# Regina Food Bank Project

Team Mangos

DongYun Kim (200405213)

Nikita Khavronin (200404949)

Swarnim Amit Kumar (200433973)

Yi Xu (200457644)

ENSE271: People-Centred Design

Project Sponsor: Dr. Tim Maciag (ENSE 271 Lecturer)

April 8, 2022

**Business need and opportunity**

This is a twofold issue. The ongoing effects of the pandemic have created an increased need for support in our community, this is coupled with inflation that is particularly acute in its impact on non-discretionary items, food, rent, etc. have created demand growth that is not levelling off. While the food bank has the logistic capacity to meet the need on the supply end the surge in demand has in many cases overwhelmed the organization's capacity to book orders. High wait times and increasing call drops are presenting an increasing barrier to service.

The opportunity here is to design, prototype, and explore the development of a solution that allows clients to book hamper pick-ups 24/7. Thus, eliminating barriers to access and potentially freeing up time to grow the overall impact of the Food Bank on the lives of people facing food insecurity.

## 1.0 Reflections on project planning

### 1.1 North star & Carryover customers

Our north star customer is John Bailey, Chief Executive Officer (CEO) – Regina Food Bank (RFB) and we are proposing an Idea of Chatbot to reduce the load on the call center, and please the employee-customer interaction. When designing, we always think whether John will be satisfied with our product.

Our carryover customer is the RFB. We built this chatbot for RFB, and if RFB operates better because of the chatbot, the chatbot can be applied to other institutions as well, showing that this is a successful design to relieve the stress on employees and increase the capacity of the organization.

### 1.2 Assumptions and constraints

We are going to use NodeJS, MongoDB, and React for the realization of the project. NodeJS for chatbot systems and functions, MongoDB for storing user's sign-up pieces of information and React for user interfaces. Since some plugins on WordPress don't have an admin part or they need to be upgraded to a pro version, we needed to explore and test our chatbot on another platform.

### 1.3 Key findings from affinity diagramming and empathy mapping

From the empathy map and affinity diagramming, we found that with the high volume of calls, long customer waiting time and overwhelming workloads on staff, the RFB had difficulty operating normally. So we eventually got an idea about building a chatbot which can answer different questions and deal with orders. It may have a friendly interface and guidance to new

users. The administrators can also get feedback from users and collect some useful data like the preferences of various users.

## 1.4 Initial and evolution of our USM/MVP

We set our USM with the following ideas:

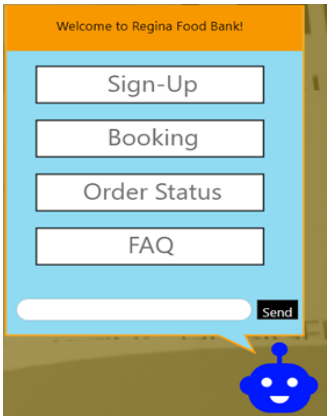
- Build a chatbot that can answer questions and serve orders. Users can also track the delivery status. These functions are basic but useful. If possible, bots are upgraded to answer more questions and give better answers with AI.
- People can select the hamper they want from the menu when ordering. Since customers have various needs, this function can be necessary.
- Collect and store data from users in the database so administrators can check and analyze them. It is indispensable for the RFB to make full use of the data to improve their services.
- Improve customer experience with a better interface and pop-up window notification for new users. The user experience must be ensured. Thus, they are more likely to use this chatbot next time, rather than still calling staff for help.

## 1.5 Summary of prototyping activities and findings

In Activity 2, we built lo-fi prototypes by Adobe XD or paper and pencil. In Activity 3, we built hi-fi prototypes by Adobe XD. Here are our findings:

- Exploring High-Functionality Applications. We concentrate our ideas and prototypes on a low threshold and high ceiling application. People now can ask questions, book a hamper and check delivery status, which can meet the needs of users. If time permitted, we can add more functions like remaining time estimation or driver location tracking.


