Null Set Poster

Benjamin Duke, Tino Pimentel, Alexander Swanson

CSCI 462

April 21st, 2020

Abstract

The goal for this project is to practice software development knowledge in a practical way with an open source software. In our case we are working with Zulip through Github to maintain their codebase. In particular we have been working on fixing several bugs throughout the entire semester. Ultimately, learning and strengthening our software engineering skills.

What is Zulip?

Zulip is a powerful, open source group chat application that combines the immediacy of real-time chat with the productivity benefits of threaded conversations. Zulip is used by open source projects, Fortune 500 companies, large standards bodies, and others who need a real-time chat system that allows users to easily process hundreds or thousands of messages a day. With over 500 contributors merging over 500 commits a month, Zulip is also the largest and fastest growing open source group chat project.

Lessons Learned

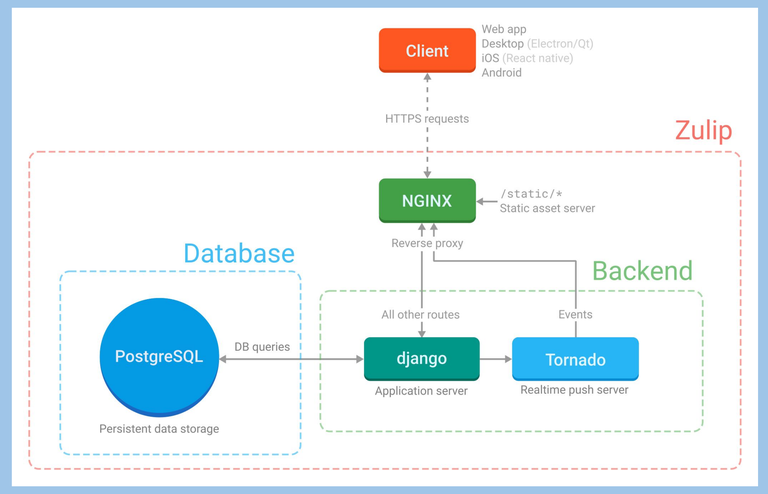
Over the course of 15 weeks our group has learned many lessons while working with software. The most important lesson we learned was patience, many things change in the environment over time. The project structure itself is very agile. Scheduling is the next important lesson we learned throughout the semester as it helped a lot for determining our timing for the project.

How Zulip Works

Zulip provides the benefits of real-time chat, while also being great at asynchronous communication. Zulip is inspired by email's highly effective threading model: Every channel message has a topic, just like every message in email has a subject line. (Channels are called streams in Zulip.)

Topics hold Zulip conversations together, just like subject lines hold email conversations together. They allow you to efficiently catch up on messages and reply in context, even to conversations that started hours or days ago.

Zulip has a front end where users can see the messages they write and send to the stream(s), and then Zulip compiles the messages in its backend using Python based Django for compatibility and functionality to return the Message to the receiver back on the front-end.



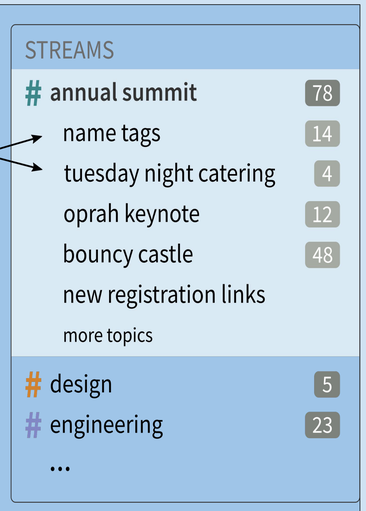
Contributing to Zulip

Included in the documentation on the Zulip GitHub is a contributing guide.

The guide explains the different ways a user or developer can contribute to the open source project.

Ways to contribute include not just developer tasks like bug fixes and feature implementation, but also improving documentation, providing user feedback, or finding a bug and creating a bug report to notify the community. This way users with many different levels of experience improve the software over time.

The community is very welcoming, and include tags for open issues that are good for first time contributors.



Accomplishments

Experienced how to contribute to an open source software project with Zulip.

Learned how the program works by looking into open issues on the project's GitHub, experimenting with the code, and attempting to fix bugs.

Understand more about the complexity of Github's version control system when projects push large-scale updates, leading to us developing better code.