**Satsang Adhikari**

101145635

COMP 3008

Assignment #4

# Part 1: System Proposal  (Taken from Assignment 3)

I was designated to work on AliExpress.com for Assignment 2. AliExpress.com is an e-commerce website which allows users to buy and sell products. The proposed software project falls under the same general domain as AliExpress, however it has a humanitarian aspect to it.

The project is titled *We Are One Big Family (WAOBF)*. It is a mobile application with two sets of users: one set of people in a neighbourhood who are in need and request for support (called Requestors); and another set of people who look to support those in need (called Supporters). The most important connection between AliExpress and *WAOBF* is that both platforms allow its users to purchase service as well as offer service. However, *WAOBF* adds a humanitarian aspect to the e-commerce domain. This project encourages people to help out in their community and look out for one another, hence why the project is named *We Are One Big Family*. The following scenario is to aid the reader's understanding of the project:

Joe lives in a neighbourhood within Barrhaven, Ottawa. One day, he unfortunately slips on his icy driveway and he is taken to the hospital. When he returns home after two weeks of recovery, he finds that he is having difficulties doing things he used to be able to do. He opens the WAOBF application on his smartphone and puts in a request to pick up groceries for him from the local market. Alex, a support person, sees Joe's request and chooses to accept it. Alex can view Joe's basic profile information including his home address so he knows where to deliver the groceries. At this point, Joe is notified that Alex is helping him and he is also able to see information about Alex from his end. Joe's credit card details are stored within the application which allows for swift payment to Alex once Joe's request is completed. Then, Joe can give a rating to Alex and likewise Alex can give a rating to Joe. Providing high quality service is very important for support persons such as Alex as it leads to gaining favourable ratings and receiving recognition in the community. Being easy to work with is very important for requestors like Joe so they continue to receive support from the community.

# Part 2: Metaphor and Interaction Types

## User Interaction Metaphor

### (i) Metaphor

A community bulletin board is the metaphor of choice for the system. The purpose of a bulletin board is to convey various kinds of information to large groups of people in the community.


Source: https://shutr.bz/3IhUj9Q

### (ii) Advantages

A clear advantage of this metaphor is its ability to showcase information in a centralized location, This is much similar to how the mobile application works; with the users being able to open the "one-stop shop" app on their mobile devices. A bulletin board offers an effective way for communication between many people and also promotes community awareness. Additionally, it encourages people in the community to participate in offering and requesting services. Furthermore, this metaphor also captures the idea of priority and precedence. A bulletin board contains posters of various sizes and usually the bigger posters indicate higher priority. Similarly, our mobile application categorizes requests as low priority, medium priority, or high priority.

### (iii) Disadvantages

A major disadvantage with this metaphor is its inability to portray conversations. Let us create a situation to address this issue: Person A puts up a poster on the bulletin board and Person B sees the poster and seems interested. However, Person B cannot find any contact information on the poster to learn more about what Person A is offering. This creates an information barrier. In our mobile application however, users have the ability to view profiles of one another and communicate effortlessly. Another major disadvantage with the metaphor is its failure to integrate rating functionality. This functionality is critical in the application as it provides feedback to both support persons and the requesters for ensuring quality service.

### (iv) Is the Metaphor Sufficiently Promising?

The bulletin board metaphor is promising enough to worth making a prototype. It suggests to the users that they can imagine the system as a collaborative tool which encapsulates the goal of the project: to foster a community whose members work together to help each other. Furthermore, some of the disadvantages highlighted in the above section should not create any difficulties for the users in terms of understanding the system. It is common knowledge that communication is vital for such a platform. Users would expect to be able to communicate with one another. Also, the rating functionality might not be something that users would immediately start seeking when imagining the platform. In this sense, the pros outweigh the cons. Thus, this would be a worthwhile metaphor to pursue.

