Ideate

Team Rinnegatos - Ethan, Jessica, Melissa, Niki

Overview

I.   Motivation

Problem

In a meeting, whether in the workplace or in college, many voices often do not get heard. This can be because of several reasons:
- Someone is more quiet compared to the loud voices in a meeting
- Someone has a less senior position than other people in the meeting
- Some people might be unconsciously biased towards certain types of people

This means that some good ideas will never be heard in a meeting. Ideate solves these problems, by allowing all ideas to be heard.

System Description

Ideate is a web application that allows people to submit ideas anonymously during a meeting. These ideas will be shown as 'bubbles' on a board. During the meeting, people can look at the board, and if they like a certain idea, they can upvote the idea by clicking on that bubble. Whenever a person upvotes an idea, that bubble grows larger. This allows everyone in the meeting to submit ideas, assess ideas and see the popular ideas. People can then submit anonymous notes on an idea through the web application. People can also flag an idea if they want to discuss that particular idea in the meeting.

Key Purposes

Allow all voices to be heard

Ideas are submitted anonymously, so all ideas can be heard and discussed. This allows quiet, or less senior people to 'speak up' in a meeting, and also removes any unconscious biases towards certain types of people.

Get real-time feedback for certain ideas

The application can be used during a meeting and allows people to submit feedback in real time. This is useful because if an idea bubble gets very big, then the idea can be discussed in that meeting.

Organise ideas

Ideate captures all ideas and upvotes in a meeting which creates a more organized display of ideas during and after a meeting.

Design Essence

I.   Concepts

Idea

Purpose: allows a user to propose a solution or topic to the group.
Operational Principle: if a user wishes to propose a topic to discuss or vote upon, then he can anonymously add it to a board, which then allows all other users participating in the board to view the idea.
Misfit: two users could submit the same idea.
Description: a bubble/resizable shape that shows the text of the idea.

Board

Purpose: allows users to consolidate ideas in one overarching group.
Operational Principle: a board moderator can start a board that all attendees will sign into. The board will then show ideas and their popularity as attendees start posting. Boards can then be saved by participants with accounts to be viewed later.
Description: a page that shows all of the ideas relating to that board.

Upvote

Purpose: allows users to get a sense of which ideas are most agreed on by the group.
Operational Principle: a user can upvote an idea to show their approval for the idea. Other users will be able to see how popular an idea is by how many upvotes it has. Users can also un-upvote an idea if they wish to retract their statement of approval.
Description: an icon on an idea that allows a user to upvote that idea.

Flag

Purpose: allows a user to alert the moderator that they wish to discuss a specific idea.
Operational Principle: a user can flag any idea on the board for discussion later, which other users can also see. When the group begins to discuss ideas, they can add this to the list of ideas for discussion.
Description: a flag on an idea that can be clicked by any user at any time. Indicates that at least one user wants to discuss a specific idea.

Moderator

Purpose: allows a person to create a board and own the ideas in it.
Operational Principle: if a user wants to start a new board and allow users to contribute ideas, then they can become a moderator and create a board to organize those ideas. The moderator also has additional administrative privileges such as deleting ideas, unflagging any idea, and saving the board.

Description: a type of user who owns a board. Anyone who starts a board becomes the moderator for that board.

Explanation

Purpose: allows a user to add a description to an idea that all other users can see.
Operational Principle: a user can create an idea with some keywords, and then add an explanation for more details about the idea. Other users can then view an idea's explanation if they want to get more details about that idea.
Description: a text editing page that allows a user to add an explanation to an idea. Other users can click on an idea to see the explanation.

Notes

Purpose: allows a user to write down personal notes associated with an idea.
Operational Principle: a user can add notes to an idea; for example, points they will want to discuss regarding that idea. Only the owner of an idea can see the notes.
Description: a text editing page that allows a user to take notes on an idea.

Observation: The concept of Explanations and Notes may be outside the scope of our project. The implementation of this concept will be a reach goal for our project.

