Handover

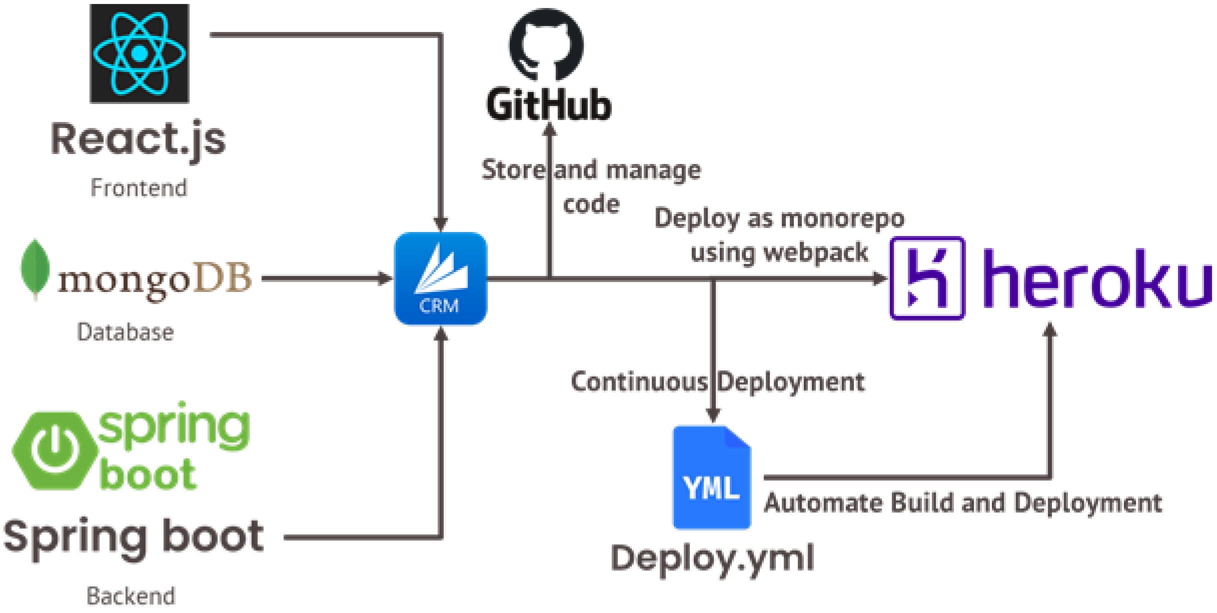
Our product CandHCRM supports a variety of functionalities, e.g. contact management, mass mailing, online chat, voice/video call, screen share, meeting scheduling, and real-time notification.

Architecture

The backend server uses Spring Boot framework and provides restful API service that the React frontend uses. The database used is MongoDB. The project is deployed to two Heroku apps from a multi-language monorepo. With the secret key HEROKU_API_KEY set in our GitHub reposiotry, the file `.github/workflows/deploy.yml` runs automatic deployment to Heroku once there's a push in main.

GitHub: https://github.com/C-and-H/comp30022

Heroku URL: https://crm-c-and-h.herokuapp.com



Requirements

1. Download node.js and npm from https://nodejs.org/en/download
2. Download IntelliJ IDEA from https://www.jetbrains.com/idea/download
3. Update JDK to version 16 or later.
4. Has a valid Gmail and MongoDB account.
5. Packages requirements: https://github.com/C-and-H/comp30022/network/dependencies

Run

Server:

1. Open `/backend` folder in IntelliJ IDEA, and load `pom.xml` to link maven project.
2. Create 5 MongoDB collections that follows `/backend/src/main/java/candh/crm/model/*.java` in `crm` database.
3. Edit run configuration > Environment variables > Apply.
4. Run `CrmApplication.java`.

App:

1. Go to directory `/frontend`.
2. Create a new file `.env` of environment variables, or run `echo REACT_APP_API_URL=http://localhost:8080 > .env`.
3. Run `npm install` to install packages.
4. Run `npm start`.
5. Browser visits `localhost:3000`.

Environment Variables:

Frontend

```
REACT_APP_API_URL=http://localhost:8080
```

Backend

```
MONGO_USERNAME=<mongo-usrname>
MONGO_PASSWORD=<mongo-pwd>
MAIL_ADDRESS=<gmail-address>
MAIL_PASSWORD=<gmail-pwd>
APP_URL=http://localhost:3000
JWT_SECRET=crm
```

Design

See https://bingzhej.atlassian.net/wiki/spaces/CH/pages/12157334/Design+Diagrams for details.