## Interaction Types

### (i) Conversing

In the conversing style, users have natural conversations with the system. The main benefit of conversing style is that it gives the system a more life-like shape capable of communicating as if it is a human. This can make users, especially non-technical users and users with certain disabilities, feel comfortable about navigating the system. On the other hand, the major issue with this style, as mentioned in the lectures, is that the system may misunderstand what the user says and produce unwanted results. A common situation in our mobile application could be that a user wants to request for help using their voice but the system interprets it incorrectly, possibly due to their accent, leaving the user frustrated. As the issue outweighs the benefit, conversing style may not be the best fit for the application.

### (ii) Manipulating

In the manipulating style, users manipulate virtual objects using the knowledge of how those objects work in the physical world. These virtual objects can be dragged and dropped, opened, closed, and zoomed into. The main benefit of a system that uses this style is that it is very easy to grasp even for new users as the state and behaviour of digital objects will fit their mental model of how the physical version of those objects work. However, the major drawback of the style is that not every system extensively involves actions where users have to manipulate digital objects. Furthermore, users with certain disabilities may have a hard time on a system that relies heavily on physical movements. Our mobile application mostly does not require users to manipulate objects, therefore this style may also not be a good fit.

### (iii) Instructing

In the instructing style, users tell the system what to do. It makes use of menus, touch screens, voice commands, and button-presses which are all actions commonly associated with mobile applications. Furthermore, as the lectures mention, this style is suitable for when users need to perform repetitive actions. Most applications offer shortcuts that make it easier for users to access certain parts of the app they use frequently. Our mobile application could make use of these traits. While the instructing style may not be as easy to grasp initially as manipulating style or while it may not give a life-like feel to the system as conversing style, it seems to best allow users to give instructions through the mobile application. The benefits outweigh the issues, therefore instructing style is the best choice for the system.

# Part 3: Low-fidelity Prototypes

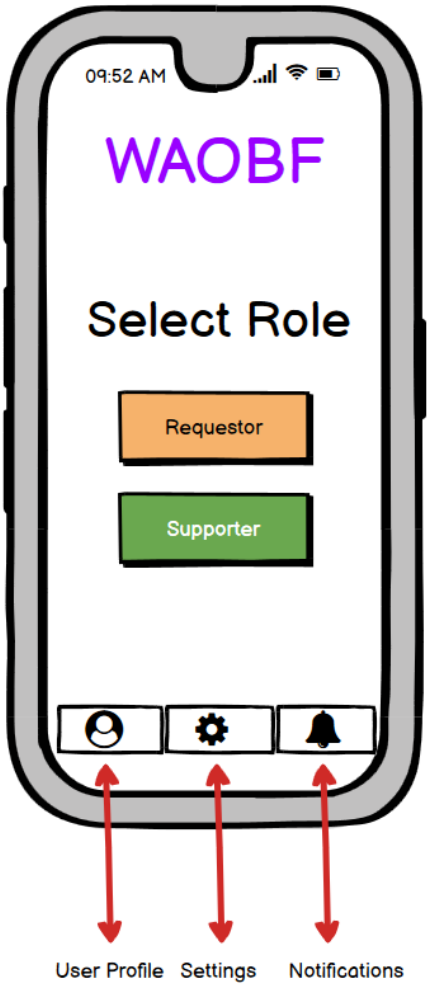
The prototype screens presented below cannot be interacted with due to formatting issues when bringing them into this PDF. I have provided the Balsamiq file in my submission containing the prototypes which have clickable links that you can download and interact with.

## Prototype 1

When user opens the app, they are asked to select their role. Each button is a link to a different screen in the prototype. There are menu options at the bottom of the screen for this design.

