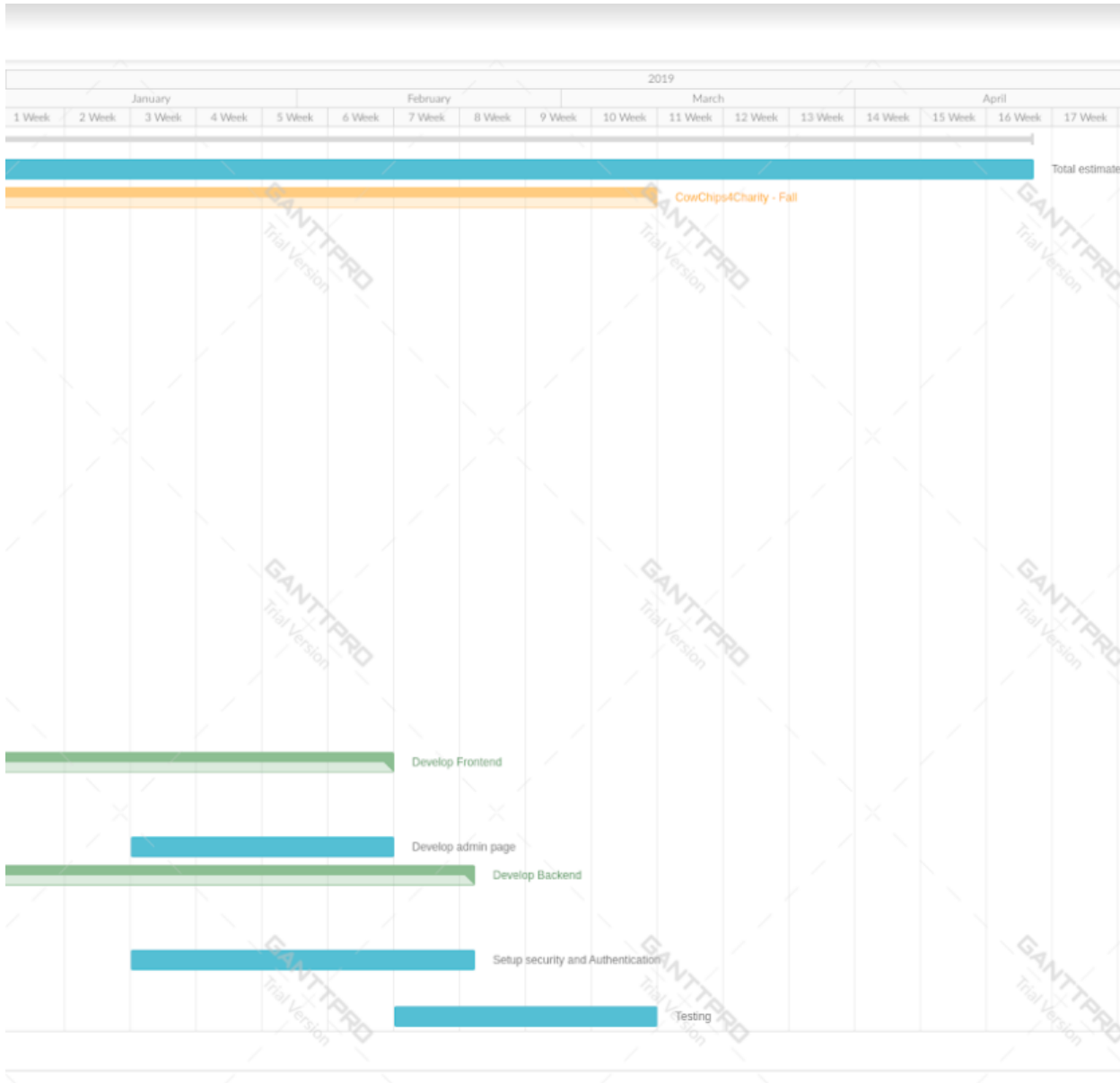
4.3 APPENDICES

Item 1: Project timeline / Gantt chart (chart on the right is first semester, chart on the left is for spring semester)

1st Semester



2nd Semester



Cited Sources:

“cowchips4charity.” *Cow Chips 4 Charity*, The Boo Radley Foundation, 2018, cowchips4charity.com/.

“Facebook for Developers.” *Facebook for Developers*, Facebook, 2018, developers.facebook.com/.

Johnson, Ken. “The Boo Radley Foundation - Healing Man and Man's Best Friend.” *The Boo Radley Foundation - Healing Man and Man's Best Friend*, The Boo Radley Foundation, 2018, www.booradleyfoundation.org/.

Johnson, Ken. “Pre-Launch Demo (Final) by Booradleygames.” *Itch.io*, Boo Radley, 2018, booradleygames.itch.io/pre-launch-demo-final?secret=wl657EJQhf0Erg5sv7OALj0.