Clava

Team 11 - Product Backlog

Hunton, Alex
Lee, Christopher
Leungwattanakij, Kris
Pan, Leonard
Tinkess, Kai

Problem Statement

At this time, most college clubs and organizations do not have a centralized system in which general information, member data, and funds are managed. This directly impacts their ability to effectively operate by increasing the amount of time it takes to complete basic tasks, such as granting individual permissions or providing reimbursements.

Clava will provide college organizations with a suite of tools to manage their members, expenses, and documentation. The user interface will be designed with the clubs in mind, with feedback directly pulled from Purdue organizations like the Boiler Book Club. Due to all of these tools being centralized, tasks like managing member permissions for documentation and finances will be streamlined, and organizational information will be much easier to view.

Background Information

The majority of college clubs and organizations use some combination of Google Suite, Boiler Link, and other tools such as SquareSpace or Wix. Specifically, the areas where the use of these existing systems fall short is the decentralization of information and overall unfriendly user interface. Separating member data, financials, upcoming events, board documentation, and other important aspects of administration makes accessing and understanding the general state of the organization difficult, especially for new board members. Most clubs will spend valuable time struggling to create the ideal centralized system in order to streamline their management process. We desire our solution to be the direct answer to this need.

A common tool used by Purdue-specific organizations is BoilerLink, which is a platform known for not being user-friendly. Frequently needed tools such as APFs and Proposals are hidden behind many unintuitive menus, and there is little customizability in the member list. More conceptually, BoilerLink's primary customer is the university, and the student experience is often lacking. Its true functionality is to ensure that Purdue organizations meet the requirements that the university sets out for clubs. We hope to fix this specific issue by building a straightforward user interface with students in mind using the React framework. Rather than failing to utilize the impractical BoilerLink toolset, members of college organizations can instead choose to use our solution. We also don’t limit our users to Purdue students – anyone that attends universities which offer similarly flawed tools are welcome.

Presently, if an organization wishes to escape a college-specific ecosystem like BoilerLink, there are only management solutions that target organizations such as startups and small businesses. There are no centralized systems developed for college clubs and student organizations. Some examples of these existing applications are:

Join It is a “Membership Management Software'' that offers integration with other apps, which is a primary focus for us as well. Despite this similarity, our target audience is different enough to warrant divergent implementations. This is self-evident by their pricing and their target audience. The majority of Join It’s features are priced around $1000 a year, which is out of reach for many of the organizations we hope to serve. They also advertise their system as appropriate for a wide variety of groups, such as non profits, gyms, and HOAs. An expanded audience might appear to be a strength, but for our chosen problem acts as a limitation; it fails to fulfill specific needs (such as reimbursements) that clubs would require.

Hivebrite and **Glue Up** have many of the same issues. They don’t target the college organization space and thus can’t deliver on anything beyond a general member database. Both of these tools additionally provide an interface for their general members to interact with. This is adjacent to the problem we are solving and not relevant. There is no need for there to be a front-facing platform for non-administrators to access.

Requirements

Functional

Member Management

1. “As the **President** of a club, I would like to easily be able to create different roles for the club.”
2. “As the **President** of a club, I would like to easily be able to assign the different roles of a club to other members.”
3. “As the **President** of a club, I would like to easily be able to modify the permissions of different roles of a club.”
4. “As the **President** of a club, I would like to easily be able to delete existing roles.”
5. “As the **President** of a club, I would like to be able to see the activity log of other officers.”
6. “As an **Officer** of a club, I would like to be able to add members (their name, email, and expiration date) to a central repository.”
7. “As an **Officer** of a club, I would like to be able to remove existing members from a central repository easily.”
8. “As an **Officer** of a club, I would like members to “expire” and remove themselves after the date specified when they are added.”

Documentation Hub

9. “As an **Officer** of a club, I would like to easily be able to view club documentation through an organized collection of Google document links.”
10. “As an **Officer** of a club, I would like to be able to search through the existing documentation.”

11. “As an **Officer** of a club, I would like to be able to create links for the documentation hub.”
12. “As an **Officer** of a club, I would like to be able to delete links from the documentation hub.”
13. “As an **Officer** of a club, I would like to be able to add video links (such as showing the process of submitting an APF) to the central documentation hub and have them be treated similarly to the Google docs.
14. “As an **Officer** of a club, I would like user access to documents to be based on role permissions, with automatic Google Doc permission syncing.”