Principle of Layout: Contrast

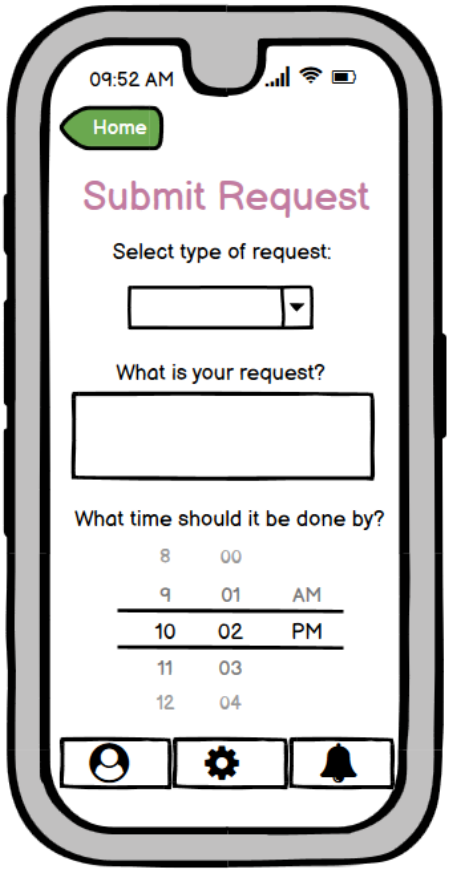
Contrast between the two buttons is captured through the use of varying colours such as orange and green. They each serve a different purpose in the application therefore they must be visually distinguishable.




UI Pattern: Extras On Demand

Like in most mobile applications,
the layout of an app typically consists of showing the current important content up front and the rest of the content, which are not immediately needed for the users, are tucked away in categorical boxes towards the bottom of the screen.

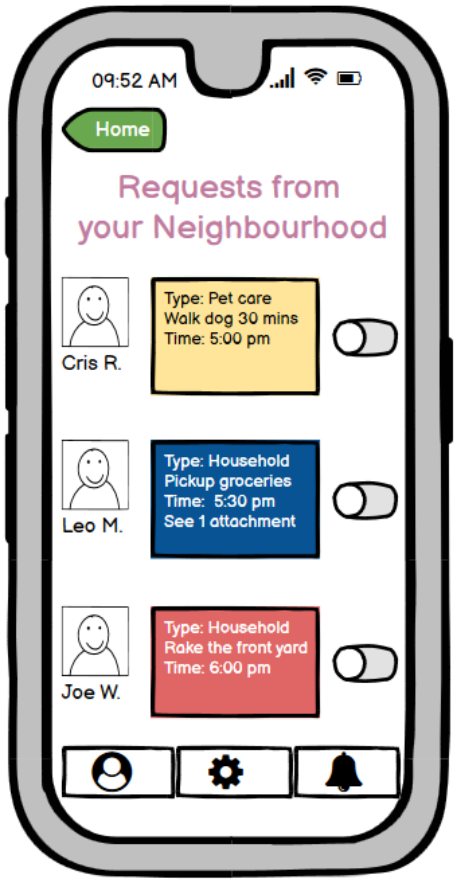
If user selects Requestor role in the Home screen, they are asked to fill out a short form. Firstly, it will ask them to select the type of request they are making. This could be, for example, household-related, car-related, pet-related, etc. Then they must provide a short description of the task for the support persons as well as the time the task must be done by. There is also an area where user can post attachments (which is currently hidden in this screen).




Principle of Layout: Repetition

The menu options at the bottom of the screen stay consistently no matter which page the user is currently at. This repetition in our design brings cohesiveness and the user is able to access exactly what they need without wandering around in confusion.

If user selects Supporter role in the Home screen, they see a list of requests made by requestors in their neighbourhood. Each list element contains the requestor's profile picture, their name, the details of their request. and also any attachments the requestor may have included. Currently, all the requests are toggled OFF. If the support person wants to designate a particular request to themselves, they would toggle ON the button beside that request.