Signifiers/Affordance (Default)




(After clicking on the icon)

- Affordance and Signifiers are important. There is a robot icon and a little chat box on the bottom-right corner. After clicking that, a big chat window appears and people can type in keywords or press buttons to get what they need. When people see this robot icon and chat box on the bottom-right corner, they may notice them and understand that they can interact with this to get a hamper or ask questions. The robot icon also indicates that users will interact with AI, instead of staff in RFB.
- Logical constraints are applied. We used to explore functions such as in-person messages or calling. However, these functions are almost useless since there are not enough people working in RFB so an automatic chatbot can better help them with the current situation.
- Keep it simple. With main functions appearing on the chat window, other functions can be hidden in a drop-up menu. If users have difficulty finding what they want, they may refuse to try the chatbot next time.

## 2.0 Reflections on project results

### 2.1 Overview of the project

Learning the concept of people-centred design to implementing those ideas into our project design was very satisfying and get to understand the design ideas quickly. When we first introduce chatbot design to our customers, we got to see the customer's reactions and received feedback for more improvement on our design system. From this project, our perspectives of the technology design system changed to a good designer/engineer who thinks of people-centred first. The only issue we had was the short time of this project. Since we introduced the chatbot design system, we had to explore the many functions and designs of that system but we only had a limited time to finish the activities.

### 2.2 Translating prototypes into API chatbot

Our team had successfully done working from the low-fidelity to high-fidelity prototyping and finally into JSON API chatbot. After a team meeting with Dr. Maciag, our group decided to explore our ideas outside of the WordPress environment and implement them using JSON API. WordPress only gave us the limited configurations or functions of our design system, so we decided to test them out on different a platform.

### 2.3 People-centred design ideas discussed in lectures

Every topic we learned in lectures supported our design explorations and even team development. When we first introduced the Regina Food Bank's problems, we immediately start thinking of design ideas and what to do for our next steps. But this was our mistake and we only had a big idea and missed all details. So we started to follow the activities details, going from the

empathy map, affinity diagram and User Story Mapping. Going through those processes helped us settle our design ideas and guide us to the next process. When we were thinking of a chatbot design system, we immediately thought it was good discoverability with a signifier and affordances. Lastly, going from the low-fidelity to high-fidelity prototype was the process where all of our people-centred design ideas were implemented and it produced what we exactly wanted to.

## 2.4 Plans for future projects

In future projects, we would always think of people-centred design rather than the technology (system) itself. We will always find the problem that our customer is providing and give them a solution that is useful, usable, and desirable so that satisfies our customers. Beginning of the course, Dr. Maciag told us that when designing a system, it is about crafting an understanding of people and processes. After finishing up the project with Regina Food Bank, we understood Dr. Maciag's statement on people-centred design and we will continue those skills and develop them in later future projects.

During the project, our group focused on high-functionality applications and especially on implementing our design ideas with low thresholds and high ceilings. We wanted our customers to get into our system easily and once they get in we wanted to provide many opportunities and possibilities. Also providing good discoverability like signifiers and affordances was one of our main design ideas for the project. In future work, we will keep exploring design ideas for high-functionality applications and good discoverability that the customers can easily find and easy to understand.

## 3.0 Configuration

Our group explored and tested the technical side of our idea outside of WordPress configuration and decided to implement them using Node.js (API), MongoDB (Database and storing data) and React (Graphical User Interface for chatbot itself).

Creating an API was very critical to our design system because we can adapt the API to work on WordPress or any other platform. It was one of our main design ideas to use an API because our customers can implement it on their platform application.

## 3.1 What is an API

An application programming interface (API) is a type of software interface that shares the applications' data and functionality with different computer programs. In simple words, we can call or bring API into different computers and uses all data and functions from the API. On the admin side using API will improve and increases productivity where API provides a powerful mechanism for accelerating development and on the user side, it will improve the customer experience and makes a new relationship between customers and the technology.

