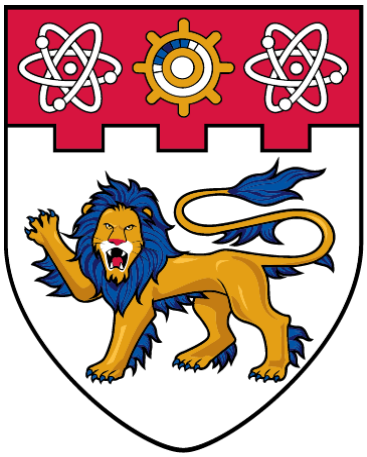


NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

IM3080 Design and Innovation Project

(AY 2020/21 Semester 1)

Group Project Report

Title: ReSharePeace Cooking webapp

Github: https://github.com/zerriet/Food-Sharing-WebApp-DIP.git

Submitted by: Group 2

Supervisor: Professor Chua Hock Chuan

Table Content

List of illustrations

Figures

1. Background and Motivation

With COVID-19 evolving into a global pandemic, activities like dining out were disallowed whilst home cooking / food delivery options were encouraged instead. Thus, as local lockdown measures were prolonged, home cooking became an increasingly popular activity that not only serves as a novel leisure hobby, but also supplies meals for families.

It is worth noting that various virtual-conference platforms were developed during this period. However, almost all of these platforms focused on supporting work or educational purposes, rather than supporting social interactions via the carrying out of group leisure activities like cooking. As such, even though there was an increase in demand for people to be able to share and learn how to cook, there were no existing platforms that fill this void. Therefore, our group hopes to develop an online platform to address this and target a market who have just developed cooking as a hobby over this quarantine period.

2. Objective & Target Audience

- To create a single platform that enables users to be able to share their recipes on our webapp and learn interactively in real time, through a live cooking class. In addition, this platform also serves to provide another revenue stream for cooks, both amateur and professional, who wish to share their knowledge. Thus our target audience are mainly:

    - Learners who want to learn how to cook or find new and interesting cooking recipes online as well as for people who want to pick up a new skill during the quarantine period

    - Cooks (professional cook businesses) who are willing to share their recipes online, host online cooking classes and have a form of side-income

3. Review of Literature / Technology

Following are the libraries / API needed for the webapp coding.

java IO	Used in implementation class such as FileOutputStream FileReader
java.sql	Connection between java IDE and MySQL database
javax.servlet	Interfaces that describe and define the contracts between a servlet class and the runtime environment

Following are the libraries needed for the video application coding and are all open source which allows the flexibility between the connection of web clients.

Npm	Package manager to install and run libraries for video / js / bootstrap
NodeJS	Runtime environment for single event-driven applications
ExpressJS	Web application framework that implement NodeJS
Embedded JavaScript	Template which implements HTML with JS
UUID	Enable user and server to have unique identifier and no overlaps
Socket.io	Real-time, bi-lateral communication between web clients
Peer	Enables real-time communication capabilities in the browser
webRTC	JS library that implements web RTC

4. Design and Implementation

4.1 Design Consideration / Choice of components

Parameters	Design Considerations
Login / Registration Page	- Avoid / Alert when registering with repetitive username - Verfies and authorise access to profiled webpages - (i.e. profile page) - User-Friendly Interface (intuitive login process) - User contact information (e.g. email) - Password requirements (i.e. length) - Minimalist background
User Home Page	- User-Friendly Interface (intuitive tabs for main functions) - Live Chat bot giving guidance on navigation (i.e. guides to book classes) - Autocomplete suggestion for search query (based on list of recipes) - Recommended / Trending recipe - Drop down menu (top right corner tied to all pages) - Sort / filtered list of searched recipes - Minimalist background
User Profile Page	- Clear overview of uploaded recipes (picture oriented) - Upload photo files from user's personal device (recipe photo) - Resize to standardise recipe image - User-Friendly Interface (intuitive tabs for favorites, posted recipes and classes) - Minimalist background
User Account Setting	- Edits profile account information - Updates database and profile page accordingly
Video Application	- Different tiers of access to functions (Host - create video conference and Participants - mute / stop video) - Rooms have unique identifiers to prevent overlap - (Mute / Unmute) voice and (On / Off) video functionality - Appropriate video size (large screens for host) - Donation system (live counter for points / money donated to cooks)