II. Data Model
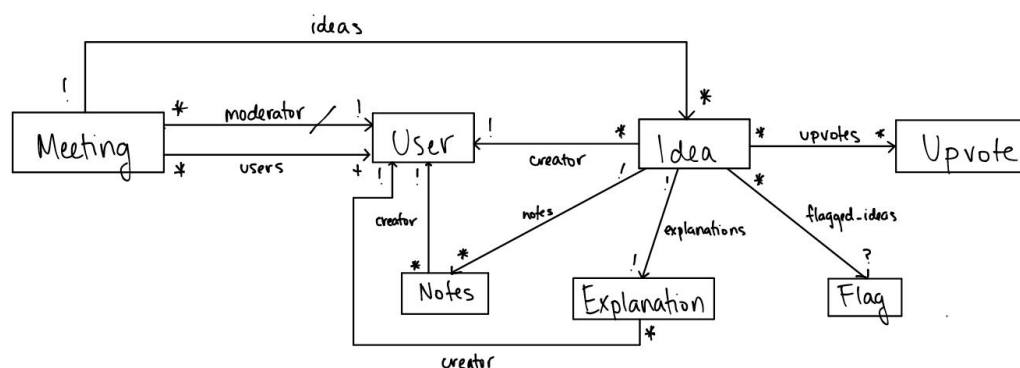
III.   Security Concerns

A. Key Security Requirements

Meeting participation

Summary

A board for a meeting should only be viewable and editable by meeting attendees. Users who are not attending or invited to a meeting should not be able to either view ideas on the board nor upvote or flag any ideas. This is necessary to prevent private and potentially confidential information from leaking to attackers, and to prevent them from unfairly swaying the popularity of an idea via upvoting.

Plan to Address

We plan to address this security issue by only allowing users to join a board if they have the unique, random code for that board that is generated by our application. The secret code is displayed when the moderator first makes the meeting, and then they can give that code to whoever they wish to invite to the meeting.

Sharing Saved Boards

Summary

One of the components of our application is the ability to save boards of ideas so that they can be examined again at a later date. These saved boards should be private, and so it is necessary to ensure that attackers cannot gain access to them and steal ideas.

Plan to Address

To address this security issue, we plan to make the moderator of a meeting be the one in charge of deciding who has access to that board after a meeting. The moderator will have the option of saving a board at the end of a meeting. If they decide to save the meeting, it will join the collection of their other saved meetings, and will be only visible to them. To share the board, the moderator can get a shareable link, and then send that link out. Users with the link to the board will have read-only permissions.

Preserving Anonymity

Summary

In order to promote ideation, our application allows meeting attendees to anonymously post ideas to the meeting board. Anonymity of ideas will help ensure that everyone at the meeting has a voice, and it can even help reduce the impact of societal biases on ideation. Thus, it is very important that the anonymity of each idea is preserved, and that no other users, even the moderator, can determine the origin of an idea.

We will preserve anonymity of users in our application in a couple of ways. For users who join a meeting without making an account, we can easily preserve anonymity by not asking for or collecting any personal information about that user. For users who do have an account and are logged in, we will preserve anonymity by never passing user information to the board so that the board, and thus other users, will not be able to identify the origin of an idea.

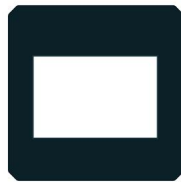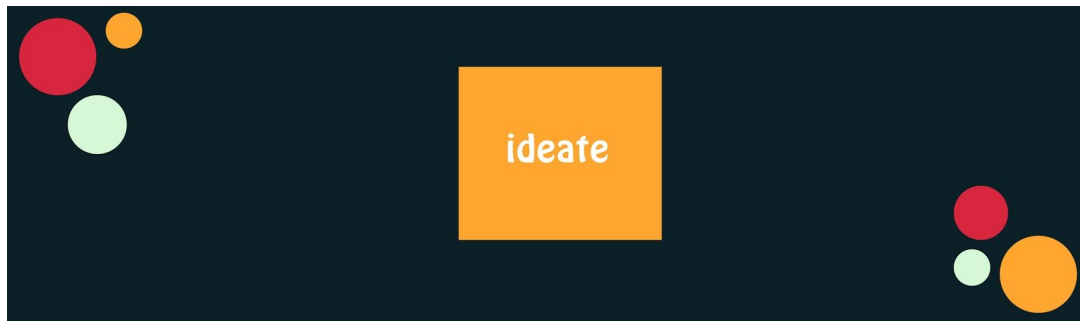## B. Mitigation of Standard Web Attacks

