DevOP by OpFlow

- James Setliff
- Christopher Lebovitz
- Courtney Hill
- Kacheef White
- Charles Young

Agenda

- Application Overview
- Platform & Frameworks
- Requirements & Design
- Implementation
- Test Plan
- Problems & Challenges

Application Overview

DevOP is based on the Electron framework and as such will run on any modern operating system. The application brings the functionality of many popular development tools under a single umbrella and supports personal and team-based record keeping, visual task board to easily convey that information, and real-time chat with persistence. DevOP is targeted at teams that want modern planning and team-work functionality without the monthly subscription and cloud-based storage of information that most of these tools require.

Application Overview

The DevOP client will be deployed to each individual team member's computer but may also be hosted on a webserver for remote connections, thanks to the flexibility provided by Electron. The client can be pointed at any MySQL database for storage of information; however, it is designed with the idea of a team-owned database managed locally. Team members will be able to create and update records, manipulate task boards, and communicate with one another or groups -- all within a single application.

Platforms & Frameworks

Frameworks

- Electron uses Chromium and Node.js so you can build your app with HTML, CSS, and JavaScript.
- Electron is an open-source project maintained by GitHub and an active community of contributors.
- Compatible with Mac, Windows, and Linux, Electron apps build and run on three platforms.

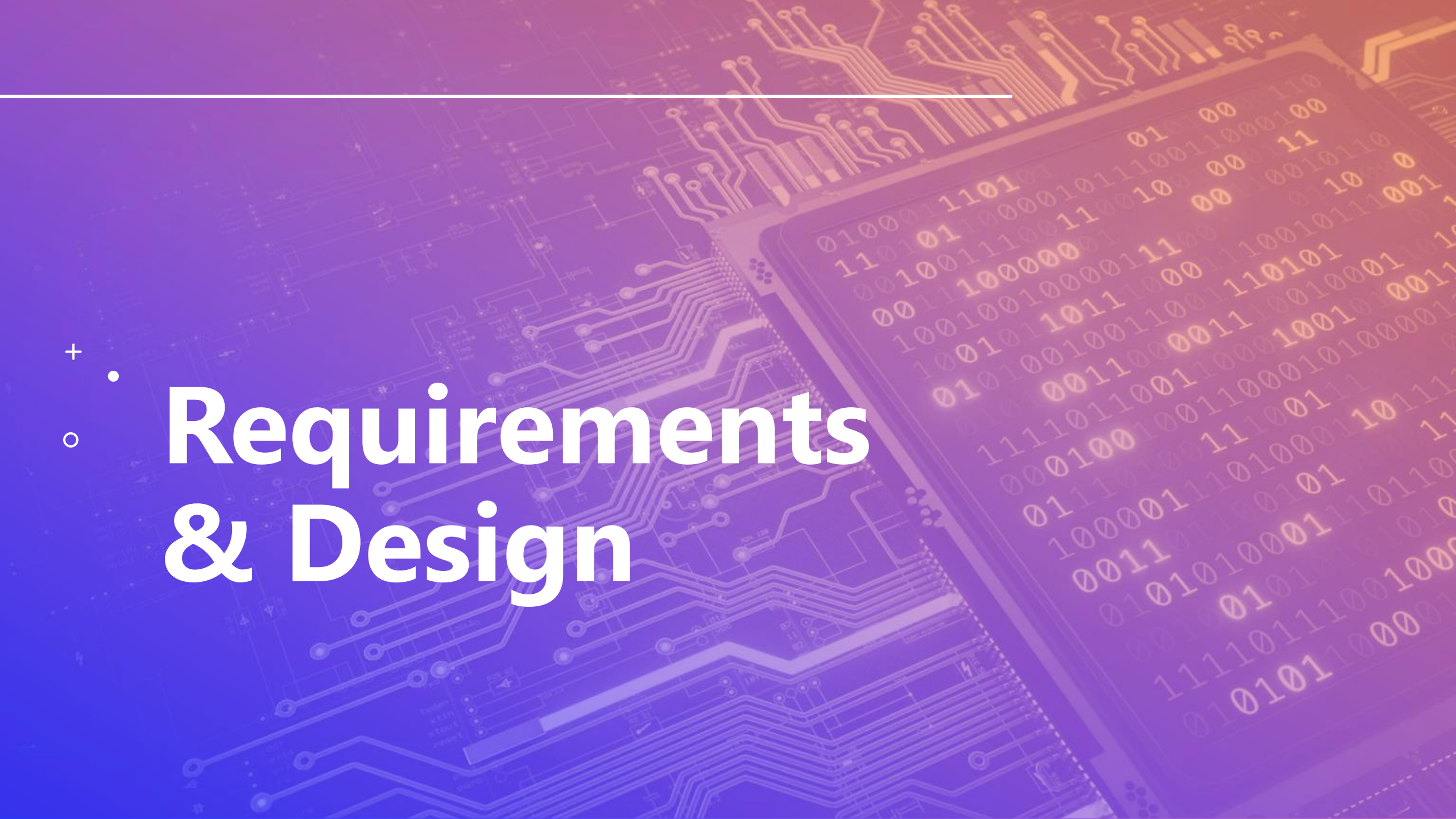
- React makes it painless to create interactive UIs.
- Design simple views for each state in your application and React will efficiently update and render just the right components when your data changes.
- Build encapsulated components that manage their own state, then compose them to make complex UIs.

- Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.
- As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications.
- Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Frameworks Cont.

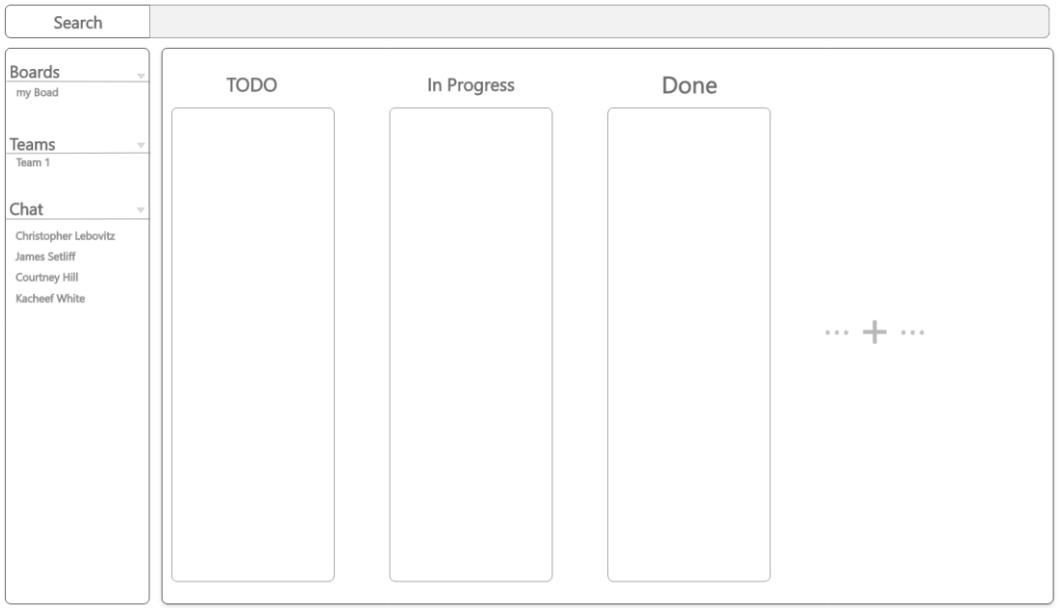
- MySQL is an open-source, relational database management system.
- MySQL Database Server is fast, reliable, scalable, and easy to use.
- MySQL Server works in client/server or embedded systems.