Deployment

See https://bingzhej.atlassian.net/wiki/spaces/CH/pages/10452999/Deployment for details.

Highlight

Contact Relations

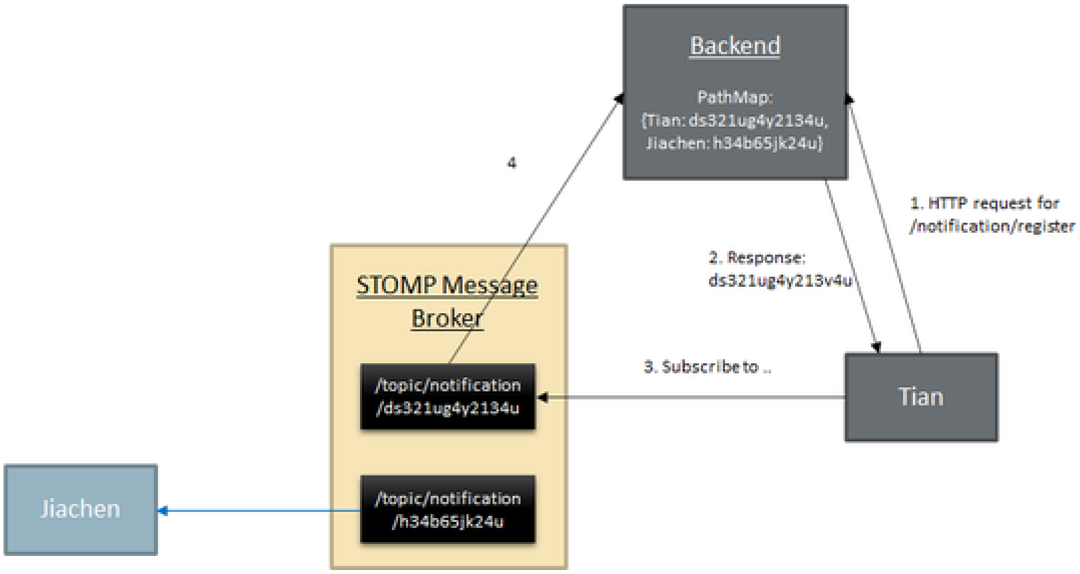
User Accepted	User Ignored	Friend Accepted	Friend Ignored	Results
T	F	T	F	Friends
T	F	F	F	user sent request but not yet responded
F	F	T	F	user received request but not yet responded
F	T	T	F	user received request and declined
T	F	F	T	user sent request and was declined
F	F	F	F	one of user and friend sent request and cancelled
NULL	NULL	NULL	NULL	user and friend has never sent request to each other

Real-Time Logic

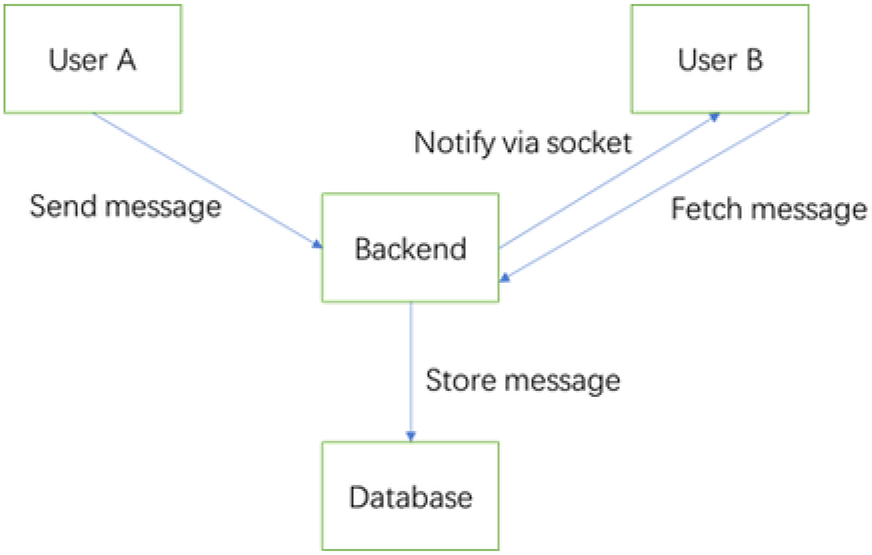
To implement this functionality of real-time notification, we utilised STOMP websockets for connection.

Firstly, a frontend sends an HTTP request to backend to register itself, and the backend responds with a path for the frontend to subscribe to and stores the key-value pair of that frontend user and the path. Then the frontend subscribes to that path.