We will mitigate standard web attacks by general application hygiene. In order to combat XSS attacks, we will use a templating framework such as handlebars.js to make sure that all untrusted content entered into our website is properly escaped. This will prevent attackers from being able to inject scripting code into the input fields in our application. We mitigate other injection attacks in a very similar manner. All input values will be validated and escaped before being passed to the application. Additionally, we will use parameterized queries to differentiate between code and data in our database queries to prevent database injection attacks.

To combat CSRF attacks, we plan to do a mix of two strategies. First, we will include a CAPTCHA in the signup view. We put the CAPTCHA on this page to prevent bot users from creating accounts. Second, we will verify same origin for requests by checking the standard HTML headers. We will check that the source origin and the target origin match, indicating that there has been no request forgery.
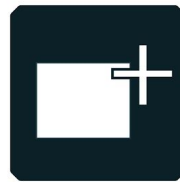
C. Threat Model

The main threats to our application come from attackers looking to steal confidential or private ideas from meetings. However, as we are not dealing with large amounts of money or other very valuable content, we do not foresee our attackers being very invested in attempting XSS or CSRF attacks. Additionally, we will limit user access to the server and databases to further prevent the opportunity for attacks.
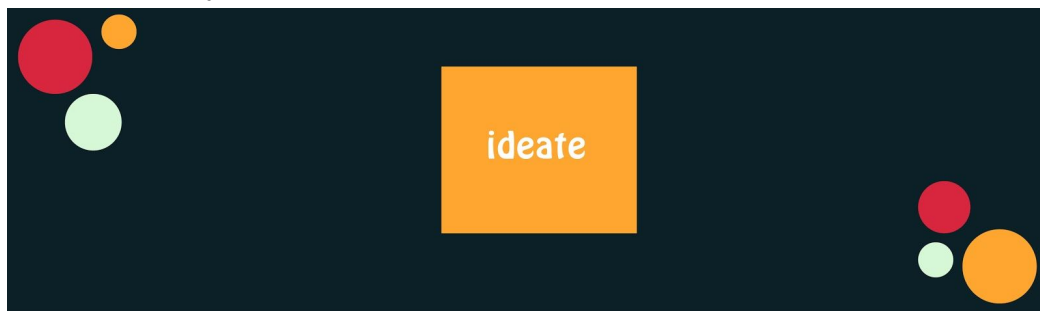
IV.   User Interface

Frame 1. The homepage of our app where users can either create a new board or join a board that has already been created.