4.2 Final Design (with block diagrams)

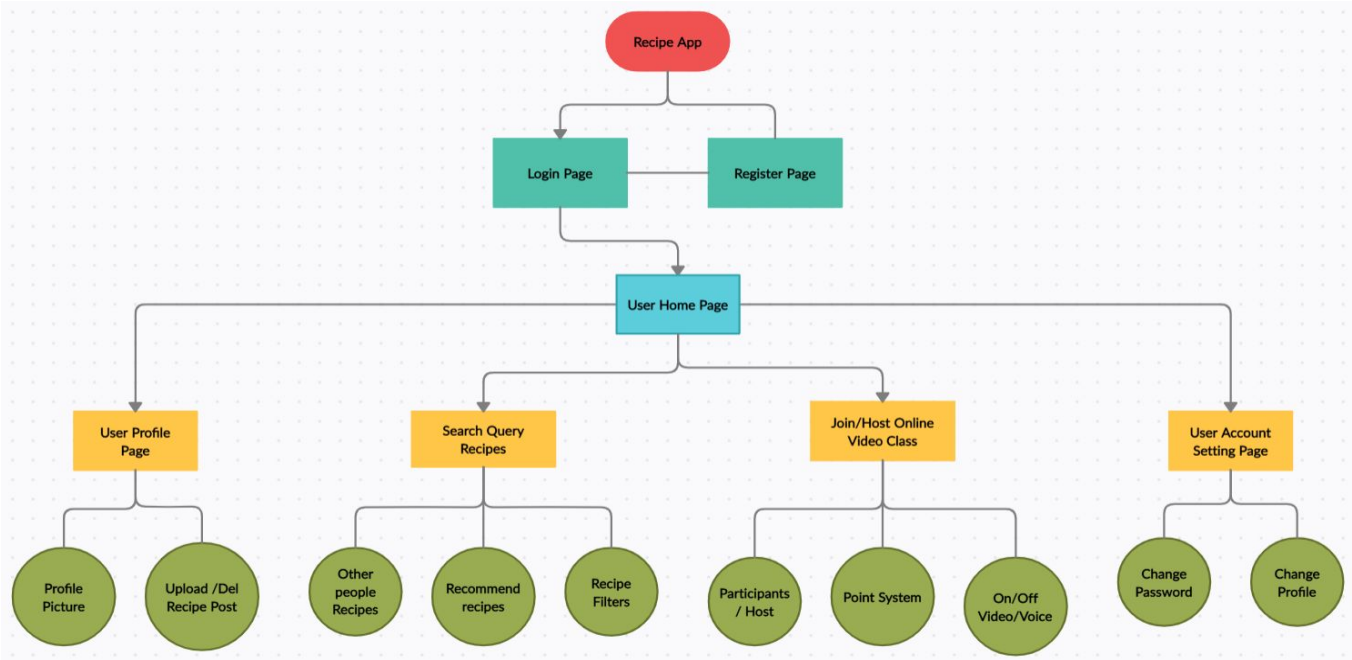


Figure 1: Block Diagram for Cooking Recipe Website

4.3 Discussion & Implementation (with photos)

During the initial brainstorming stage of this project, several ideas were presented. Among the many ideas, we have come to a conclusion to focus on the recipe and cooking webapp, as a group.

Our team started with planning and analysing possible features to be included in our webapp for recipe sharing and class. We started by listing the functions that could be applicable to our webapp like the types of filter / sorting features (as seen in the consolidated list of functions below). Then, we decided on the functions that are more essential to our webapp such as adding recipes, edit and view profile. Based on these decided features, we separated the workload into: the databases / ER block diagrams (used as a guide), wireframe / frontend developers (to develop designed static webpage), backend developers (to code decided essential functions).

