

実行方法

- Websocketフォルダに移動した上で
go run main.go
を実行する。

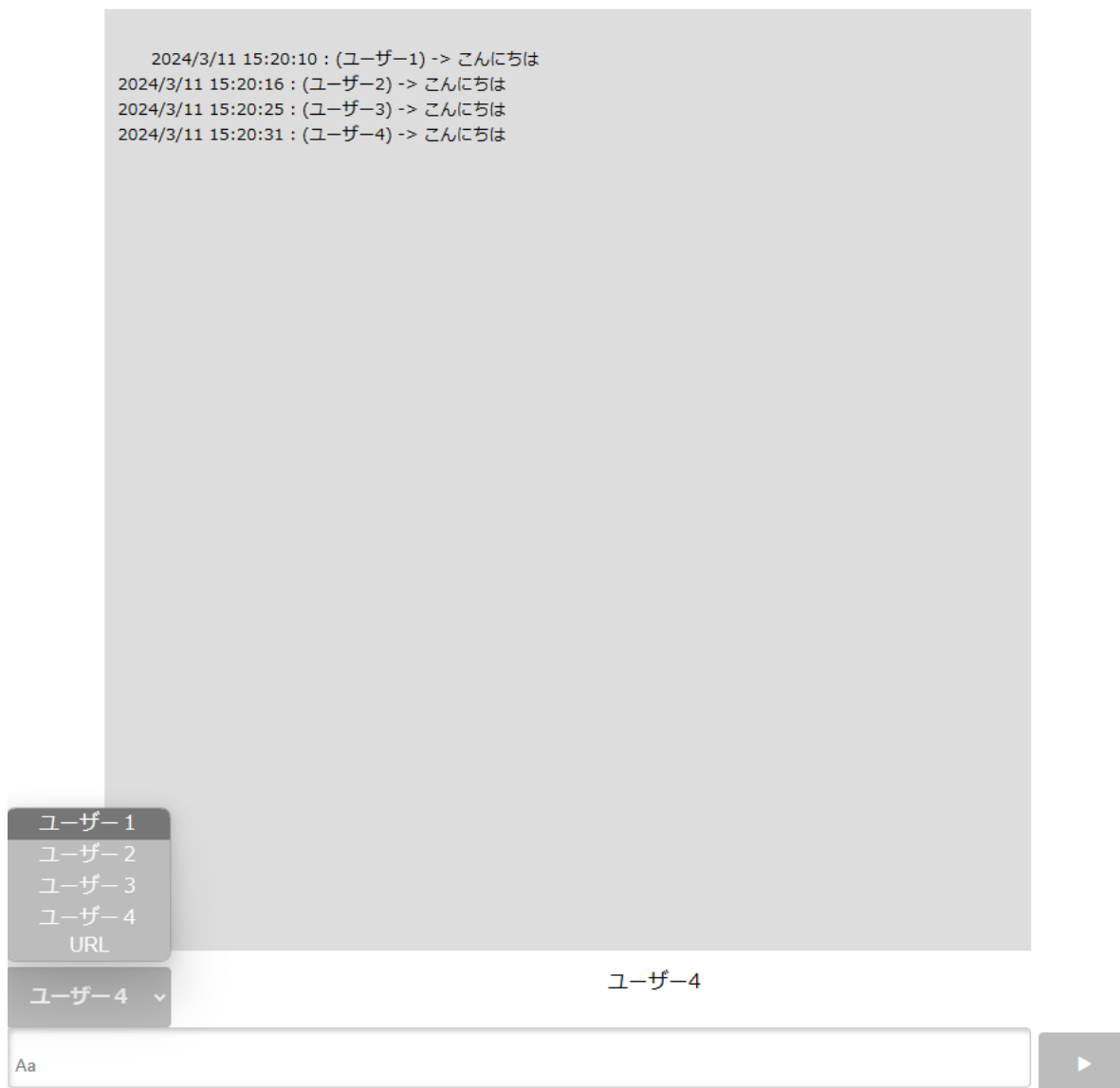
発表後の改良点

- メッセージの送信を複数回行えるように改善
- ユーザー選択のボタンで画像のURLを選択できるように変更

妥協点

- チャット画面上に画像を表示しようとする、メッセージと画像の共存が出来なくなってしまったため、画像の表示場所をチャット画面の下に設定した。

Chat App



- 変更点
 - ユーザー選択した上でのメッセージ表示
 - 時間を取得し、表示
 - 複数回のメッセージの送信
 - 画像のURLもここで選択可能
-
- 妥協点
 - メッセージの一行目と送信ボタンの位置がずれてしまっている。