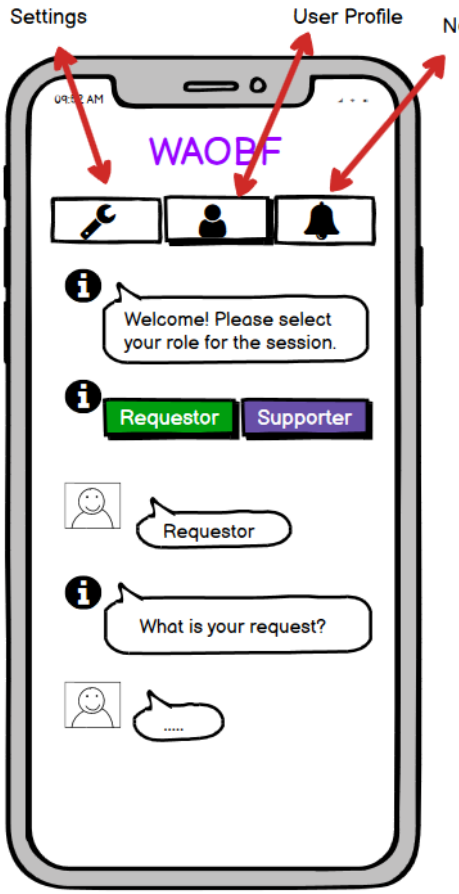



UI Pattern: Center Stage

Everything important for the user to see and interact with is displayed in the middle of the screen. Other aspects of the application that are not currently relevant are cluttered towards the bottom of the screen.

# Prototype 2

While the first prototype used Instruction interaction type, this prototype uses two different interaction types. For the home screen in this prototype, Conversing interaction type is followed. When user opens the app, they are greeted by a chat bot asking user which role they want to have for the session, followed by the two buttons. If user selects Requestor role, the bot asks user what their request is and a few other questions. After they have answered all the questions, the request is automatically created and posted for the user by the bot. If user selects Supporter role, they are taken to a different screen through a link. There is also a link to the User Profile from the menu section at the top.




Principle of Layout: Contrast

Contrast between the two buttons is captured through the use of varying colours such as green and purple. They each serve a different purpose in the application therefore they must be visually distinguishable.

UI Pattern: Responsive Disclosure

The chat bot guides the user in a step-by-step fashion. The UI is minimal when the user opens the app and gradually expands as they interact with the application.

If user selects Supporter in the Home screen, they are presented with this screen which uses Manipulating interaction type. They can see a map of their neighbourhood which can be zoomed in/zoomed out, scrolled around, etc. All of the requests from different requestors around the user's neighbourhood are displayed by little colorful popup icons. If user clicks on an icon, they can see the requestor name and their requested task. When user picks a request icon, a button is displayed at the bottom for continuing with that request.

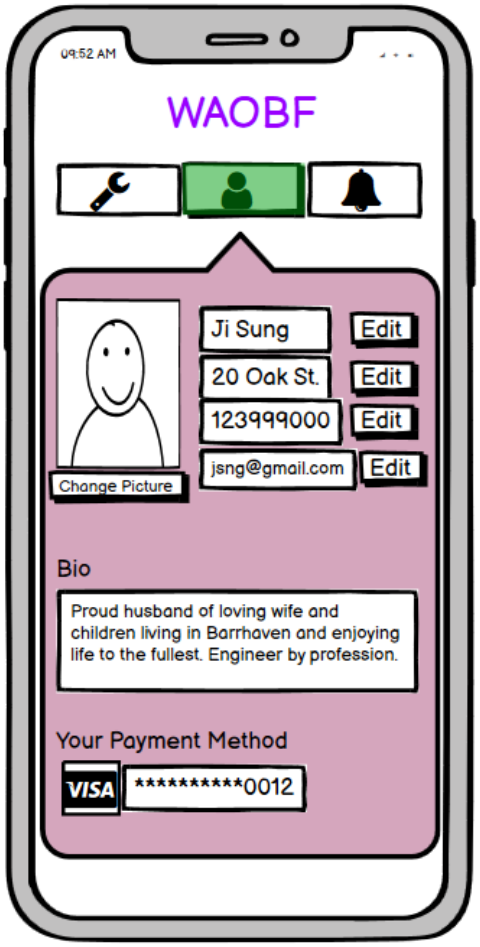

Principle of Layout: Repetition

The menu options at the top of the screen stay consistently no matter which page the user is currently at. This repetition in our design brings cohesiveness and the user is able to access exactly what they need without wandering around in confusion.

UI Pattern: Center Stage

Everything important for the user to see and interact with is displayed in the middle of the screen. Other aspects of the application that are not currently relevant are cluttered towards the top of the screen.