- express
- socket.io
- mysql
- react
- react-dom



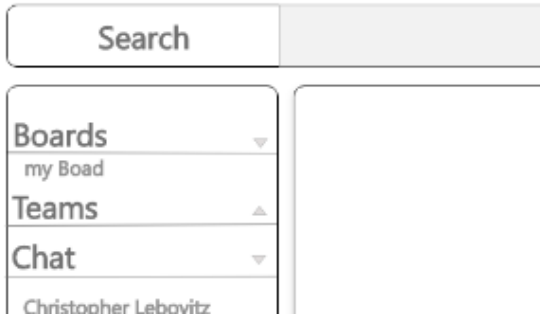
Requirements & Design

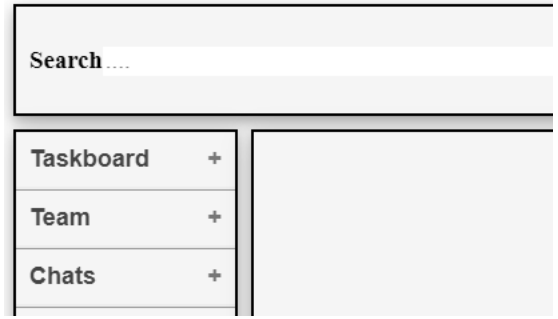
Visual Task Board

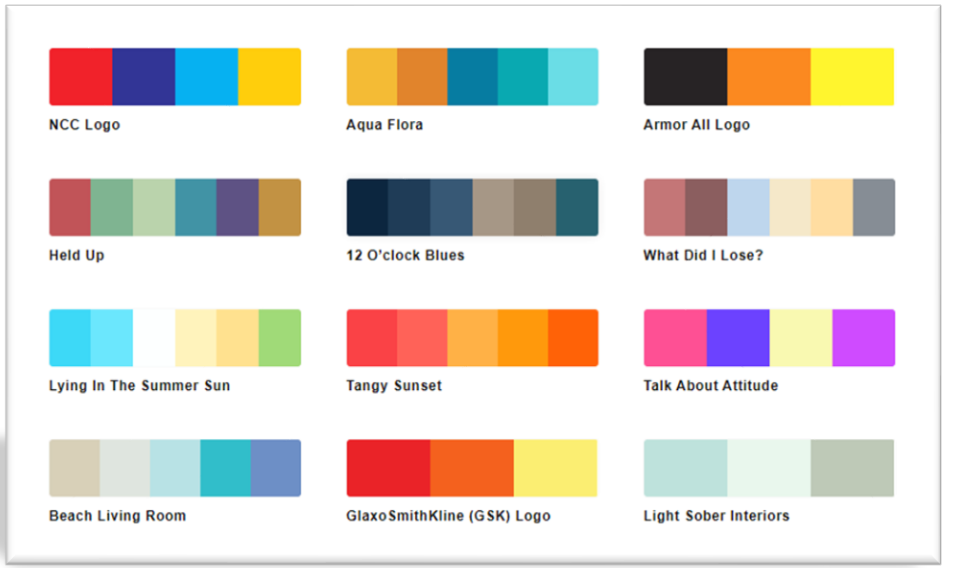


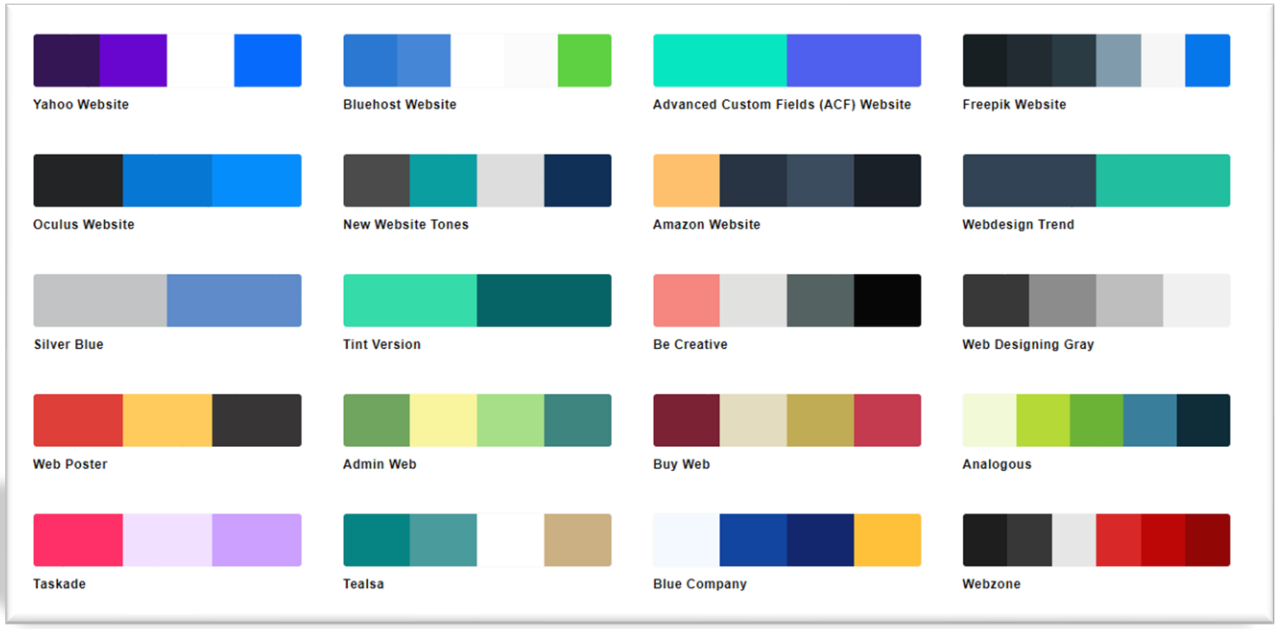
- Comprised of columns that contain “Task Cards”
- Columns are renamable to meet team’s needs
- Task Cards may be moved between columns to track progress
- Columns and Cards may be added or removed

