Course project group information
Group Name: NicePeople

Name	Student number	Tuni email
Rafin Akther Utshaw	152253048	rafin.utshaw@tuni.fi
Minh Hoang	152103143	minh.hoang@tuni.fi
MD Iftekhar Hossain Tanveer	152258658	Iftekhar.tanveer@tuni.fi

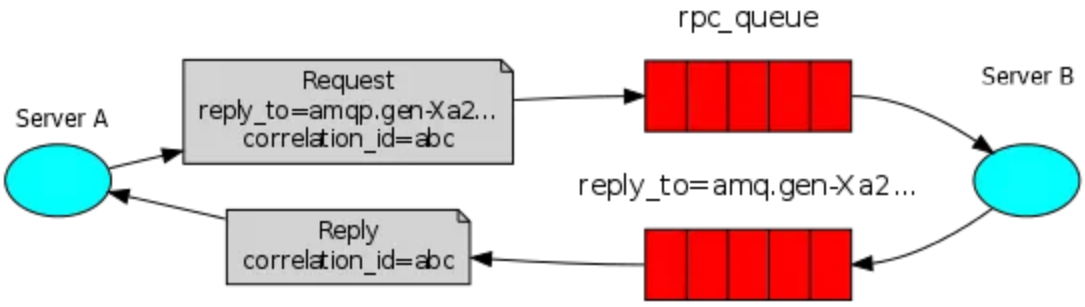
Gitlab repo URL: https://course-gitlab.tuni.fi/compcs510-spring2024/nicepeople

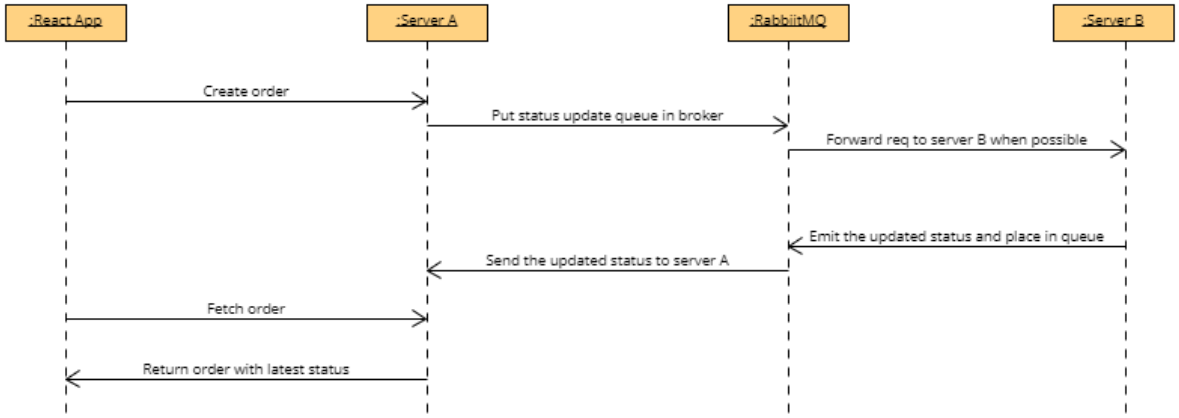
Working during the project

Weekly time commitment:
Rafin Akther Utshaw - 4h
Minh Hoang - 4h
Tanveer - 4h

Time table for the project:
Research -> 20/3/2024 to 31/4/2024
Design -> 1/4/2024 to 10/4/2024
Implementation -> 11/4/2024 to 28/4/2024

System architecture
The system architecture is designed to provide a seamless sandwich ordering experience. It consists of a frontend built with ReactJS for client interaction and two servers, Server A and Server B, implemented using Node.js. Communication between servers is facilitated by RabbitMQ, a message broker.





Technologies

Server side: NodeJS for developing server A and server B. For communication between the two servers RabbitMQ message broker is being used.
Client side: ReactJS is being used for the client to communicate with the server.

The components of the system are placed in the following directories within the repository:

- Frontend: `/frontend`
- Server A: `/backend/server-a`
- Server B: `/backend/server-b`
- RabbitMQ Configuration: `/rabbitmq`

Running the testing the application

Steps to run and test the app goes here…

To deploy the system on your own computer for testing:

- Clone the GitLab repository.
- Navigate to the root directory of the repository.
- Follow the instructions below

Testing the System:

To run and test the application, follow these steps:

To Run Server A:

```
cd server-a
npm install
node index.js
```

Ensure that RabbitMQ is running and properly configured for communication.
To Run Server B:

```
cd server-b

npm install

node index.js
```

To Run Frontend:

```
cd frontend

npm install

npm start
```

Progress of Group's Work:

- Research Phase: During the research phase, the group explored various technologies and architectural patterns suitable for the project requirements. Research included studying ReactJS, Node.js, and RabbitMQ.

- Design Phase: The design phase involved creating architectural diagrams, defining component responsibilities, and planning communication protocols between components. Currently we are in the process of defining our components and the communication between the servers and clients.

Learning During the Project:

- Rafin Akther Utshaw: Learned about message queue systems and their implementation in distributed systems.
- MD Iftekhar Hossain Tanveer: Gained experience in frontend development with ReactJS and improved understanding of asynchronous communication between frontend and backend.
- Minh Hoang: Enhanced knowledge of Node.js server development and gained practical experience with RabbitMQ message broker.

Additional Considerations:

- *Any extra documentation or considerations go here...*