Gaucho OrganAppZation

TEAM SEGFAULT

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeffrey Liu</td>
<td><a href="mailto:junfeng_liu@ucsb.edu">junfeng_liu@ucsb.edu</a></td>
<td>Developer</td>
</tr>
<tr>
<td>Justin Tjoa</td>
<td><a href="mailto:justintjoa@ucsb.edu">justintjoa@ucsb.edu</a></td>
<td>Developer</td>
</tr>
<tr>
<td>Megh Shah</td>
<td><a href="mailto:meghshah@ucsb.edu">meghshah@ucsb.edu</a></td>
<td>Lead</td>
</tr>
<tr>
<td>Nathan Wu</td>
<td><a href="mailto:yangwu@ucsb.edu">yangwu@ucsb.edu</a></td>
<td>Developer</td>
</tr>
</tbody>
</table>

INTRODUCTION

Gaucho OrganAppZation is a web application to help incoming college students to select the club(s) that would best match them. Unlike similar club searching platforms, this recommender sorts relevant organizations by their ratings and allows for peer reviews. Critically, Gaucho OrganAppZation tailors the results to best fit the students’ unique interests and desires based on personalized queries.

PROJECT OVERVIEW

College freshmen struggle to discover which organizations or clubs to join, based on their interests, personality, and career aspirations. Current club-recommending technologies are unoptimized. For instance, these technologies do not rank organizations or clubs. And most prototype software to do such things take objective parameters to judge a subjective choice (like picking a club/organization) and do not take into account unique personality and corporate culture at certain clubs that students would prefer. These technologies fall short in terms of creating a diverse pool of statements that relevantly describe a club or organization, yet not overflood the users of this app with content. In short, relevancy and conciseness are an unseen combination for these apps.

Gaucho OrganAppZation is a web app that promises to bridge the gap between relevancy and conciseness for these students. The app splits clubs by category and narrows down students’ searches using objective parameters such as quantitative scores out of 5 stars to give a threshold of clubs that a student should go after. In order to aid the student in picking the club based on factors that objective ratings cannot give,
the app appropriately matches a subjective choice with subjective parameters, such as smartly and economically placed hashtags that shadow a club’s name, as well as certain keywords that give indication of certain properties of a club the students would like. Comments by other students who previously or currently are in the club will best tell students about factors of the club students may or may not enjoy. By using both objective and subjective parameters, club recommender accurately picks organizations that a student not only will succeed in, but will take pleasure in and become a part of its unique culture.

In conclusion, Gaucho OrganAppZation helps indecisive students - based on his or her answers to personality questions, the app can spontaneously give a club/organization choice of “best fit.”

**REQUIREMENTS**

**User Story (#1) :** As a user, I can search for a club based on its name.  
**Estimate:** 10 days  
**Acceptance test:** Scenario 1 -  
Name of the club is present in our database.  
The app looks for the name in the database and displays its details on the screen.  
**Tasks:** https://github.com/NathanWoo/CS48/projects/1#card-20102461

**User Story (#2):** As a student, I can search for clubs based on a keyword so that I can get all the clubs related to a specific topic.  
**Estimate:** 3 days  
**Acceptance test:** Scenario 1-  
Keyword is invalid like "qwerty"  
There are no clubs related to the keyword "qwerty"  
The web app informs the user club doesn’t exist.  
**Tasks:** https://github.com/NathanWoo/CS48/projects/1#card-20415149

**User Story (#3):** As the software developer, I will make a firebase to store our clubs and profiles.  
**Estimate:** 1 days
Acceptance test:
1. Can see that our firebase has a clear url
2. Can manually add nodes to the tree in firebase

Tasks: https://github.com/NathanWoo/CS48/projects/1#card-20415063

User Story (#4): As a user, I can review all categories of UCSB clubs, and after I click in a specific category, the webpage will list all clubs in this category with the rating score higher than 3.

Estimate: 4 days

Acceptance test:
1. In a single webpage, user can find clubs by clicking buttons for different categories of clubs listed on the webpage.
2. After click on some specific button, the webpage will list all clubs belonging to this category with a score higher than 3 (full score is 5).

Tasks: https://github.com/NathanWoo/CS48/projects/1#card-21249672

User Story (#5): As the software developer, I can connect the firebase to working Node.JS code so that I can maker user profiles.

Estimate: 4 Days

Acceptance test:
1. Connecting to the database with the correct API key and url should print out “connected”.
2. Putting the wrong API key and URL should print out “not connected, exiting”.