Design and Color

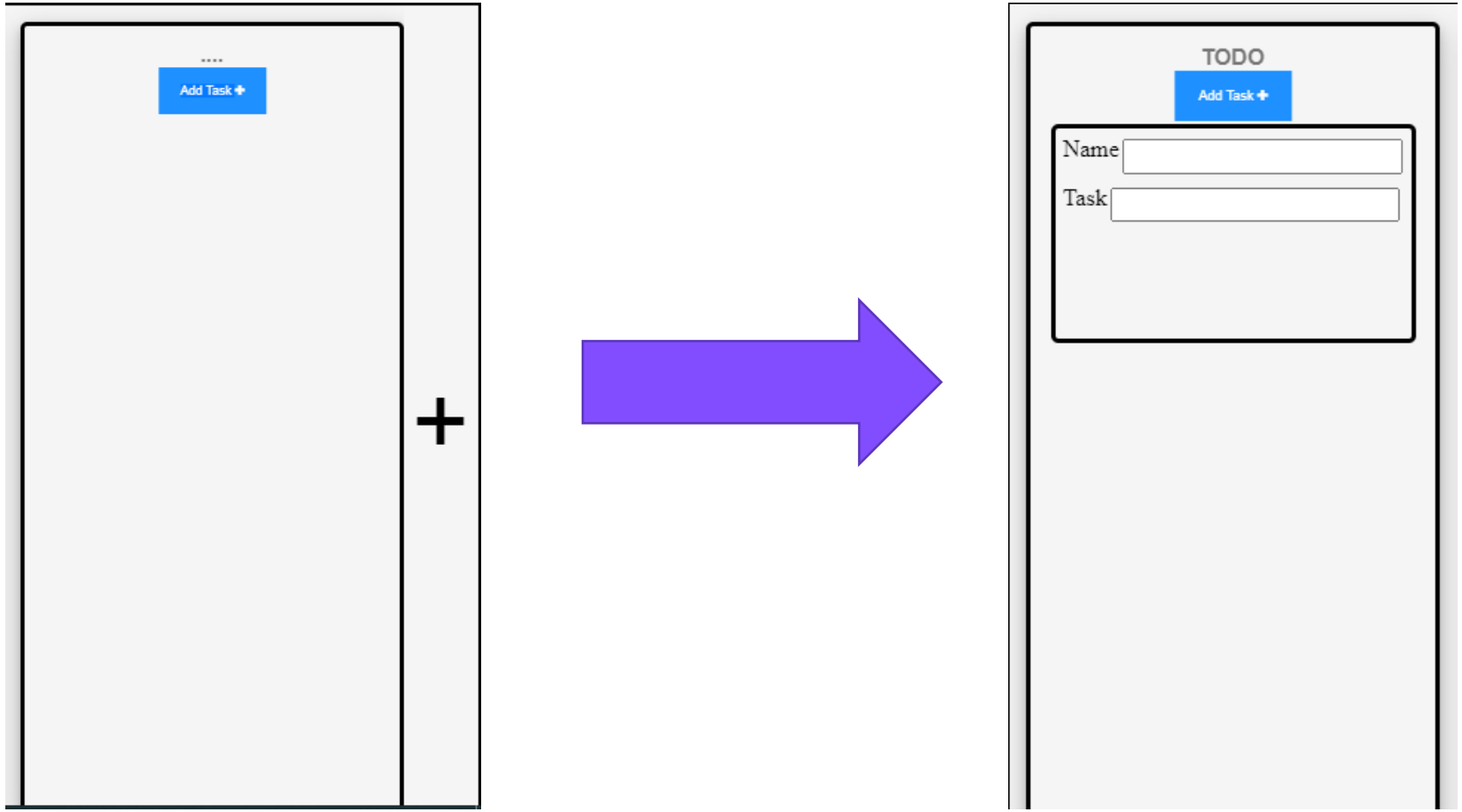






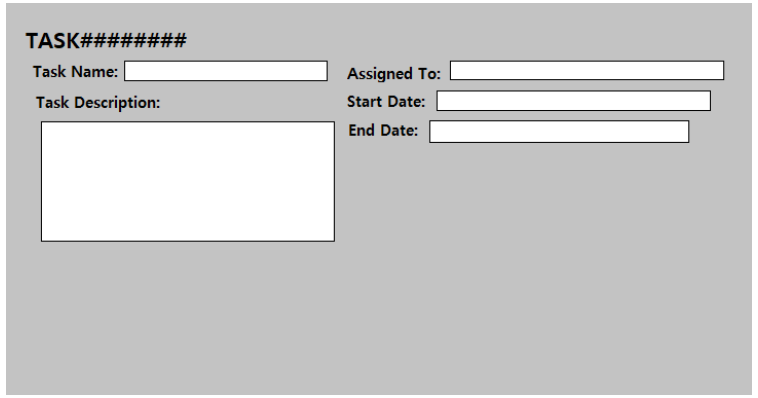


Column and Card Layout



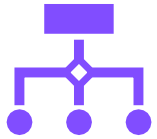
Task Tracking

- Automatically assigned a unique number
- Contains name, number, description, assignee, start date, and end date
- Title and description are indexed in the database making them searchable