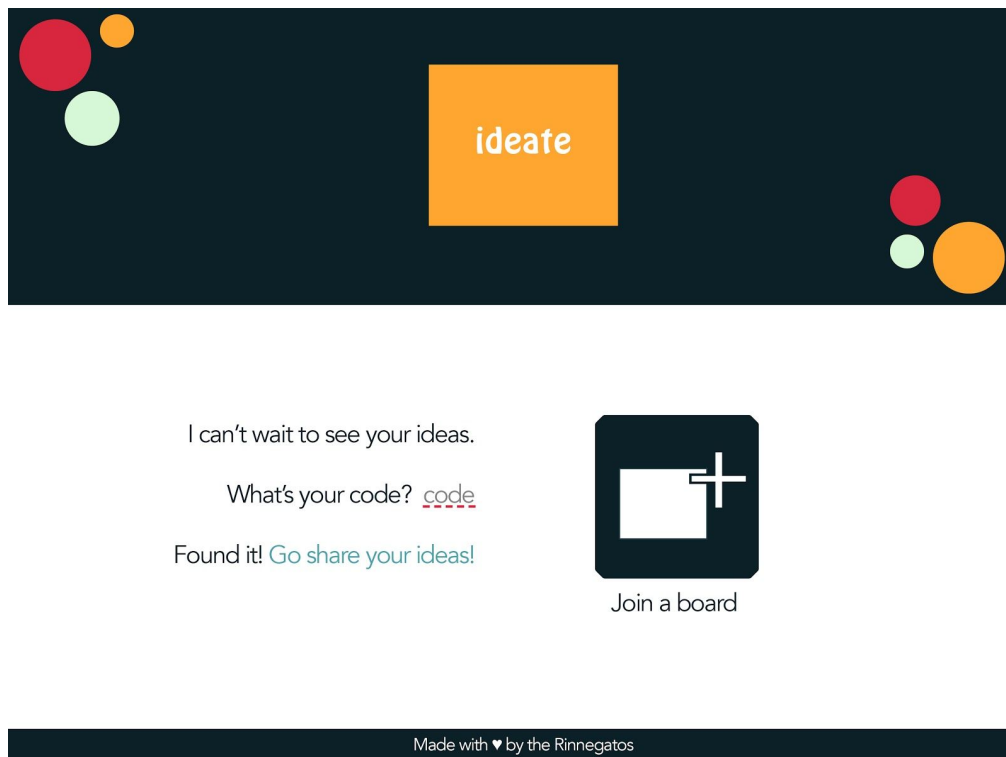Frame 2. After users click on 'Create a new board', the other option fades out and this text fades in. The code (underlined in red) is automatically generated for every board. The blue is a hyperlink that leads to the board corresponding to the created code.

ideate

I can't wait to see your ideas.

What's your code? code

Found it! Go share your ideas!