Financial Hub

15. “As a **Treasurer** of a club, I would like to be able to add expenses to a central list of club expenses. Each item would have the amount and the source.”
16. “As a **Treasurer** of a club, I would like to be able to add income to a central list of club income. Each item would have the amount and the source.”
17. “As a **Treasurer** of a club, I would like there to be a central location for officers to submit links to images of receipts for any purchases on behalf of the club.”
18. “As a **Treasurer** of a club, I would like to be able to view all submitted receipts in a central list.“
19. “As a **Treasurer** of a club, I would like to be able to mark receipts as reimbursed when completed.”
20. “As a **Treasurer** of a club, I would like to share income and expenses with other officers of the club through a dashboard.”
21. “As a **general member** of a club, I would like to be able to donate to the club treasury using my credit card, or, alternatively, through a link to the bursar.” (stretch goal)

Event Hub

22. “As a **general member** of a club, I would like to be able to mark myself as present at events using an attendance code through a QR.”
23. “As the **Secretary** of a club, I would like to be able to generate an attendance QR code for members to scan at club meetings.”
24. “As the **Secretary** of a club, I would like to be able to see meeting member attendance statistics.”
25. “As the **Secretary** of a club, I would like to be able to see event attendance statistics.”
26. “As the **Officer** of a club, I would like to be able to create upcoming events with a specific name, date, and description”.
27. “As the **Secretary** of a club, I would like to be able to view the state (i.e. approval status) of my submitted APFs”.
28. “As the **Secretary** of a club, I would like to be able to view the comments listed under submitted APFs.”
29. “As the **Secretary** of a club, I would like to be able to respond to the comments listed under submitted APFs.”

30. “As the **Secretary** of a club, I would like to be able to delete submitted APFs.”
31. “As the **Officer** of a club, I would like to be able to view all upcoming events in a master list.
32. “As the **Secretary** of a club, I would like to be able to link to APFs for each event and see a simple icon indicating the APF’s status.“
33. “As a **general member** of a club, I would like to be notified ahead of time for events via email.” (stretch goal)
34. “As a **general member** of a club, I would like to be able to authenticate using BoilerKey when marking myself as present for an event.” (stretch goal)

BoilerLink Integration (Stretch Goal)

35. “As the **Secretary** of a club, I would like Clava to simplify the APF submitting process by cloning the process and simplifying the UI.”
36. “As the **Secretary** of a club, I would like Clava to automatically send the data from Clava’s APF form to boilerlink.”
37. “As the **Secretary** of a club, I would like to add other executive members as reviewers on an APF by selecting their names instead of individually typing out their emails.”

Election Hub (Stretch Goal)

38. “As a **general member** of a club, I would like to be able to vote in official club elections.”
39. “As the **President** of a club, I would like to create links for members to vote in for elections.”
40. “As the **President** of a club, I would like to view the results of an election.”
41. “As a **general member** of a club, I would like to be able to apply for an officer position.”
42. “As the **President** of a club, I would like to notify our members of the election results via email through the press of a button.”

Discord Integration (Stretch Goal)

43. “As an **Officer** of a club, I would like to be able to add a Discord bot to the club server that can integrate with our backend information, including but not limited to: finances, members, and permissions.”
44. “As an **Officer** of a club, I would like to be able to send commands to a Discord bot and have corresponding updates occur on Clava’s database, such as role updates.
45. “As an **Officer** of a club, I would like to set up automatic reminders and event notifications for our club’s Discord server via the discord bot.”

Non-Functional

Code Aspects

- Meaningful names should be used for variables, files, and functions.

- Easily understandable and extensible React Components to provide optional customizability for technical users
  - Components should remain “Reactful” and should follow the design principles laid out by the React team. (https://reactjs.org/docs/design-principles.html)
- Each module should be documented with information including but not limited to:
  - Its function
  - Its parameters (type and purpose)
  - What it returns (if not void)
- Consistent Indentation
- The system should be able to scale with the number of clubs
- User and club data should be stored securely
- Code styling should remain consistent through the use of a linter.
- Classes should follow a similar pattern to one another and make sense in the context of the project as a whole.

Architecture and Performance

We plan to develop the application using a separate frontend and backend. This will allow us to more easily separate the logic for interfacing with club-specific information and user interface. We will have separate individuals dedicated to each task. The client and server will communicate through a REST API presented by the backend. We will use the MERN tech stack. The frontend will be developed using ReactJS, and the backend with MongoDB and ExpressJS.

Security

Bad actors exist in all organizations and it is important that they aren’t given privileges in the club software to create havoc. As a result, it is essential that our product is given the proper security to ensure that isn’t possible. Starting off, we are aware that many parts of our stack have comprehensive security features. We intend to apply those when necessary as they are our best resource for ensuring security. As a side point, the previous assertion goes hand and hand on why we find it is essential to adopt the design philosophy of the different parts of our stack as much of the security is ensured by operating within those guidelines. Outside of resources provided by our stack, we intend to be as concise as possible when moving data from the frontend to the backend to ensure that only what is absolutely necessary is given to the proper user. It is also important that we implement anti-harassment measures to ensure that external users cannot obtain personal data about other users and organizations for malicious reasons. Furthermore, within our role system we intend to receive the necessary confirmation before a promotion or demotion is granted.

Usability

The user interface should be very simple and easy to use. This is especially important for the backside of the app which club officers will be using to manage members, club data, documents and other aspects of the student organization. All club related actions that can be taken by officers such as setting member roles, viewing club documentation, managing membership, as just a few examples, should be able to be performed through Clava. The member-facing front end should be simplistic, showing only actions that can be taken by members. Similarly, actions that can only be taken by people with the correct position in the organization, should only be visible to them. The application should also be well accessible on all screen resolutions for user convenience.