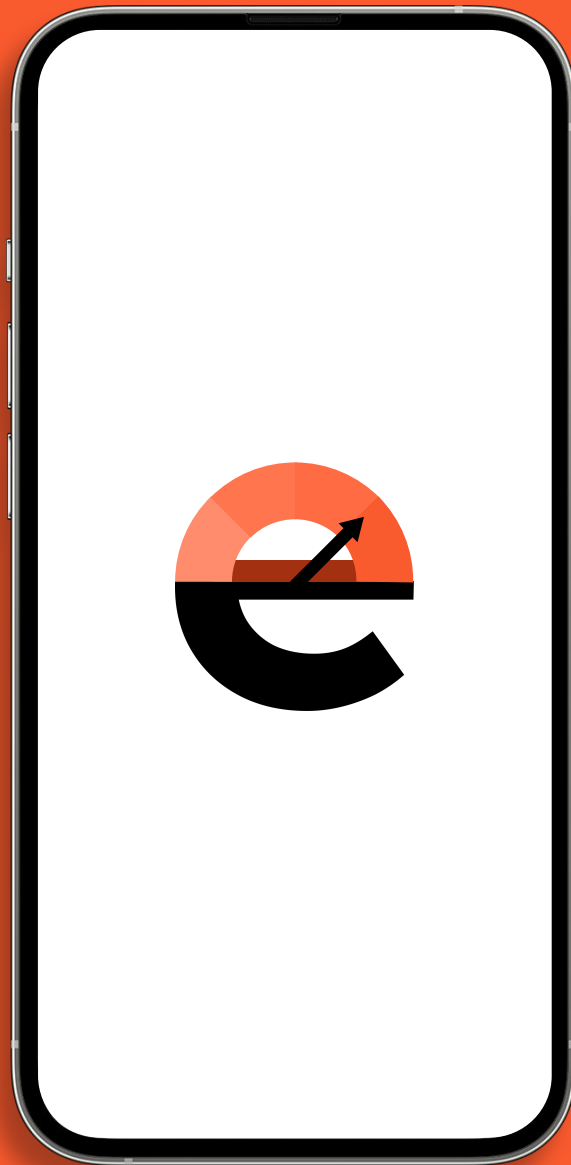# EnGauge

An Event Planning and Notification platform
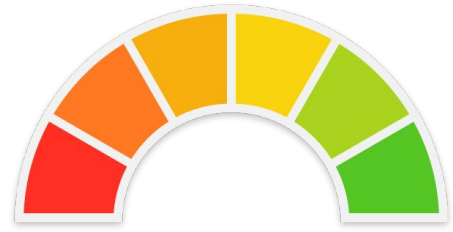
Group Members

Matthew Guck
guck

Tom Shahar
tshahar

EnGauge

Final Report

Introduction

We designed EnGauge to be used by Voluntary Student Organizations (VSOs), dorms, and social groups on Stanford's campus. As a club executive or dorm RA, it can be difficult to keep group members up to date about upcoming events— and even more difficult to gauge member interest in future events. On a high level, EnGauge is a platform that allows groups to:

1) Provide event details and timely reminders to group members,
2) Gauge member interest via a "Petition" feature for potential events.

There is an undeniable need for a better event notification system, especially on campus. Oftentimes, students aren't aware of the events happening on campus despite efforts by organizers to publicize them— Listservs are largely ignored and Fizz posts are easily drowned out by the inane drivel of karma farmers and paid moderators. Additionally, there is a clear need for a means of gauging interest in events. For example, as Class President, I (Matthew) have a very difficult time determining what Class events to hold or how to spend Class funds.

Design

EnGauge is an event notification and planning platform designed to enhance community engagement and streamline event planning for university groups. EnGauge's user interface is simple and intuitive, making it easy for students to navigate and access essential features. From discovering upcoming events to participating in group activities, EnGauge transforms the way students experience campus life.