When an event like sending a new friend request occurred, the backend can let the STOMP message broker to push the information to that specific user’s frontend.



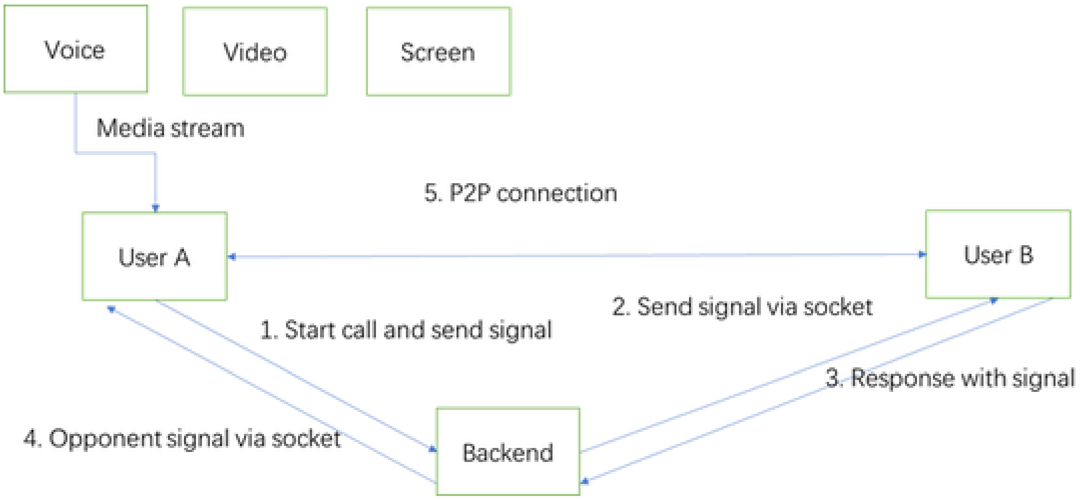
Chat Logic

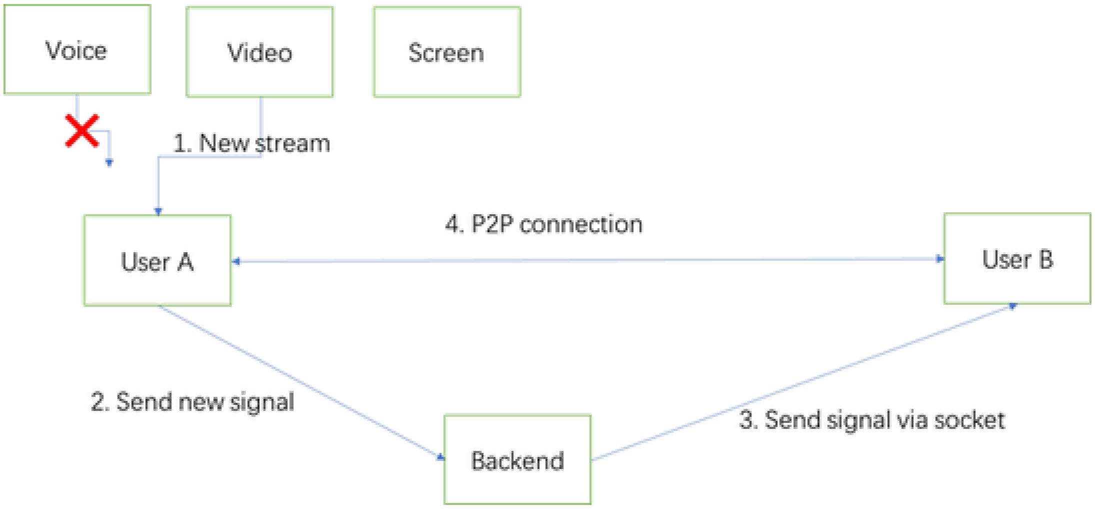


Call Implementation

Our CRM established P2P connection between two user so they can talk to each other. This requires both user having opponent's signal. So socket is used to exchange signals between user.

Our call can also switch between different media streams, and switch will create a new signal. Socket will forward the new signal to opponent to create a new P2P connection.





Future Improvement Insights

1. Missed call notification.
2. Connect with real meeting creation API (e.g. Zoom).
3. Users can search participants' names when scheduling a meeting.
4. Deleting friends should also delete the chat history between them.
5. Forget password.
6. The current logic of contact relations makes it fast for an update but slow for search, while there are more searches than updates.
7. Multiple accounts could be allowed in a single browser.
8. Meeting re-schedule.
9. Call does not record voice when sharing the screen. Tried to add one more voice stream, but this would affect the screen to show on the opponent side.