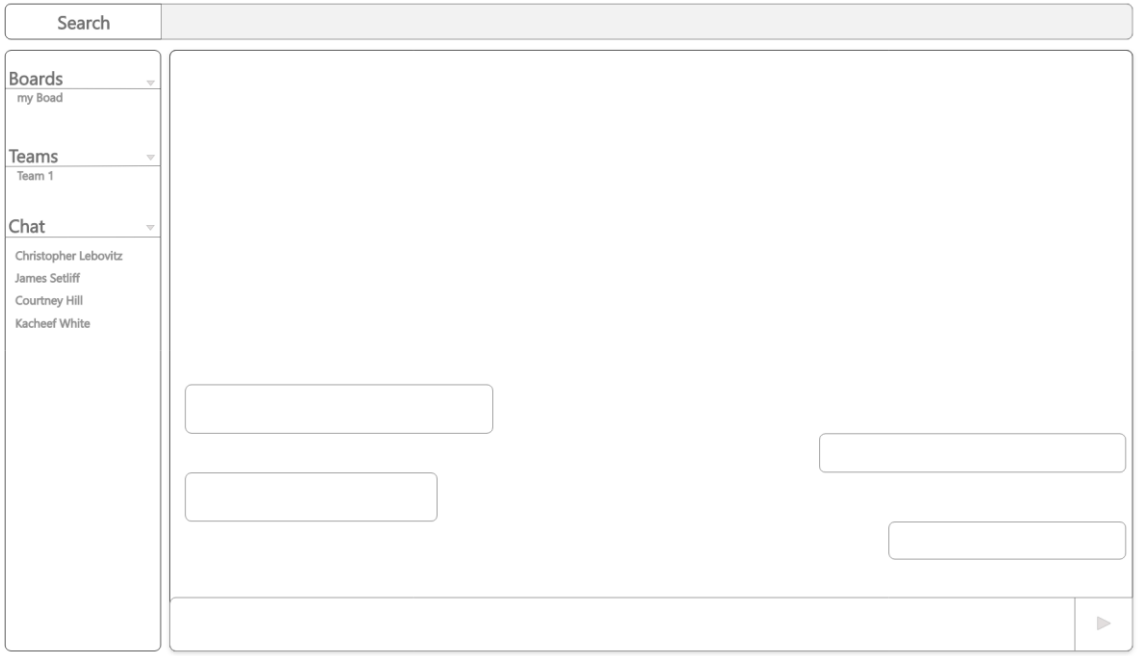


Real-time Chat

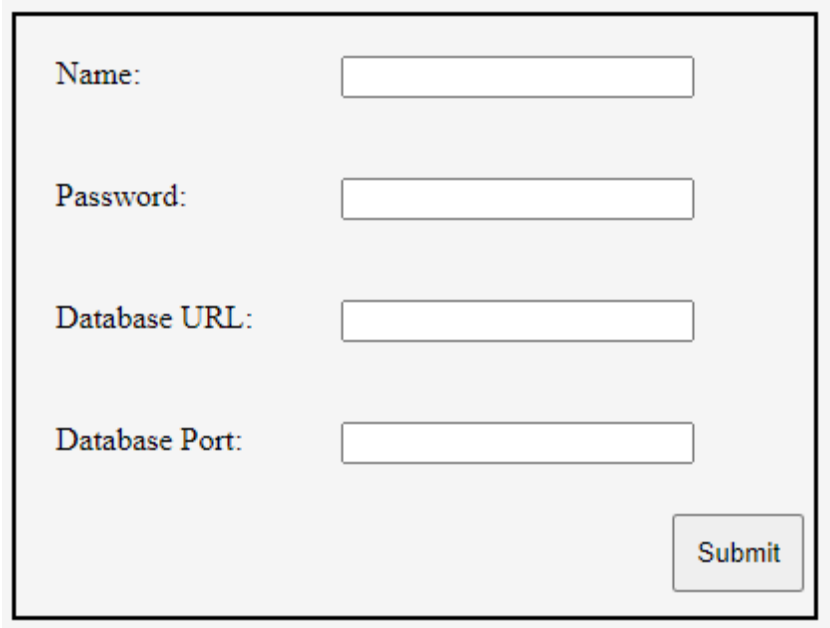
Chat supports communication amongst team members via persistent, real-time chat messaging



Messages are inserted into the database at the same time they are sent to other clients via use of socket.io event manipulation



User Settings

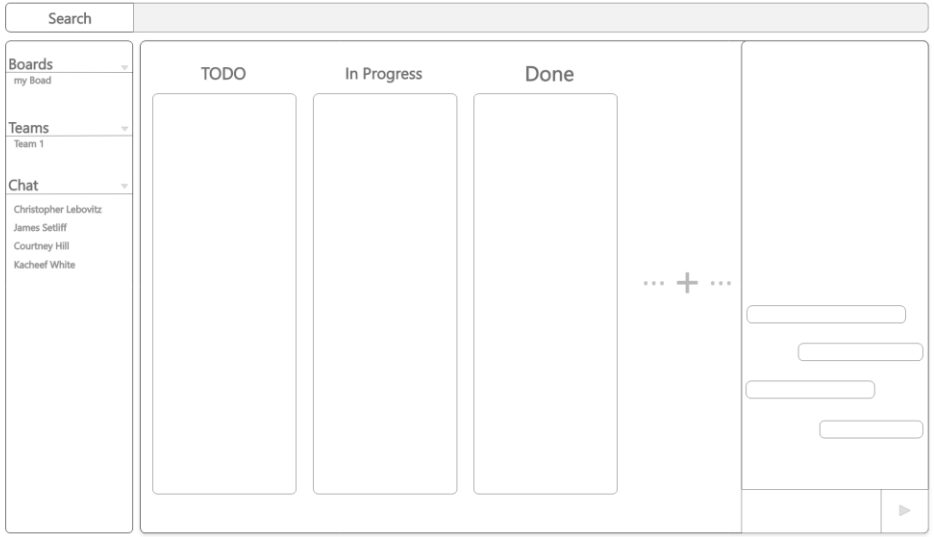


- Values entered on the settings form (username, IP addresses, etc.) will be used throughout the application
- Settings will be written to file and stored on the user's machine in JSON format