Pictured to the right is the EnGauge Dashboard.
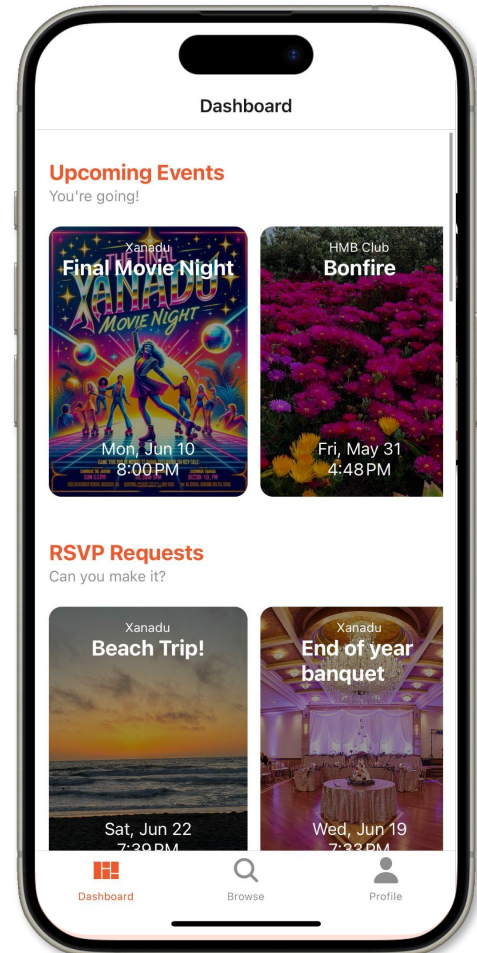
Design

The screenshot to the left is how the dashboard appears when a user has yet to RSVP to any events. RSVPing to an event is as simple as pressing ✓ from the dashboard.

The dashboard provides users with direct access to:
1) Upcoming Events
2) RSVP Requests
3) Event Petitions
4) Your Groups

Upcoming Events are placed at the top of the dashboard to allow users to quickly access information regarding each event. An 'Event' page contains:
1) The event name
2) A brief event summary
3) The event organizer
4) The time and date of event
5) RSVP button (Not Going, Maybe, Going)

The RSVP feature allows event organizers to easily get a headcount before the event, enabling organizers to plan at an appropriate scale. Group members are also able to *gauge* how popular the event will be based on how many people are attending.

Event Petitions (see right) set EnGauge apart from traditional means of communicating with large groups of people like GroupMe, Slack, or Flare. Event Petitions can be created by group owners to gauge interest in potential events. Group members can choose to sign petitions if they want the event to happen. An Event Petition keeps track of how many people have signed it, and displays that number out of the total number of group members:

"75/96 Group Members are interested"