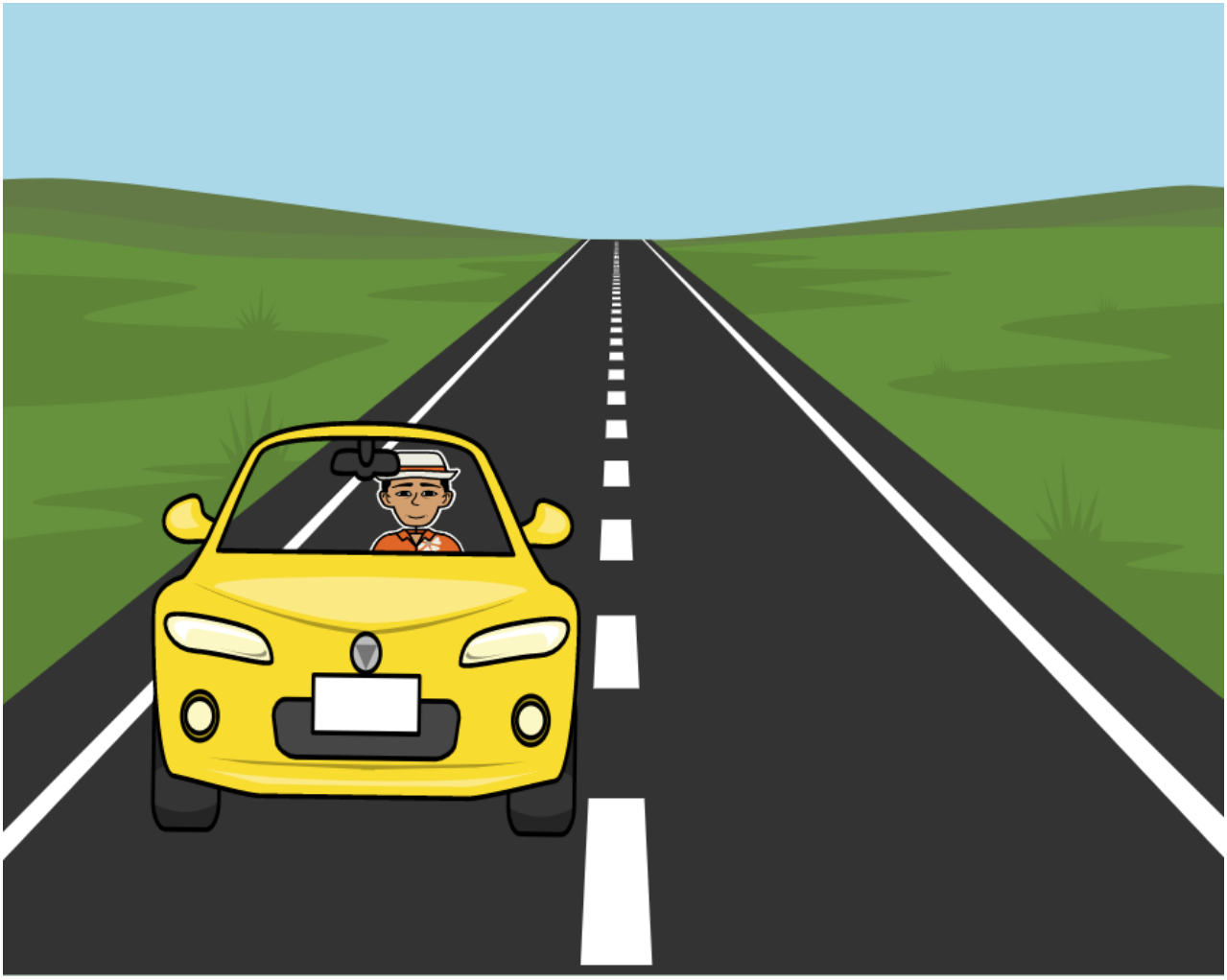
This is the User Profile screen where user can add/modify their personal information such as profile picture, name, address, phone, email, bio, as well as preferred payment method. If we recall the scenario, when a support person successfully completes a request, the requestor must pay them through the app. For this, it is required that a payment method is saved to the app. For security reasons, when user opens the app, they would be required to scan their fingerprint. (This feature is not explored in the prototypes)




After looking at the two prototypes, we see some changes were made to the preliminary planning. I had initially suggested that Instruction interaction type seems most fitting for the application. However, after building the first prototype it inspired me to come up with another prototype that is not only distinguishable in terms of design but also with interaction types. I ended up making use of all three interaction types. Also, since a mobile screen does not have the space luxury to showcase everything, some of the features that were mentioned prior, like the rating functionality, could not be integrated into the prototypes. Task priority, which was mentioned in the metaphor section, was also not integrated.