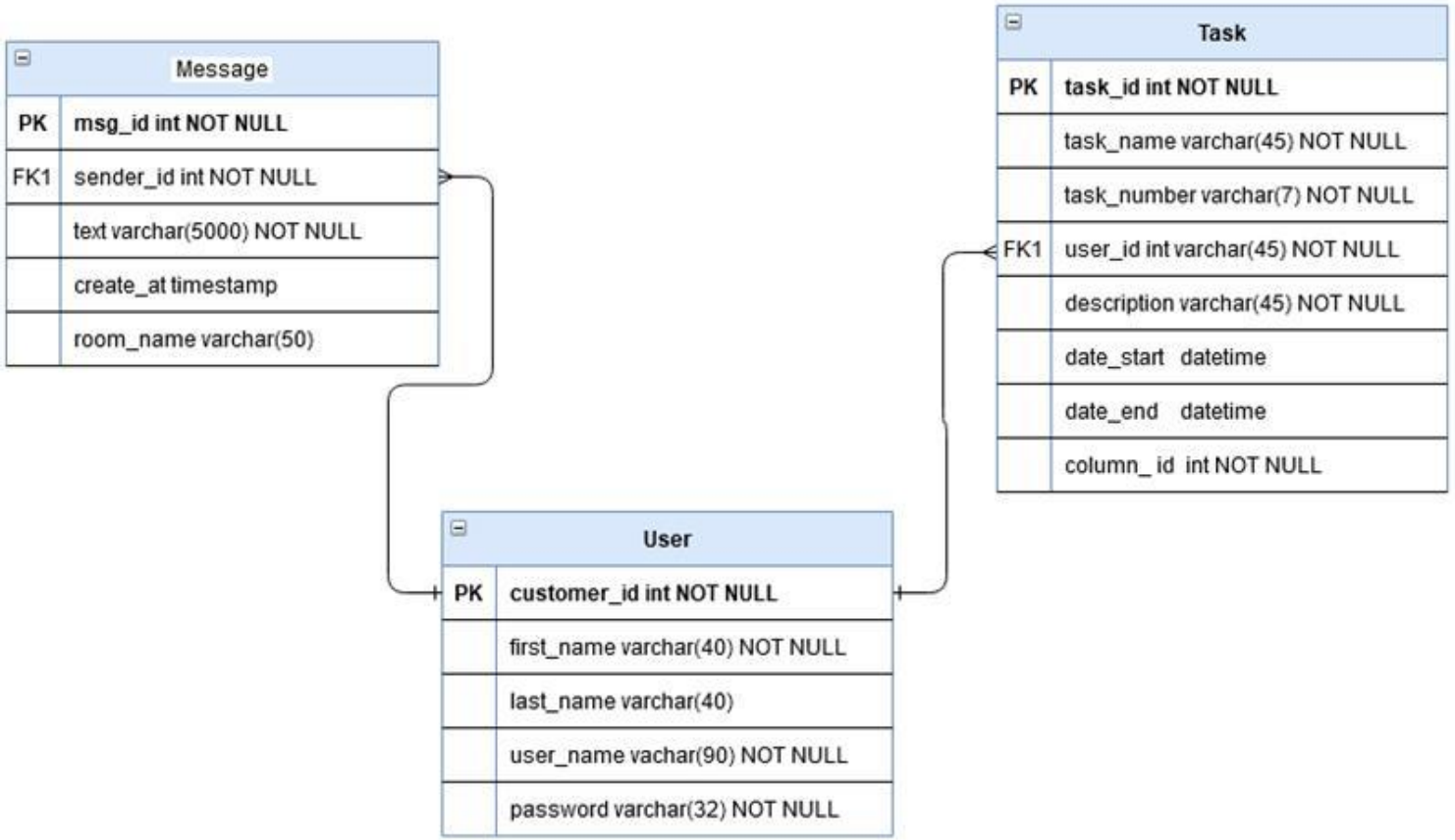
Search Functionality

Search bar queries against all full-text indexes in the database and returns any relevant results

Results will be displayed in a drop-down below the search bar and link to the relevant record in the correct view