Groups are a fairly self-explanatory, but core feature to the platform. Users can create groups for their dorms, clubs, friend groups, or even for their entire Stanford Class! A Group page contains a brief bio about the group, as well as Upcoming Events and Pending Petitions.
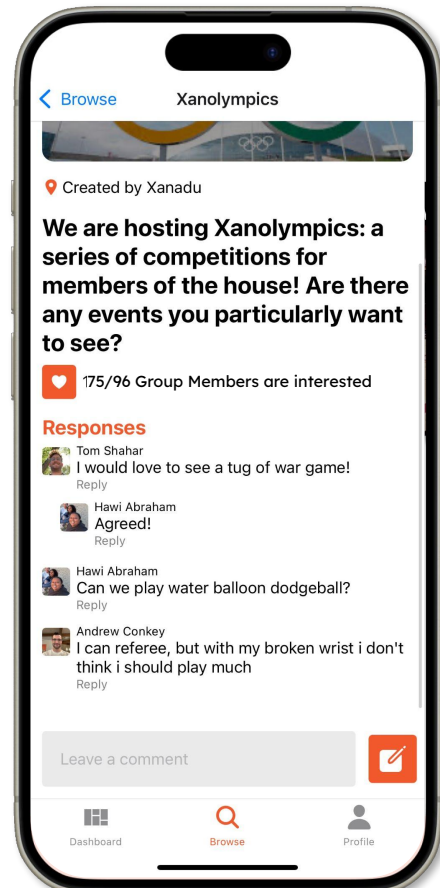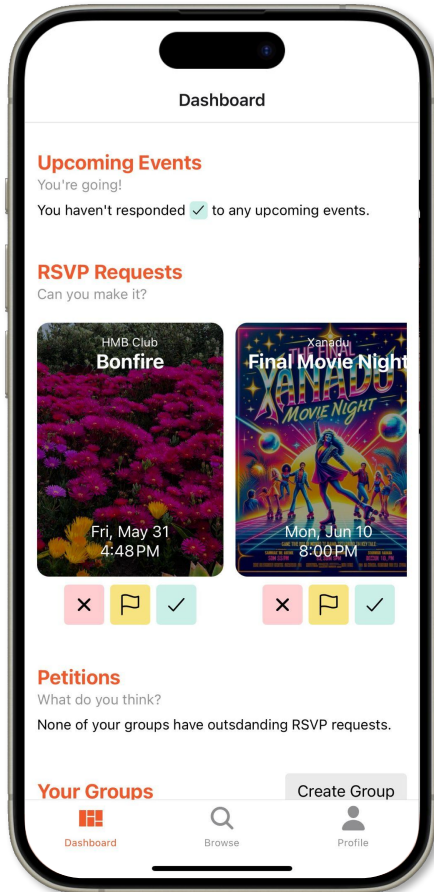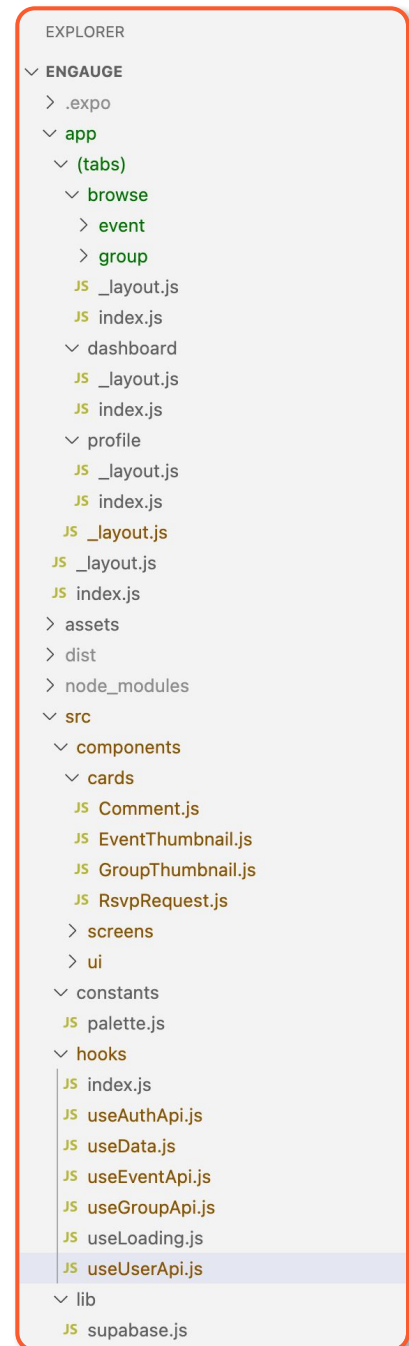
Implementation

EnGauge was designed and prototyped in Figma. We then developed our application using a React Native frontend, a Supabase backend, and Expo for deployment.