Tasks: https://github.com/NathanWoo/CS48/projects/1#card-20415182

User Story (#6): As the software developer, I can add and delete values in the firebase so that I can add new users or clubs.

Estimate: 2 days

Acceptance test:
1. Sending a request via a function called addclub() should be able to add a club with the following attributes: Name, keyword, and rating, all of which are provided.
2. Attempting to add a club with one or more of these attributes should lead to the addition of a club with only the indicated values.
3. Attempting to add a club with no values shouldn’t add a club at all.

Tasks: https://github.com/NathanWoo/CS48/projects/1#card-20415217
**User Story (#7):** As the software developer, I can request to read certain values by sending the node’s identifier

**Estimate:** 3 days

**Acceptance Test:**
1. Sending the name of a node should successfully fetch its description.
2. Attempting to send an inquiry for a node that doesn’t exist should print “node doesn’t exist”

**Tasks:** [https://github.com/NathanWoo/CS48/projects/1#card-20415149](https://github.com/NathanWoo/CS48/projects/1#card-20415149)

**User Story (#8):** As a student, I can click a button so I can see a full list of all the clubs on UCSB campus.

**Estimate:** 2 days

**Acceptance Test:**
1. If I click the Cubs List button on the home page, I will be directed to another webpage that shows me all the clubs on UCSB campus.
2. If I don’t click the Club List button, I will stay at the home page of the UCSB Club Recommender

**Tasks:** [https://github.com/NathanWoo/CS48/projects/1#card-22700678](https://github.com/NathanWoo/CS48/projects/1#card-22700678)

**User Story (#9):** As a student, I can click a sort button so I will be able to see a list of ranked clubs based on the rating scores each club has

**Estimate:** 3 days

**Acceptance Test:**
1. Once I click the [Rank By Rating] Button I’ll see a sorted club list at the current web page.
2. If I choose to not click the button, I'll remain at the current page and browse freely what clubs are there at UCSB

**Task:** [https://github.com/NathanWoo/CS48/projects/1#card-22700999](https://github.com/NathanWoo/CS48/projects/1#card-22700999)

**User Story (#10):** As a UCSB student, I want to write a comment, rate a club or do both based on my experience at that club.

**Estimate:** 4 days

**Acceptance Test:**
1. If I only want to rate the club, I click the review button to give the club a rating on the 1-5 scale and click submit.

2. If I only want to give some comments to the club, I click the review button and write down some words no more than 140 words, then I click submit.

3. If I want to write a comment and give this club a rating, I will click the review button, write down some words that are not more than 140 words then rate the club on the scale of 1-5, at last, I'll click the submit button to upload my review.

4. If I didn't rate or write any comments to the club but still clicked the submit message, I'll get a message saying "please leave a comment or give a rating"

Task: [https://github.com/NathanWoo/CS48/projects/1#card-20203966](https://github.com/NathanWoo/CS48/projects/1#card-20203966)

**SPRINT RETROSPECTIVES**

**Sprint 1**

- **What worked?**
  - We successfully assigned specific roles to each member of our group. Everybody was aware of their tasks.
  - Each member individually learnt more and started implementing code in their respective area of work.
  - The initial UI of the project was ready and approved by all the members.

- **What should change?**
  - We should meet more often in a week and work together to avoid redundant work and increase efficiency.
  - Before deciding the use of a technology tool in our project, we should discuss it with all the team members.

**Sprint 2**

- **What worked?**
  - We successfully got the front-end and back-end people to understand the opposite technologies so that more efficient code was made.
  - Each member started to meet more often so that we knew what was going on.

- **What should change**
Discussions should focus more on how each other’s code works - we are conceptually going over the idea but actually walking through code line-by-line is far more effective than referencing Stack Overflow.

While we knew what was going on with each other at the present, a better job should be done also in discussing each of our future plans so that we can see the trajectory where our individual work is headed.

Sprint 3

- What worked?
  - Meeting more often led us to realize that our tasks were more demanding than we thought they were driving us to work harder.
  - We successfully completed our project and each member of the team effectively understood each other’s code.

- What should change
  - In the future, we would like to add more features to our project and take it to a grander scale.

PROJECT MILESTONES

- Build a user-friendly GUI
- Firebase - Build a firebase implementation that stores all UCSB clubs/organizations and works in conjunction with a recommendation system and comment system
- Frontend - search bar, category pick (genre of clubs), rating system interface

CHALLENGES FACED/OVERCOME

Since most of us never used Firebase, there was a huge learning curve so we had to look up Firebase documentation only to find that it was not uncommon for some documentation to have outdated methods. We had a lack of general knowledge about databases, which led to us having pulls toward different directions for database implementation including Firebase, mySQL, and PostgreSQL without knowing the advantages/disadvantages of each. In the beginning we at times had to throw out each other’s work due to lack of communication where we each did something different and rendered each other’s work obsolete.