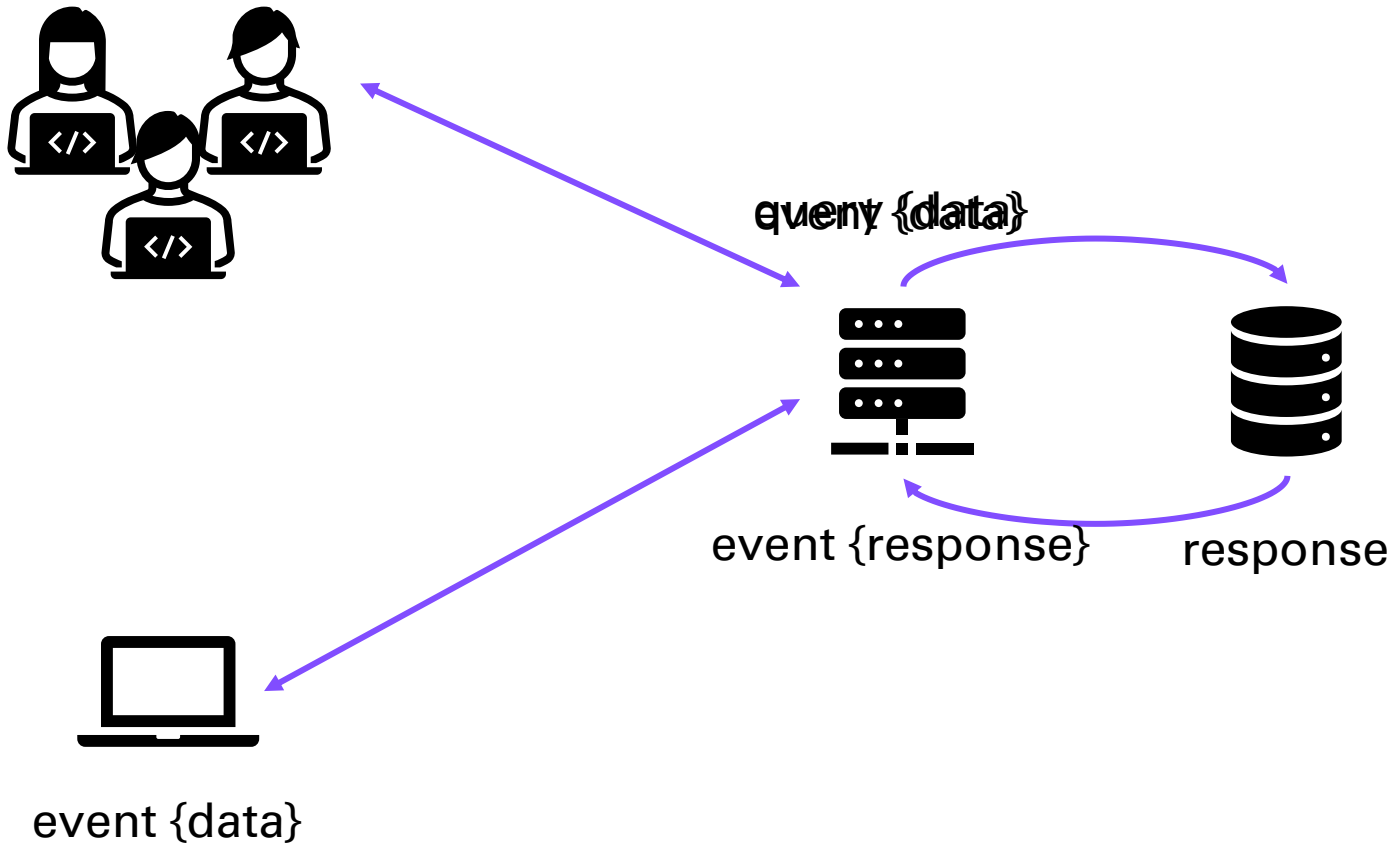


Database Design

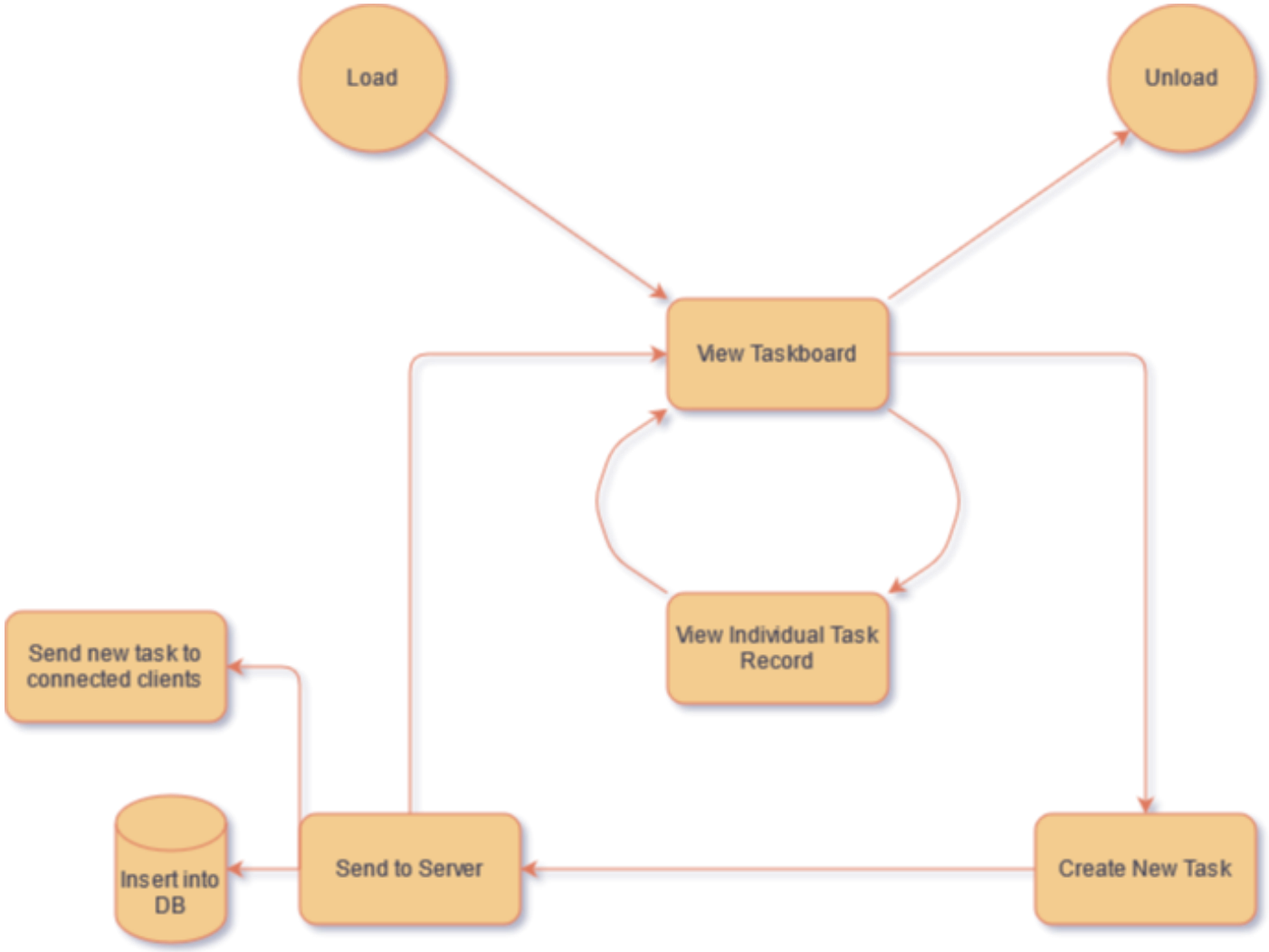


Implementation

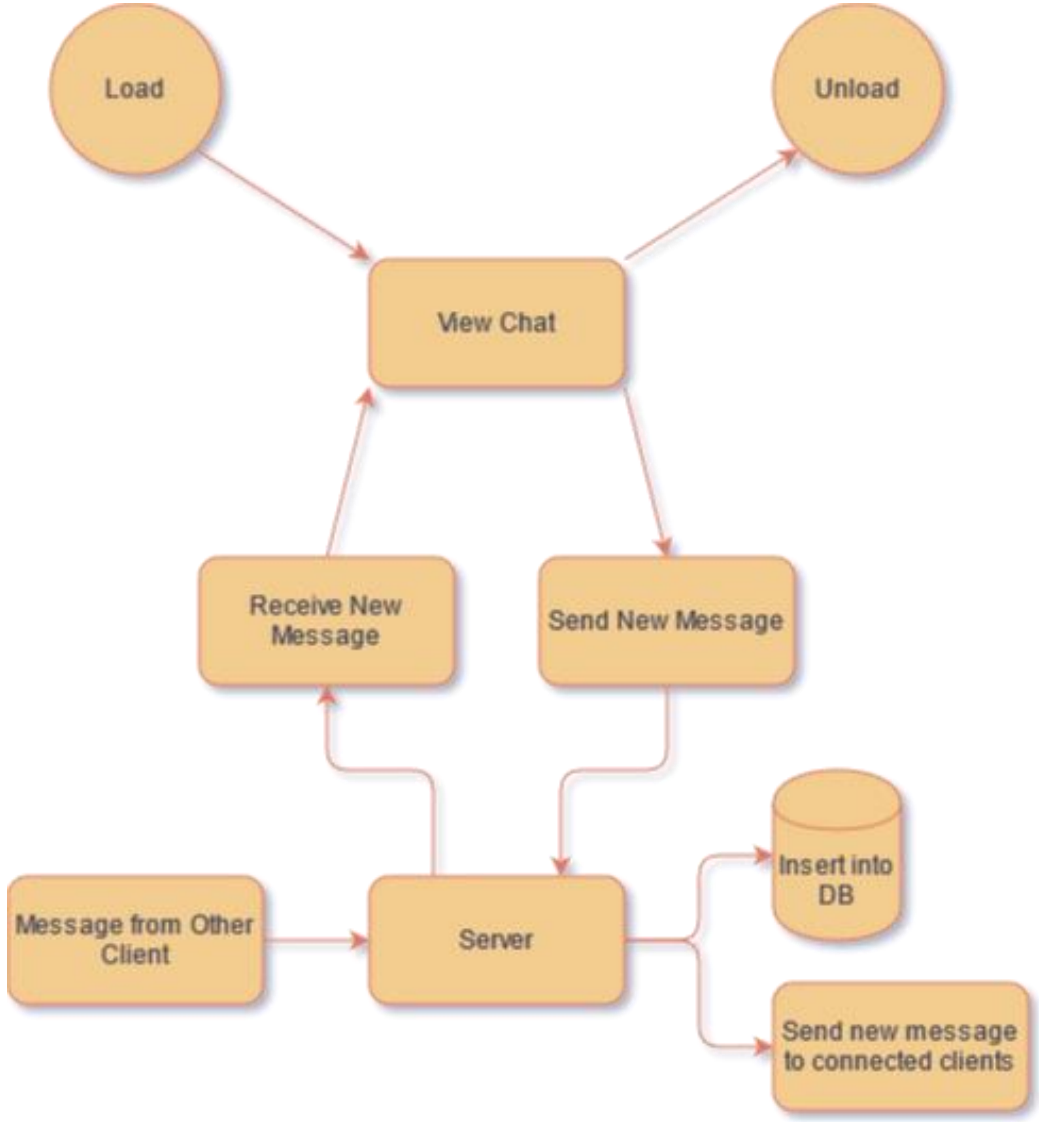
Client-Server Interaction



Task Board State Diagram



Chat State Diagram



Closing Arguments

Test Outline

Requirement #	Description
R1.0	User can launch program from any supported platform.
R1.1	User sees all applicable fields on Task form.
R1.2	User can create, update, and delete Task records.
R1.3	User can view Visual Task Board.
R1.4	User can create, rename, and delete Visual Task Board columns.
R1.5	User can move Task Cards between columns on Visual Task Board
R1.6	User can participate in real-time chat with other team members.
R1.7	User verifies that chat messages persist through repeating launches of the application.
R1.8	User can navigate to Settings page.
R1.9	User can edit values on the Settings page form.
R1.10	User verifies that Settings persist after saving.
R1.11	User can enter search queries in the search bar located at the top of the application.
R.12	User verifies that search results take them to the applicable view and/or record.

Problems and Challenges

- Lack of familiarity with languages/frameworks
- Transitioning to React for front-end
- Integration
- Time constraints
- Implementing full-text search

Summary

It is our opinion that not everything belongs in the cloud, and subscriptions aren’t always the answer.

DevOP will provide small teams the ability to track task progress and communicate without the need for multiple programs or sharing their data with Big Tech.

Questions?