Overview of React Native Frontend

We used the expo-router library to set up the navigation and layouts of the apps. Once logged in (discussed below) there are three main screens: Dashboard, Browse, and Profile, placed within a Tab Navigator from expo-router. Browse has its own Stack Navigator, which contains pages for individual events, individual groups, event creation, and group creation. The main browse screen has a tab for groups and for events, whichever you wish to browse. It also has a rudimentary search bar, that narrows events or groups down by name only. The dashboard has lists of all your upcoming events, petitions and rsvp requests from your groups, and a list of your groups. The profile tab has fields to enter your name, class year, and upload a profile picture. We wrote dozens of custom components, and six custom hooks. Each hook included around 10 separate functions to handle various api calls (useEventApi handles event creation, updates, deletion, likes, comments, etc; and useUserApi handles profile changes, authentication, collecting specific objects based on a user_id, etc.). Due to the size of the app, we wrapped the entire router layout in a custom useData hook that wraps useContext to make it so that individual components can import useData instead of needing to pass countless props throughout the entire structure. useData keeps track of all high level app state, and is in charge of initializing the various arrays of objects we need to access on the component level. Components can import specific state variables from use data (such as groups or events), but normally import specific database functions, such as eventsOfGroup, which takes a group_id and returns an array of all events of that group. The groups state variable is initialzed on app load, so that we do not need to make separate api calls every time we need to load a different screen. One place we ran out of time

EnGauge's React Native Directory

Implementation

Overview of Supabase Backend

The supabase database has eight separate tables (profiles, rsvp requests, join requests, events, groups, comments, likes, and invitations). It also has three views (accepted_user_requests, event comments, and event_rsvp_counts), which concatenate information from the relevant tables. The database includes dozens of rls policies depending on which users we want to be able to perform actions. For instance, only group leaders can create events for their group. The database was created using a combination of the supabase gui (for simpler tasks, like adding columns) and the supabase sql editor (for creating rls policies, triggers, functions, etc). We used supabase authentication to verify and authenticate users. Users sign up with their email and create a password, click on an email verification link, and then can log in. We also used multiple supabase storage buckets so that users can upload profile pictures, event thumbnails, and group thumbnails.

id int4	name text	description text	event_date timestamptz	gro... i...	created_at timestamptz	privacy bool	rsvps_requested
26	Bonfire	EMPTY	2024-05-27 23:06:26.106+00	14	2024-05-27 23:06:39.465596+0	FALSE	TRUE
27	Bonfire	EMPTY	2024-05-31 23:48:00+00	14	2024-05-27 23:49:12.108629+0	FALSE	TRUE
29	Game vs TRC	EMPTY	2024-06-01 01:02:00+00	17	2024-05-28 01:03:35.422508+0	FALSE	TRUE
30	Game vs JOR	We need to win this one! It's a playoff gan	2024-06-01 03:46:00+00	17	2024-05-29 03:47:26.920813+0	FALSE	TRUE
31	Next season!	Who wants to play? What should our tear	2024-06-30 00:03:00+00	17	2024-06-05 00:05:15.808918+0	FALSE	FALSE
32	Final Movie Night	Join us for our final movie night! We will l	2024-06-11 03:00:00+00	15	2024-06-07 00:23:40.288892+0	FALSE	TRUE
35	Xanolympics	We are hosting Xanolympics: a series of c	2024-06-08 21:00:00+00	15	2024-06-07 01:05:24.163501+0	FALSE	FALSE
36	Spikeball Tournament	Join us for our second annual spikeball to	2024-06-27 02:25:00+00	16	2024-06-08 02:25:46.631033+0	FALSE	TRUE
37	End of year banquet	Our annual end of year banquet, featuring	2024-06-20 02:33:00+00	15	2024-06-08 02:33:56.804863+	FALSE	TRUE
38	Beach Trip!	Lets go to the beach, beach!	2024-06-23 02:39:00+00	15	2024-06-08 02:40:37.422035+0	FALSE	TRUE
39	Move out :(	we all gotta move out for the summer!	2024-06-27 02:42:00+00	15	2024-06-08 02:44:45.970942+	FALSE	FALSE
41	Fountain hopping?	Can everyone suggest a date and time th	2024-06-11 02:50:00+00	15	2024-06-08 02:51:57.157774+0	FALSE	FALSE

Snippet of EnGauge's Supabase Backend

Viral Usage

EnGauge was initially launched to our closer groups (primarily our strong ties). Since most users were not Group creators, we fortunately did not face any issues with inappropriate events and petitions being created. User behavior was as predicted/designed for, though we recognize that on a larger scale, with people who do not know us, this *will* not be the case.