Join a board

Made with ♥ by the Rinnegatos

Frame 3. The screen shown after users select to join a board. 'code' is a text entry field where users can enter the code that corresponds to the board they want to join. After a valid code is entered, the bottom line of text appears so that users can go to the board.
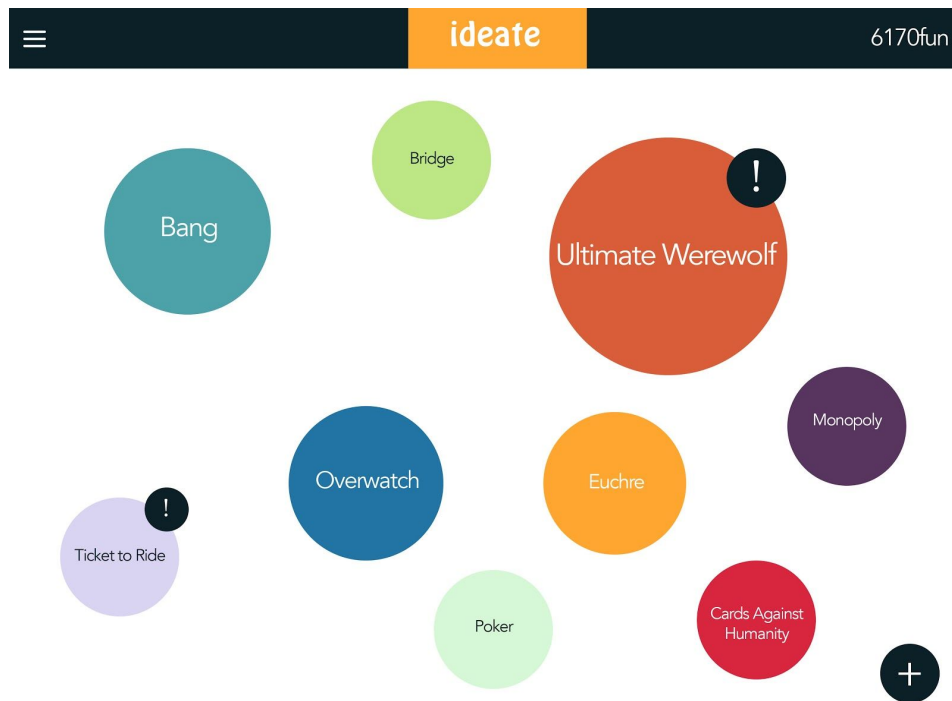
Frame 4. The main view of our app, where ideas are shown in one board. The code (in this case, 6170fun) is shown in the top right so people viewing the board can join. Ideas are presented in bubbles where the size of the bubble corresponds to the number of upvotes. Users can click on a bubble to upvote it and can double click a bubble to add a flag to it, marking it for discussion.
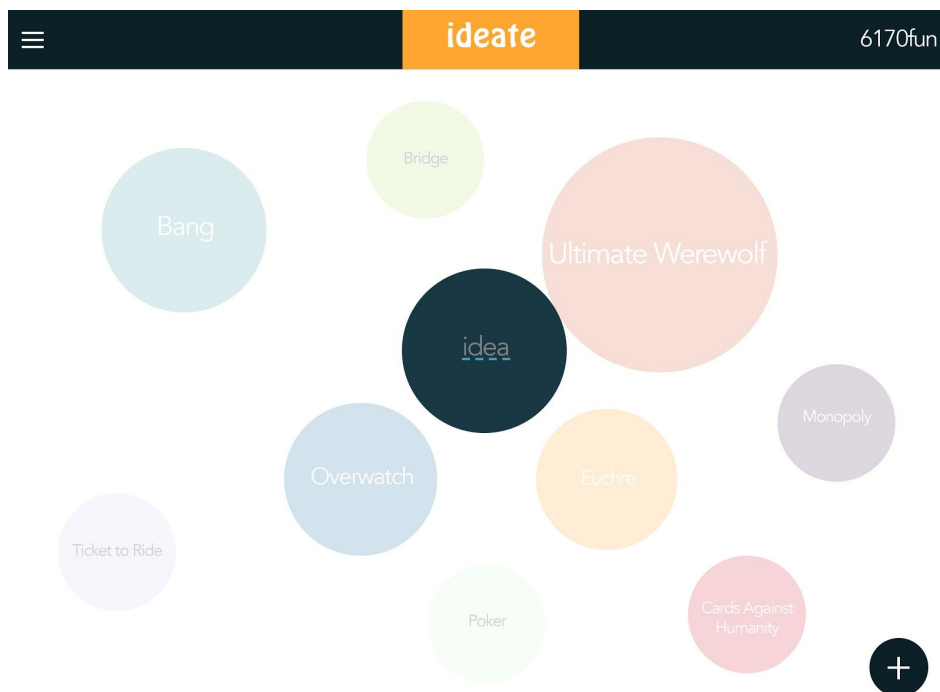
Frame 5. The view a user sees when they click on the add button in the bottom right. The user can enter their idea in the text entry field on the bubble and then push enter, which will revert to the overall board view with the new bubble added.

V. Design Features

Necessary

- Basic board functions
  - Add idea
  - Upvote idea (only once)
  - Flag idea
- Creating and entering boards
- Moderator privileges (gained by creator of board)
  - Deleting ideas
  - Unflagging ideas
  - Saving a board (if moderator has an account)
- Ability to create and use a board without having an account
- Ability to create and use a board with an account

Possible/Reach

- Animations for board (bubbles moving, rotary menu)
- Animations for home screen (buttons and menus sliding in and out)
- Saving boards to pdf (accessible to all users)
- Advanced board functions
  - Create explanation for an idea (visible to all users)
  - Create notes for an idea (only visible to idea owner)

Challenges

I.   Design Risks

We encountered several risks associated with some of our key concepts during our design process that could possibly result in misfits.

Collision of ideas

Collision of ideas occurs when two or more people present the same idea. Preventing this sort of action from occurring would be difficult. You could imagine a scenario where we implemented a fancy duplicate detection algorithm that tried to prevent users from posting duplicate ideas with different wordings, but this likely would not be very successful and would almost certainly result in false positives. Instead, we will give the moderator of the board the ability to delete ideas. This lets them decide what to do with duplicates; since users can upvote as many ideas as they want, there is not necessarily an issue with having duplicates, but giving the creator of the board the option to delete them will allow them to control this scenario.