Basic features (learners)

1. Review recipe / class (comment, rate)
2. Favourite recipe / class (bookmark)
3. Book class (online, not live class)
4. Share recipe (of others learners)
5. Add / edit recipe
6. Search bar - for both recipes and classes
7. Filter
8. Account settings - admin stuff (change password
9. Profile settings - beautify your account (upload picture, bio etc)

Basic features (cookers)

1. Host class
2. Add / edit class
3. Search bar - for both recipes and classes
4. Filter
5. Can book class and can favorite recipes / classes
6. Account settings - admin stuff (change password)
7. Profile settings - beautify your account (upload picture, bio etc)
8. View earning

With the data structure developed (as shown in the diagram below), we decided on the coding IDE environment and the programming language we were going to use to implement the rest of the project. In terms of IDE, we decided to use the Java Eclipse and Sublime as the IDE environment and Java as a main programming language. The webapp pages that we are doing include Login / Registration, User-Home, User-Profile, User-Account-Setting pages. We decided to distribute jobs for the above-mentioned pages in terms of coding and design team.

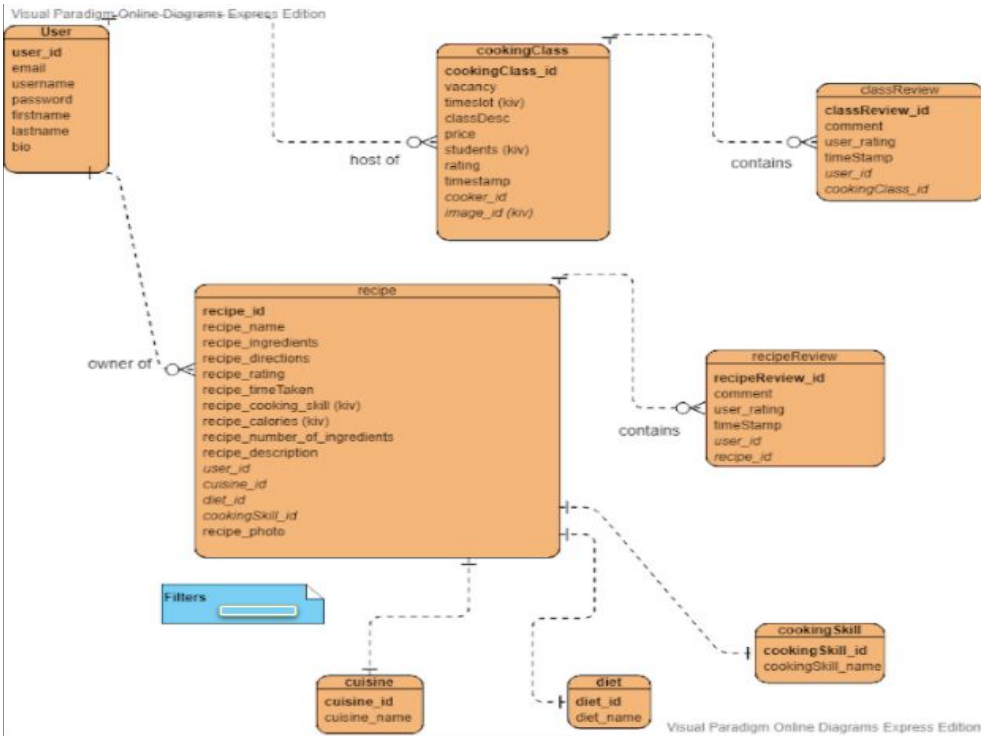


Figure 2: Database Structure Diagram

Below are the pictures for the various pages on our finalised webapp in terms of coding and the finalised design of the webpage integrated together.

