

Talkatiel Software Requirements and Planning

Brendan Byers, Ryan Sisco, Iliana J, Aidan Grimshaw, Yufei Zeng

ONIDS HERE

1 Product Description

When people move to a new area, it can be difficult to interact with new people. People move away from their hometown and have an increased reliance on social networks such as facebook and twitter to feel connected to their friends and families at home. This results in people not feeling part of their new communities. The old communities have been left behind but people have trouble fully integrating into their new towns and neighborhoods. College students are extremely susceptible to this feeling due to the fact that since 1986 the number of out-of-state freshmen has doubled[**strayer'2016**].

Talkatiel provides a safe way to engage in a community without actually meeting anyone. Users no longer need to fear going out to social events or community activities and dealing with terrifying or uncomfortable social situations. This application allows people to integrate themselves into a community without great risk to themselves. It can provide the deep seated need to be part of a community while also allowing people to be themselves without feel of social repercussion.

Talkatiel will be a Portable Web App, or a PWA. This allows it to behave as an app does while also being much more accessible and easier to use for our end users. The web app itself will consist primarily of 3 elements: the service worker, the app shell, and the backend database. A service worker is a chunk of code that exists outside the website. It stays on a users device and works to cache and manage data for the web app. It will also allow us to send push notifications to the user. The app shell is the skeleton behind the web app. It is stored on the users device, allowing the web app to load quickly. It manages the basic user interface which is then filled with content from the server. The database will store all posts along with identification and metadata information.

Previously this idea has been attempted by Yik-Yak. Originally released in 2013, Yik-Yak set out to solve many of the problems we are solving with Talkatiel. One of the biggest issues that Yik-Yak encountered was bullying and harassment on the platform[**bramson'2014**]. The only moderation present in the app was an automod that deleted posts that had certain words. This did not effectively deal with hurtful messages, only invited people to figure out better ways to circumvent the moderation. To solve this, Talkatiel will use both word filtering and actual people to moderate. As seen on sites like reddit, using trusted users to moderate other users can be a strong and effective tool.

To build Talkatiel we will use Polymer for the user facing side. It has been proven to be a simple and fast portable web app tool that allows us to quickly move from the idea stage to the testing stages of features. Polymer is built on a mix of HTML, javascript, Node.js (4.x) and Bower. The database will run on the google service FireBase. This allows us to easily store and distribute user posts and settings and integrates easily with Polymer. In addition to simply delivering data, Firebase also has extensive analytic and monitoring tools, allowing us to see the health of Talkatiel at a glance.

We will be designing this for use primarily on mobile device browsers. These include

Chrome, Safari, and Firefox. Because of the way PWAs are structured, they should also perform well on desktop browsers such as Chrome, Edge, and Safari.

Talkatiel will use a service worker to precache parts of a web app, allowing it to load quickly the next time the user uses it. This allows us to manage what the user sees while cutting down on network traffic and load times. A service worker can perform its tasks even when the web app isn't open. This means in addition to caching and managing web app data, it also has the ability to allow users to access things offline, intercept HTTP/HTTPS requests and even send push notifications to the user.

1.1 Functional Requirements

1.1.1 Posting

User will be able to create and post in discussions anonymously. They will be able to read and respond to posts by other users. Users will be able to reply to a discussion, reply to another reply and vote on posts.

1.1.2 Moderation

There will be both computer and real person moderation. The computer run moderation will filter out posts with blacklisted words or phrases, while the human moderation will be done at the best judgment of whoever is moderating.

1.1.3 Geolocation

Users will only be able to post to certain discussions if they are within a range of the discussions marker. If a user is within a certain distance they will be able to post to discussions. If they are out of range of a discussion, they won't be able to view or participate in the discussion.

1.1.4 Viewing Posts

Users will be able to scroll through posts made by other users. They will be able to see the text of the post, and possibly how many votes the post has. A user can then respond, vote, or continue scrolling to new content.

1.1.5 Reporting

Posts will be able to be reported for reasons such as spam, abuse, bullying, or other issues. These posts will then be dealt with by a human moderator, who can then choose to remove the post or let it be. This will help cut out harmful speech that isn't wanted on the platform.

1.2 Non-Functional Requirements

1.2.1 Secure

Talkatiel will need to utilize a secure connection (HTTPS). All content needs to be delivered through a secure connection to confirm that all aspects of the app are controlled by us. This

tops malicious 3rd parties from abusing the service.

1.2.2 Network Independent

The app should still work when there is little or no network connectivity. If not showing the different discussions, at least displaying a page notifying the user of the fact. This will allow it to act as though it is a native app, though it is actually a website crafted for their device. Also stops it from appearing "broken" to users, which could result in losing them.

1.2.3 Responsive

Talktatiel should be responsive to users on both phones and tablets. The pwa should react quickly to user inputs, meaning that both the application presented to the user and the back-end database need to do their tasks speedily.

1.2.4 Deep Linking

Each page should have a distinct URL. This means each discussion, post and reply should have it's own individual URL. This allows user to share the post with others, such as on facebook or other social media services.

1.2.5 Standard App Manifest

Talktatiel should include a file that specifies all the metadata associated with it. This allows us to set various settings for how the web app is displayed, such as orientation, display modes, image links, URL settings, and theming. This is an integral part of making a portable web app and signals to browsers how to handle the website.

1.3 Documentations

In addition to a simple and intuitive user interface, there will also be a limited help section present. It will show users how to do different basic functions of the application, such as navigating, posting, voting and the purpose. The help section can walk a new user through the application and introduce them to the features. By keeping the Talktatiel simple we can limit the amount of documentation needed while also improving user retention.

1.4 Major Features

- Post text as part of discussion
- Reply to other user's posts
- Ability to vote up or down other posts
- Both computer and human moderation of posts
- Ability to participate and view discussions depending on location
- Ability to report posts to Moderators or Administrators

1.5 Minor Feature / Stretch Goals

- Ability to save discussions for later viewing.
- Ability for a user to view all of their own posts
- Keep track of the total votes a user has recieved
- Integrate ads into the platform

2 Use Cases

- Use Case 1
- Use Case 2
- Use Case 3

3 Planning

3.1 Milestones

3.2 Timeline

3.3 Project Tracking

3.4 Risk Management

4 Meeting Report

4.1 Progress Made This Week

4.2 Plans for Next Week

4.3 Team Member Contributions

4.4 Customer Meeting Status