Misuse of anonymity

Any anonymous system is open to potential risks of misuse. While this probably would not be of huge risk in a company meeting setting, the ability to remain anonymous while submitting arbitrary text that will be displayed on screens of everyone participating comes with a huge risk in settings like high school classrooms. Because of this very likely use case, we will give the moderator the ability to require that users are signed in to the website instead of being anonymous. This will present a barrier to prevent random people from joining the meeting. Moderators will also be able to kick out users who propose certain ideas without actually seeing which user presented which idea. This preserves the anonymity of all users while allowing the unruly or unwanted user to be kicked out of the meeting.

II.   Design Choices

Example Use Case

We have one primary use case that we considered while designing our app. As an example of our primary consumer, we considered a recitation section where a TA is leading a review session for a test and wants the students to suggest topics to review. However, students may not want to admit what topics they are uncomfortable with, so the TA could use our app. She would create a board and share the short url with her students, who could then view the board on their laptops and anonymously add the topics that they would like to review. If a topic is already on the board, students can upvote the topic so that the TA knows that many students need to review this topic, but individual students do not have to admit they are uncomfortable with the topic. The TA may also flag ideas that she particularly wants to cover.

Anonymity

One of the issues that arose was how anonymous we wanted users to be to the moderator of a board, particularly if they had to sign in to a profile to join a meeting.

Options: Make it so the moderator can see which users propose which ideas, or keep it all anonymous

Evaluation: If we made it so that a moderator could see which users proposed which ideas, it would ensure that users wouldn't act in an unruly manner because they could face consequences for their actions. However, this would also be in conflict with the point of the app: to get ideas from everyone in a group without any possible judgment of the ideas.

Decision: We decided to go with full anonymity. Moderators will be able to kick the author of an idea, but they will not have any way of knowing who the author was. This ensures some level of security while preserving anonymity.

Scope of Application

We discussed what exactly the scope of our application would be, and whether it would simply be for sourcing ideas or if it would include feedback and discussions for ideas as well.

Options: Allow users to give "feedback" for ideas on the board, or have the board purely be a place for the sourcing of ideas and let discussion happen in the real world

Evaluation: If we let users give feedback on the board, it would further improve meetings in terms of fairness. However, in-person meetings certainly should not fully take place on phones interacting with a web app, and including feedback in the app would probably lead to a splitting of concepts and purposes; for example, the board would no longer just be for the sourcing of ideas, but would also allow for some discussion of ideas.

Decision: In order to simplify our user experience, not split purposes, and maintain focus, we decided to not include feedback for ideas, but instead allow individual users and the moderator to take notes if they have any thoughts they wish to bring up during the in-person discussion or to save for later.

Voting and Flags

Upvoting was definitely something we wished to include in our design, since we needed a way to determine which ideas people agreed with. But what about if people disagreed with an idea?

Options: Add a downvote option and show both upvotes and downvotes to see how divided the group is, make ideas flaggable to let the group know when people wish to discuss an idea further

Evaluation: Downvoting comes with some benefits. It immediately lets the group know both how many people like an idea and how many people dislike it. But chances are that ideas that would likely get a lot of downvotes would be ones without many upvotes. Not only would people feel bad if their idea got immensely downvoted, but it doesn't provide us with any new information. When downvotes will make a difference, it will be for contentious ideas. Making ideas flaggable for discussion shows that members of the group believe that an idea is worth discussing in more detail—whether they really like it or really hate it—and then lets them present their ideas in-depth out loud during the meeting.

Decision: We went with flagging instead of downvoting because of the reasons listed above.

UI Choices

When designing the UI for the app and working on the wireframe models, we had to balance usability with simplicity. We want people to be able to quickly learn and use our app, but also need it to be functional.

Options: Create many options and have a feature rich UI that may be more complicated to use, or have a streamlined simple app that is easily learned

Evaluation: As mentioned earlier, we could allow users to create notes or downvote ideas, but in addition to the problems above, this would also cause the UI to become more cluttered and harder to understand. Removing most functionality except the required actions required to create/join a board and add/edit ideas makes the UI much easier to learn and understand.

Decision: Since one of the main goals of our app is to make it really easy for users to quickly enter a code, join a board, and start adding ideas, we made a very simple UI with only the necessary options.