使用したライブラリ

- gin

httpサーバを起動し、ルーティングを設定

- Websocket

Webにおいて双方向通信を行うために必要だったので使用。

参考文献の追加

<https://qiita.com/TetsuyaFukunaga/items/4c83a8dedd34e65ffbd9>

GoでWebSocketを使いチャットサーバー構築

https://zenn.dev/show_yeah/articles/bece10823d182c

【入門】Goでwebsocketを使ったリアルタイムなチャットを実装してみた

チャットアプリの開発 (最終発表時点)

沼田大悟

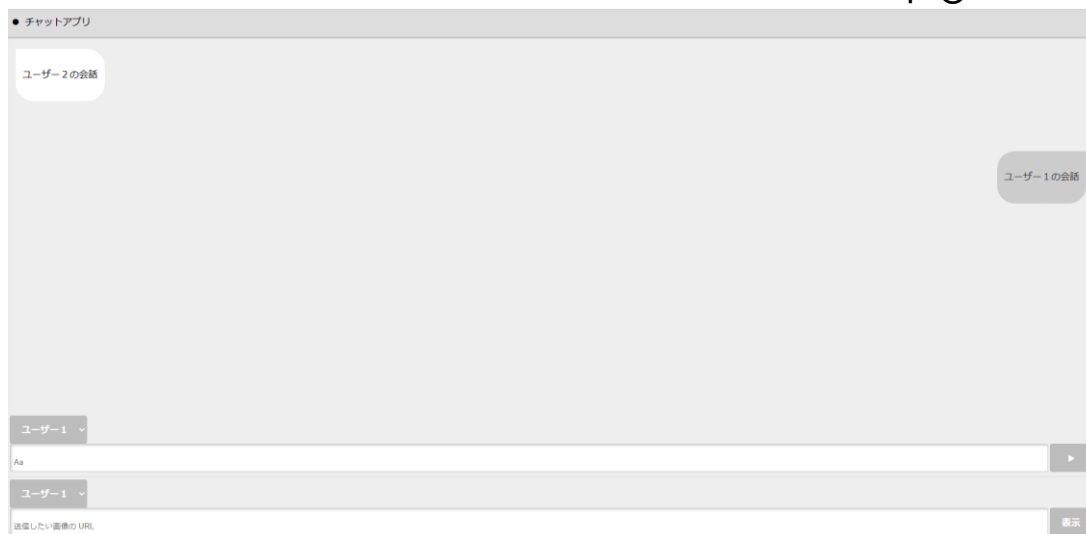
概要

- ローカルホスト上でチャットを行うソフトウェア
- ウィンドウ幅の自動調整を行う
- チャットを送信しあう
- ネット上にある画像のurlを貼ることで、画像を送信する。
- 複数のユーザーで使い分ける