We were initially weary that people may not respond to petitions, but we were pleasantly surprised at the utilization of the feature. This was likely due to us placing petitions very clearly on the dashboard. We had similar concerns about people actually using the RSVP feature; however, because it is necessary to populate the "Upcoming Events" field on the homepage, users also readily engaged with the feature.

Our platform had ~7 *active* users, but 15 total.

Due to our simplified system (and small scale launch), we did not see much unintended behavior.

Design Reflections

What Worked?
The petition system worked great! Tom used the petition feature to effortlessly organize events with friends. This feature allows users to bypass a lot of the difficulty that typically comes with making plans. Tom created a couple different petitions for event ideas and his friends were able to sign petitions to indicate what they wanted to do.

Matthew's EnGauge Planned Event!

The RSVP feature also worked quite effectively. Matthew hosted an end-of-quarter get-together, and was able to buy an appropriate amount of snacks for his 5 friends who came— a number given to him via the RSVP counter on EnGauge.

What Didn't Work?
Our current Group privacy options are less than ideal. During the prototyping phase, it was our intention to allow some groups to be invite/request only, and for other groups to be fully public (meaning anyone on the EnGauge platform could join without direct owner approval). Currently, groups are request only, which limited the navigability of the system.

Event privacy options faced a similar issue as group privacy. We wanted groups to be able to host open events like "Get to Know SSI" or "Coffee Chat with FashionX"; however, events are currently only visible to members in a certain group. This again limited the navigability of the system and made it feel less social than we had intended.

Our petition system is still pretty primitive. Currently, the only options on a petition page are to sign the petition or leave a comment. Ideally, users would be able to add amendments to petitions that could be voted on by other users. This is partially present with the current comment system; however, that use case isn't explicitly pushed by our design system, leaving it up to group members to use the comments as a separate ideation space.

Reflections
Our system currently has no way for group owners to moderate comments. Joining a group is by request only, meaning group owners, especially at our current scale, are really only accepting members they know and interact with already. Because of this we did not experience any anti-social behavior, but readily recognize that negative comments or profane petitions or events will inevitably occur if the platform were to scale. Moving forward, we would like to implement moderation such that group owners can moderate comments on their posts, and separate moderators can moderate public events.

We think that a platform more focused on events themselves (rather than group *communication*) is really promising. The petition and RSVP features are powerful and something we both would utilize as regular event planners.

Theory

Motivation

EnGauge was design to fulfill the need of effective communication of event details to members of social groups, large or small. While EnGauge is designed for use by both group executives and group members, we wished to solve problems that social group executives experienced while planning events. Our target audience can be characterized concisely as "people who plan events." Event Petitions and a designated RSVP feature set EnGauge apart from traditional means of group communication like GroupMe, Slack, or Flare. While the aforementioned platforms are full of incredible, versatile features, this makes them far more susceptible to Grudin's Paradox, in the sense that the socio-technical system may be benefiting group members, while complicating the process for group leaders. We believed that our design would add more benefits to the event planning and detail communication processes. By limiting direct communication on our platform (i.e. no DMs), EnGauge is more effective at filling a specific niche.

Analysis

The Cold Start Problem is real. Our first users were a bit unsure about the norms and exact use cases on the platform, since it was essentially barren. Before asking more people to join the platform, we decided to populate it with our own groups and events, which made it much more clear how the system was designed to work to new users. After populating the system, users reported little issues with understanding its purpose and navigating the system.

For our platform, we identified that Easy Side was group owners, and the Hard Side was group members. Our initial users who created groups reported difficulties convincing their group members to join our platform; however, we think that with a more populated and polished platform, a large group could easily incorporate EnGauge into their communication flow.

Since our goal was to strip away the bells and whistles characteristic of more all-in-one platforms, "posts" on our platform are flat, and generally just come from "spaces" rather than networks or commons. If we were to continue developing our platform, we would "commons" posts would be more prevalent (see "What Didn't Work?").

How to Access

1) Download Expo Go on your device
2) Follow this link on your devices or scan the QR it displays
3) Login with:
   Email: tshahar@stanford.edu
   Password: 12345678