## 3.3 Demo

Examples of calling an API in the FAQ section:

1. Calling an API
2. Retrieve data from API (data.)
3. GUI (graphical user interface)

1

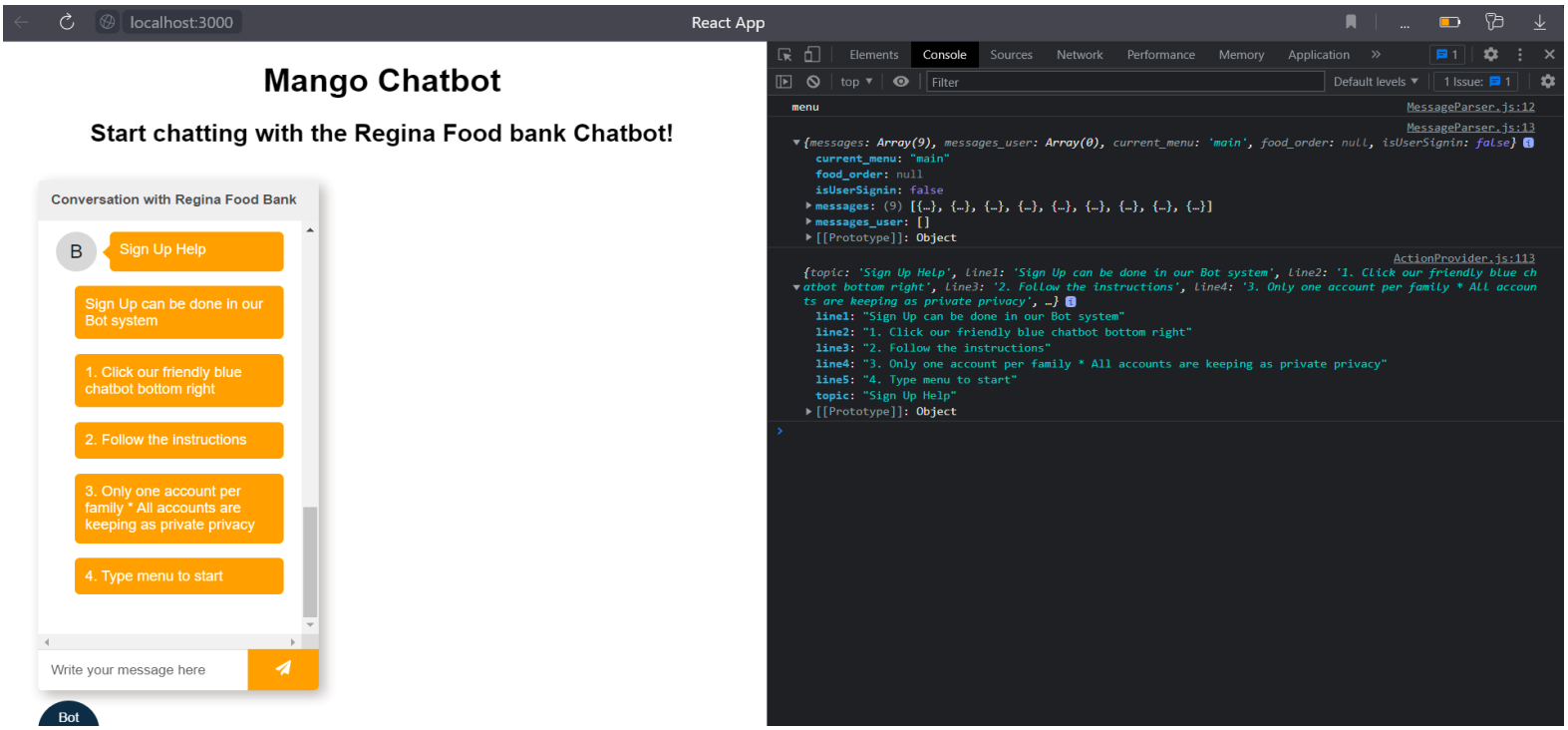
```
const response = await fetch("http://localhost:7777/nikita/faq", {method: 'GET'})
const data = await response.json()
```

2

```
const message = this.createChatBotMessage(data.topic);
const message1 = this.createChatBotMessage(data.line1)
const message2 = this.createChatBotMessage (data.line2)
const message3 = this.createChatBotMessage(data.line3)
const message4 = this.createChatBotMessage(data.line4)
const message5 = this.createChatBotMessage(data.line5)
```

3

```
this.set_chat_botstate(message)
this.set_chat_botstate(message1)
this.set_chat_botstate(message2)
this.set_chat_botstate(message3)
this.set_chat_botstate(message4)
this.set_chat_botstate(message5)
```




(Visualization of Chatbot and FAQ function)