ウィンドウ幅に合わせて拡大

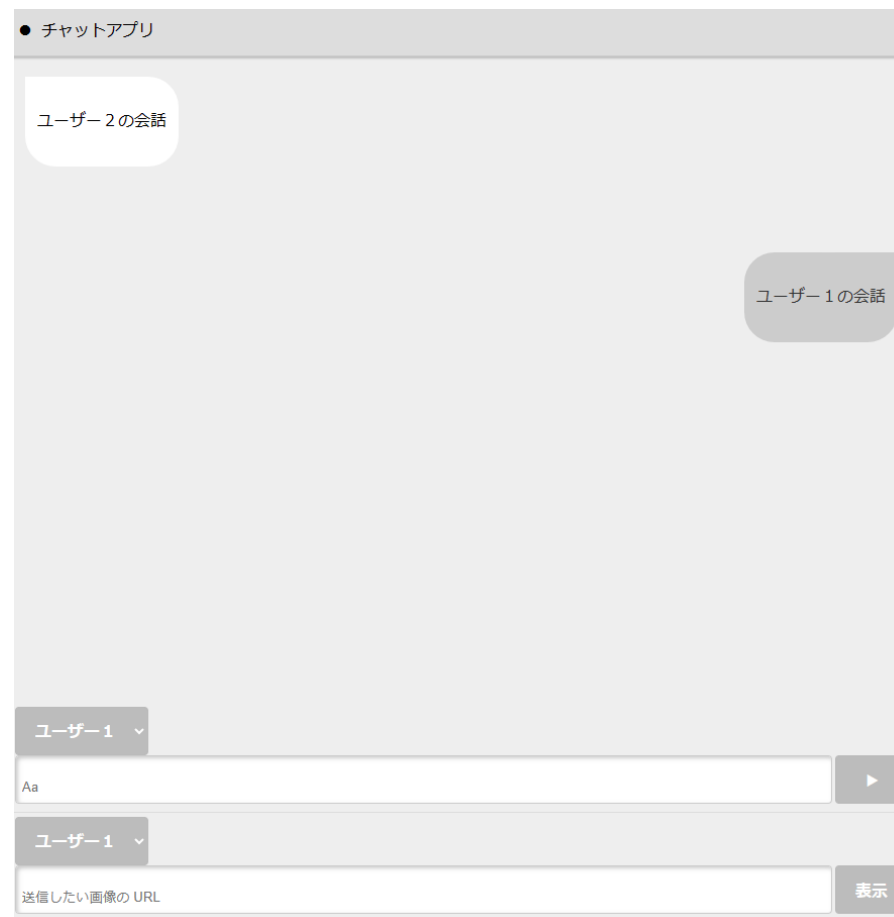


スマートフォン

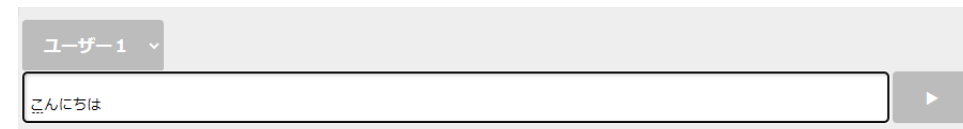


PC

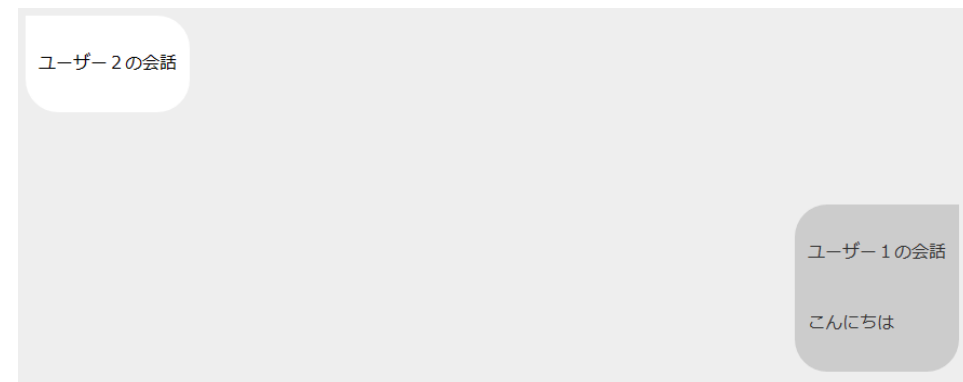
タブレット



チャットを送信



ユーザーを選択した状態でメッセージを送信する



インターネット上の画像を送信

ユーザー 1 ▾

送信したい画像の URL

表示



送るユーザーの選択が可能。
画像のurlを入力する。
urlが入力された状態で表示ボタンを押すと画像が出力される。

ユーザー 1 ▾

`https://camo.githubusercontent.com/89e3fc81036629031e787c8d09cb08bb50993b1b4c6f6ecd72e4a7c41f398c5b/68747470733a2f2f6`

表示

画像を送信する



画像のurlから情報を取り出し、UI上に画像を表示する

画像のサイズは400×200に固定している

urlの例(Gopherが複数):

<https://camo.githubusercontent.com/89e3fc81036629031e787c8d09cb08bb50993b1b4c6f6ecd72e4a7c41f398c5b/68747470733a2f2f676f6c616e672e6f72672f646f632f676f706865722f6669766579656172732e6a7067>

工夫した点

- Htmlファイルとcssファイルの組み合わせ
- コマンドプロンプト上でも操作が見える
- UIの改良

改善点

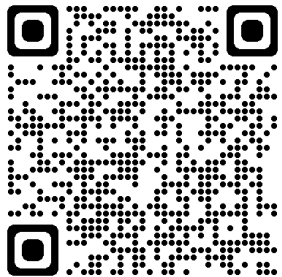
- チャットを記録する機能の作成
- チャットを送信した日時の記録
- チャットと画像のユーザ選択を同時に行えるようにする。
- ローカルホスト上でしか、チャットができないためオンラインでも公開できるようにする。

参考文献

- <https://qiita.com/sususuAI/items/ad4806c086b02a25d45d>



- <https://astaxie.gitbooks.io/build-web-application-with-golang/content/ja/04.1.html>



- <https://zenn.dev/tomi/articles/2020-10-07-go-web5>