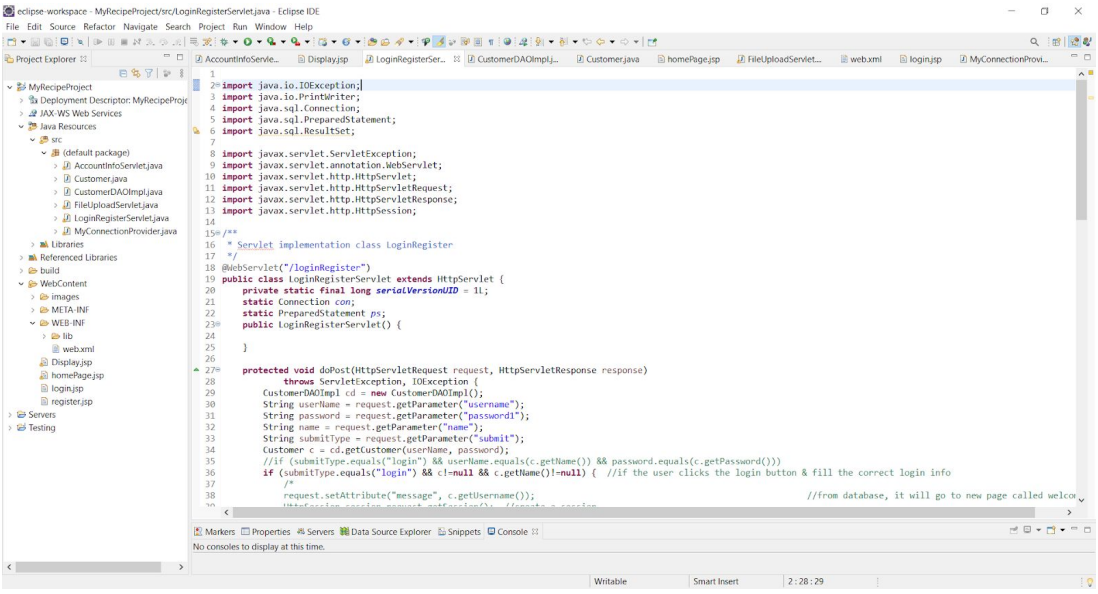


Figure 3: Example of coding for the ReSharePeace webapp

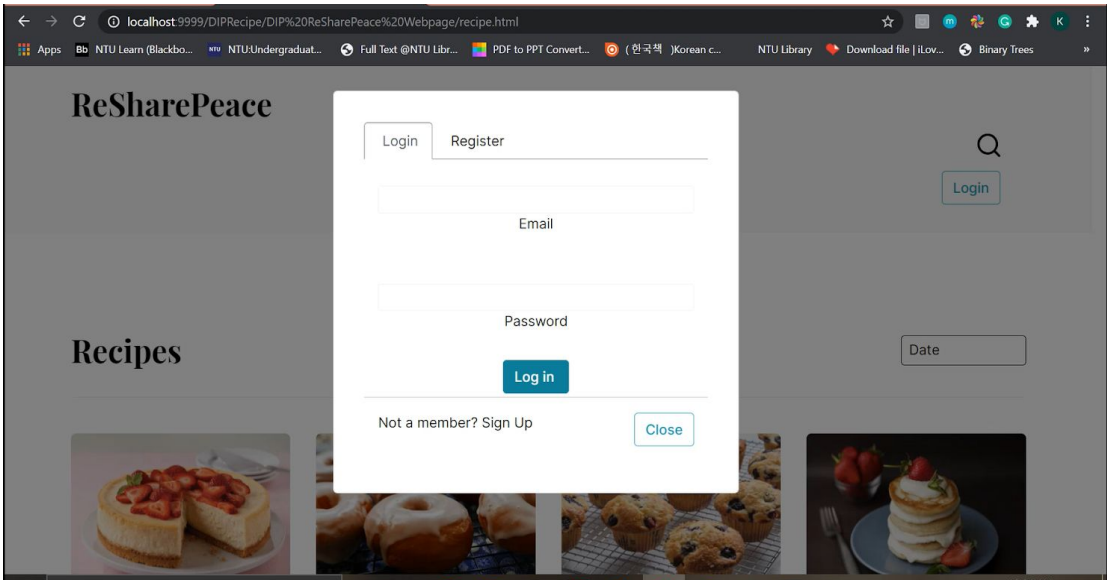


Figure 4: Login / Registration of ReSharePeace

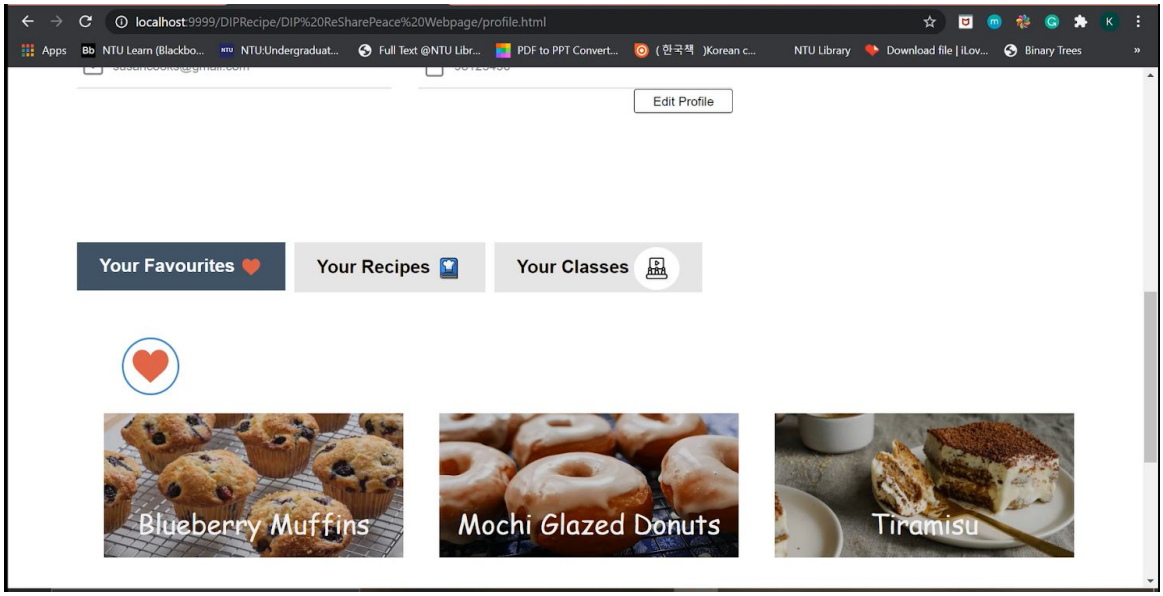


Figure 5: Profile Page of ReSharePeace

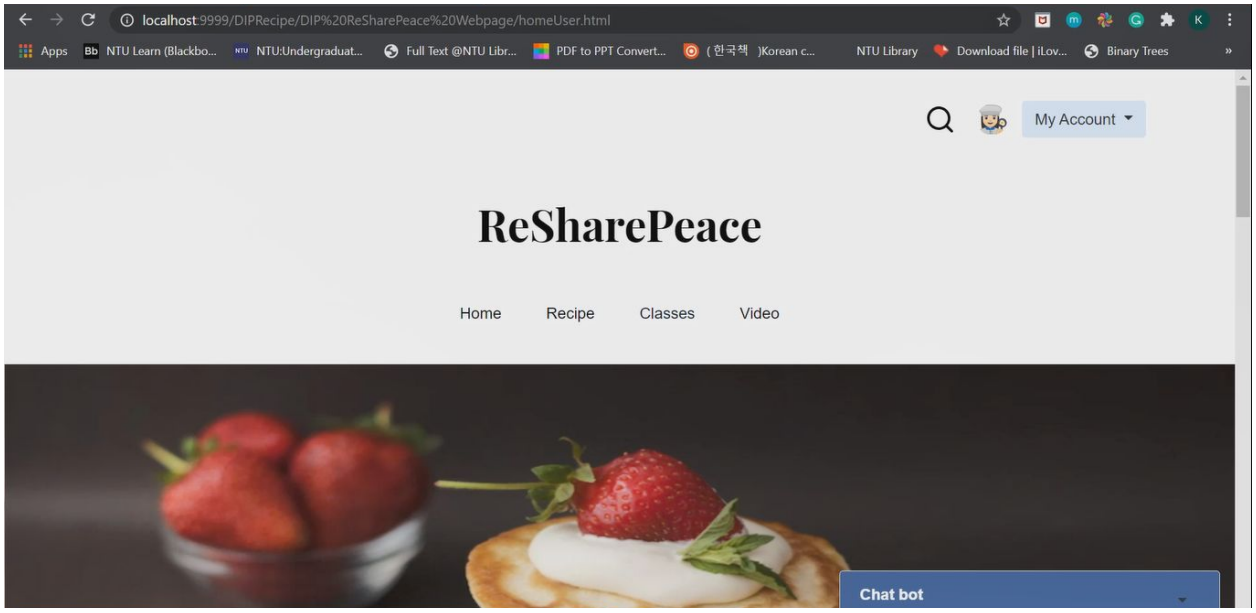


Figure 6: Home Page of ReSharePeace

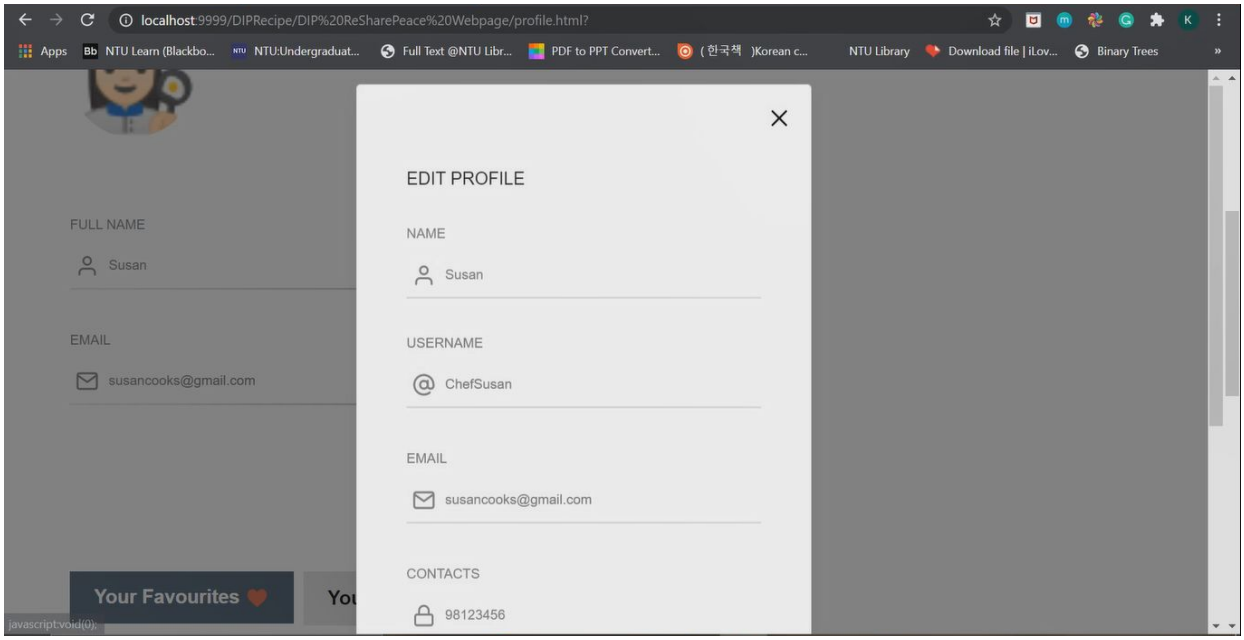


Figure 7: Account Setting Page of ReSharePeace

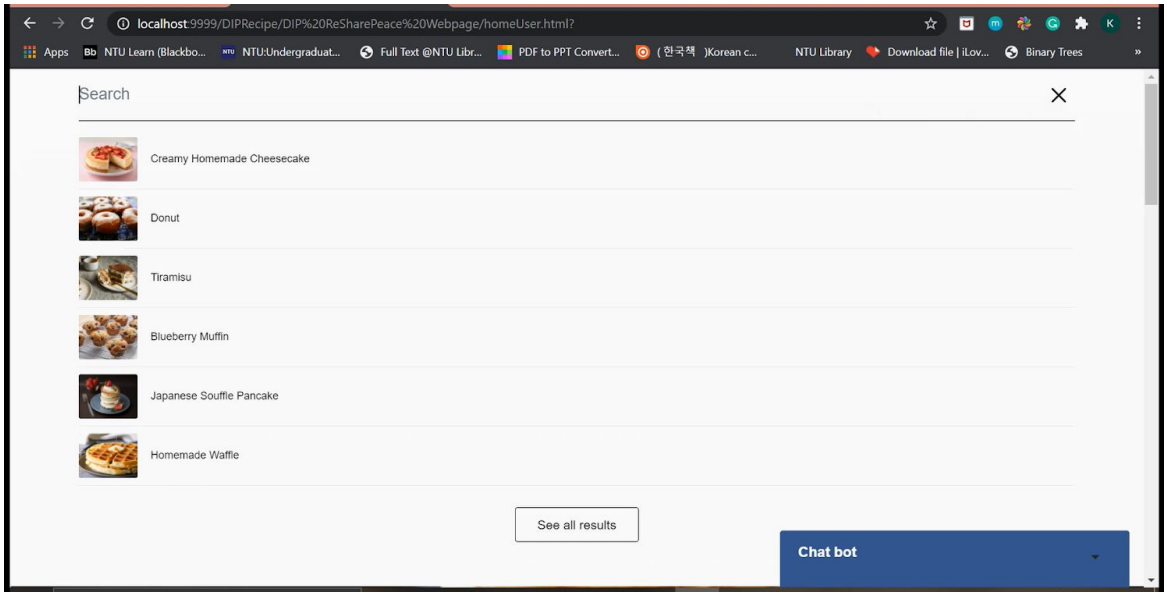


Figure 8: Search Query features of ReSharePeace

4.4 Unique feature-Using Web RTC to hold live classes