We struggled with using our user stories as intended - initially we treated the creation of user stories as a means-to-an-end for points. The consequences of superficial user stories were clear in sprint 2 when we didn’t understand what each other was doing leading to redundant tasks. This also led us to often start a task that
would have a harder process later on for our specific application - for instance we switched from SQL to firebase when we learned Firebase was more ideal for our purpose.

We overcame our lack of Firebase experience via copious documentation study. We overcame our poor central planning by focusing more on user stories and consulting our mentor, Scott. At the end we were able to come up with a product that characterized our perseverance despite lack of implementation knowledge.

REMAINING FEATURES OR FUNCTIONALITY

At the time of this writing, we will implement a rating option for the club as well as an add-keyword function. In the future we will have a club recommendation system based on personality and best-fit parameters. Lastly, there will be an interface for adding and viewing subjective reviews for the clubs.

TESTING

Test file for the project:
front_end:
    selenium_test.side
test.side
test1.java

Backend:
test.js, which is inside test folder

The test files tested all functionalities that our website could achieve. The frontend test will automatically click all buttons in our website and check the performance. And the backend test tested all the functions of our firebase database.

In the frontend testing, the test files tested 3 edge cases our search should cover: searching the name of a club that exists club, searching for clubs via a keyword, and searching for a non-existent club in which the response “Club Not found” should be printed. All 3 cases use the same search bar (the website recognizes if the word is a keyword or a club name). The first edge case was tested by searching for a club in our
database titled “Coders SB”, “IEEE”, and etc. The webpage successfully returned us the club name, rating, related keywords, and club description. Our second test case was tested by searching for clubs with a series of keywords such as “Hacks”, “Acapella” and “Inclusive” in the search bar. The website returned the information of all clubs that had the keyword as if we individually searched each club in the first case. Finally, we tested the third edge case by putting in the club name “Apple Store” (a club that doesn’t exist in database), after which the webpage printed the expected - “Club Not Found”. Besides, we also type in some random text such as “xuidsafhdsif”, which also returns us “Club Not Found”. In the backend testing, we tested our backend functions “search by club name” and “search by keyword”, and we run our backend test on travis CI.

HIGH LEVEL DIAGRAM

![Diagram](image-url)
Actor

add club

send club proposal

check if club exists

boolean club exists = false

add club

confirm club creation

new club instance

confirm club creation
Welcome to UCSB

Get Involved! Now is the time to explore your interests and have fun at the same time. There are many opportunities for you to get involved.

How does this app work?

UCSB Club Recommender is a web application to help incoming college students to select the club(s) that would best match them. Unlike similar club searching platforms, this recommender sorts relevant organizations by their ratings and allows for peer reviews. Critically, UCSB Club Recommender tailors the results to best fit the students’ unique interests and desires based on personalized queries.

- **Search by Keyword**: Search for Clubs based on Keywords!
- **Review System**: Rate and Give Feedback for a club you have been a part of.
- **Best Fit Club**: Find the best suited club based on personality questions.
TOOLS

- Languages: JavaScript
- Web Technology: HTML, CSS, Firebase
- Testing: travis CI, Mocha, JavaScript, Java, Selenium
- Version Control: GitHub
- Project Management: Github Project, Google Drive
- Communication: Slack

GITHUB COMMITS

Link: https://github.com/NathanWoo/CS48/commits/master

Most Recent Commits by Feature:

Jeffrey Liu: Last Commit - use test.js for testing website
Update test.js

ljfjeffrey committed 5 hours ago

Justin Tjoa: Last Commit - allowing single clicks to buttons, integrate front/back end
single click and frontend-backend integration

justintjoa committed 24 days ago

Megh Shah: Last Commit - changed images on website
Merge pull request #4 from NathanWoo/nathan2 ...

megh98 committed 16 days ago

Nathan Woo: Last Commit - polished CSS stylings
Final Push

NathanWoo committed 16 days ago
DEMO VIDEO

Link: https://www.youtube.com/watch?v=0lVOQ2fVCn4&feature=youtu.be

IMPORTANT LINKS

Github: https://github.com/NathanWoo/CS48

Travis-ci: https://travis-ci.org/NathanWoo/CS48

Slack: cs48projectgroup.slack.com
Burndown Charts

Sprint 1

Sprint 2