Current social networking platforms are not specialised for the use of a target interest group like cooking hobbyists. As such, those who have developed cooking as a novel hobby have to pick up cooking tips from recorded recipes on various platforms like Youtube or Instagram. However, these clips are often pre-recorded, which are non-interactive and are unable to guide amature cooks through the cooking steps in real-time.

Thus, we decided to fill the above void in the market by developing our own real-time video communication feature to host cooking classes between cooks and amateur cooking hobbyists.

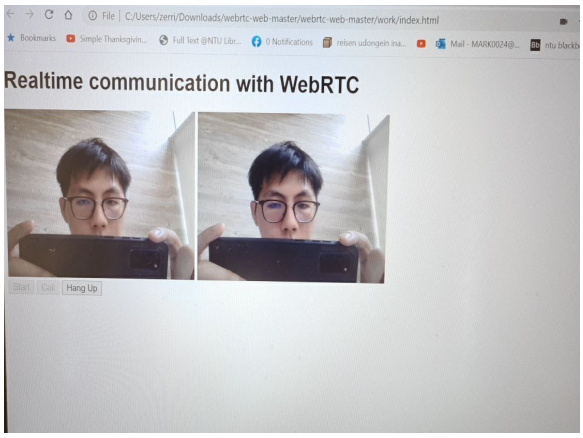
One reason for designing our own video application is because it gives us the flexibility to be able to customise our own features, such as a subscription feature into the video conference. This would not be possible had we rely on existing video conferencing applications. Moreover, existing video conferencing apps have not only made certain features (i.e. Increasing the maximum number of users allowed in a room) exclusive for paying users, but there have also been multiple reports against the security of their services making them unsuitable and unreliable for our platform.

4.5 Implementation of Web RTC video live class

For the live video application, we decided that the main coding language to be used will be Javascript and the IDE environment Visual Studio Code. To tackle this task, we separated the job into three categories (Coding, design and the integration for the video webapp with our ReSharePeace webapp). For the video coding scripts, below are the details as shown:

script.js	Js code to set up the video connection, size, functionality, etc.
server.js	Js code to set up mainly for peer to peer connection
room.ejs	Js code written in html for the user interface of the video application
style.css	For the styling of the video webapp

After we built the fundamental video web application, we decided to add more features such as (mute / unmute voice, on / off video, etc.) as well as improve the styling of the ReSharePeace video web application for the user-interface. Below are two pictures for comparison



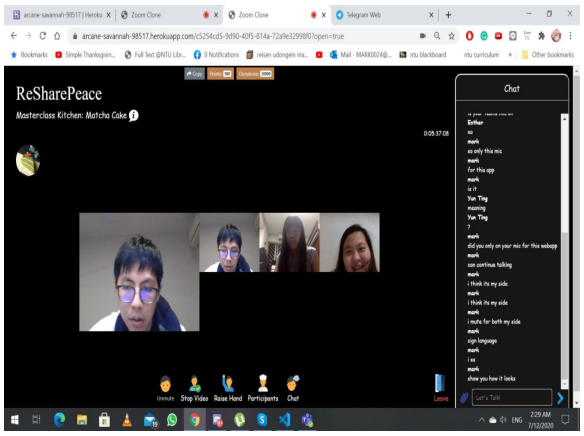


Figure 9: Before and After the ReSharePeace video design

4.6 Market Value

These days, there are multiple social platforms that quickly became viral after targeting interest groups like TikTok for sharing trending dance moves and Twitch for gamers to live stream their gameplay. As such, our application hopes to achieve similar success through targeting the cooking hobbyist interest group. Prior to gaining a high and sustained user traffic, we are making our web services free to attract users and cooks, who would also be incentivised to host class to earn a side revenue. After gaining the traction we need, we could then appeal to advertisers and earn a more sustainable income apart from the commission received from cooks, which are similar to business models of YouTube,Twitch and TikTok. These advertisers (likely culinary appliance companies, food delivering companies or restaurants) would likely be interested to advertise on our platform as they will be able to have confidence that their target market overlaps with ours.

5. Conclusion and Recommendation

5.1 Conclusion

We learnt a lot regarding teamwork, such as discussing ideas on how to make the webapp more interesting, adding more features and what we can continuously improve on for the web-application. Through sheer dedication, we stepped up from being mostly clueless when it comes to building an application, to acquiring a practical understanding of modern software development frameworks (e.g. JavaScript, MySQL, etc.) and various other developer tools (Heroku cloud, Toaster, etc.). This hands-on experience in building a software webapp application has had an incredible impact in developing our research, software engineering skills and teamwork. Having gone through various iterations and redesigns based on feedback from team members, as well as from the professors, we are now much more aware of the various considerations one should make when building software. These will then be able to help us smoothen the development process in the long run.

We strongly believe that this webapp fulfills the objectives we have set for our team. With most people not being able to travel outside due to COVID-19, we definitely feel that our product has hit new milestones.These include implementing cooking classes to allow users to occupy their time during the epidemic period, providing aspiring cooks opportunities to host lessons, as well as creating an online space which encourages recipe sharing and learning among users. However, there were initial concerns regarding development of many features which may not fulfil our expectations. For instance, our webapp is unable to support many users unlike zoom due to an issue with

bandwidth support. Nevertheless, we are proud of how our webapp and video applications are able to deliver a coherent and enjoyable user experience.

5.2 Recommendation for Future Works

Due to time constraints, there are a few features which we were not able to implement at this time. In future, developers could work on these ideas:

- Ability to upload video and make video content on the ReSharePeace webapp, as well as creating a system where users can earn money through the number of subscribers or views

- Creating a shopping cart to buy specific chef recipe books

- Integrating with multiple cloud-based platforms such as AMAZON, either through hybrid deployment or on multiple servers. This will allow ReSharePeace to be able to operate as a single, cohesive IT infrastructure for an enterprise

- Further development of design assets (home screen and background for video applications)

- Further coding to recognise ingredients and steps input by cooks. These will then be able to display relevant images of the ingredients, or gifs (i.e.short clips of how to separate egg yolk and egg whites) that are then automatically positioned between each step of a recipe as opposed to placing them manually

- Using AI to predict the type of cuisines / personality cooks users enjoy, and put them on personalised suggestions for the classes / videos they can attend / watch

Appendices:

User Guide,How-To Guides,Source Code are accessible on:

https://github.com/zerriet/Food-Sharing-WebApp-DIP