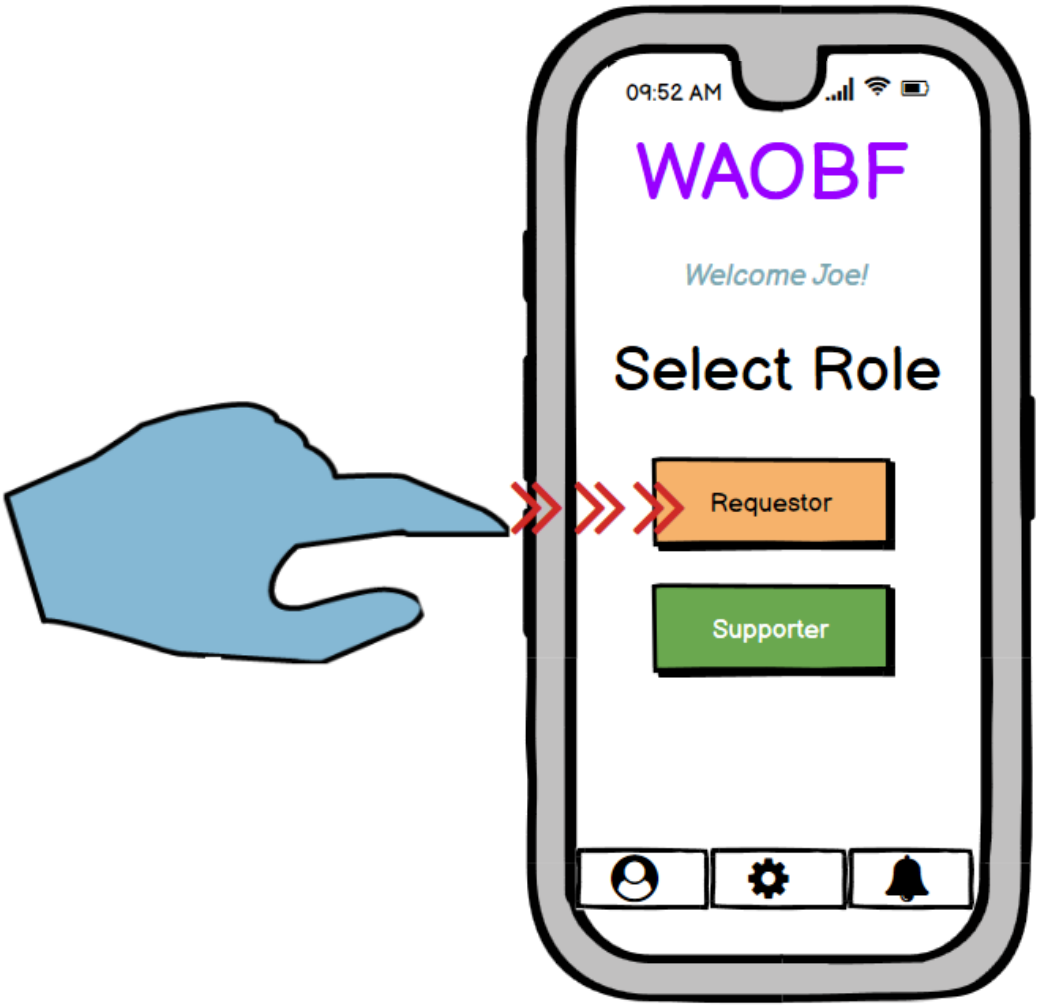
# Storyboard

**Step 1:**



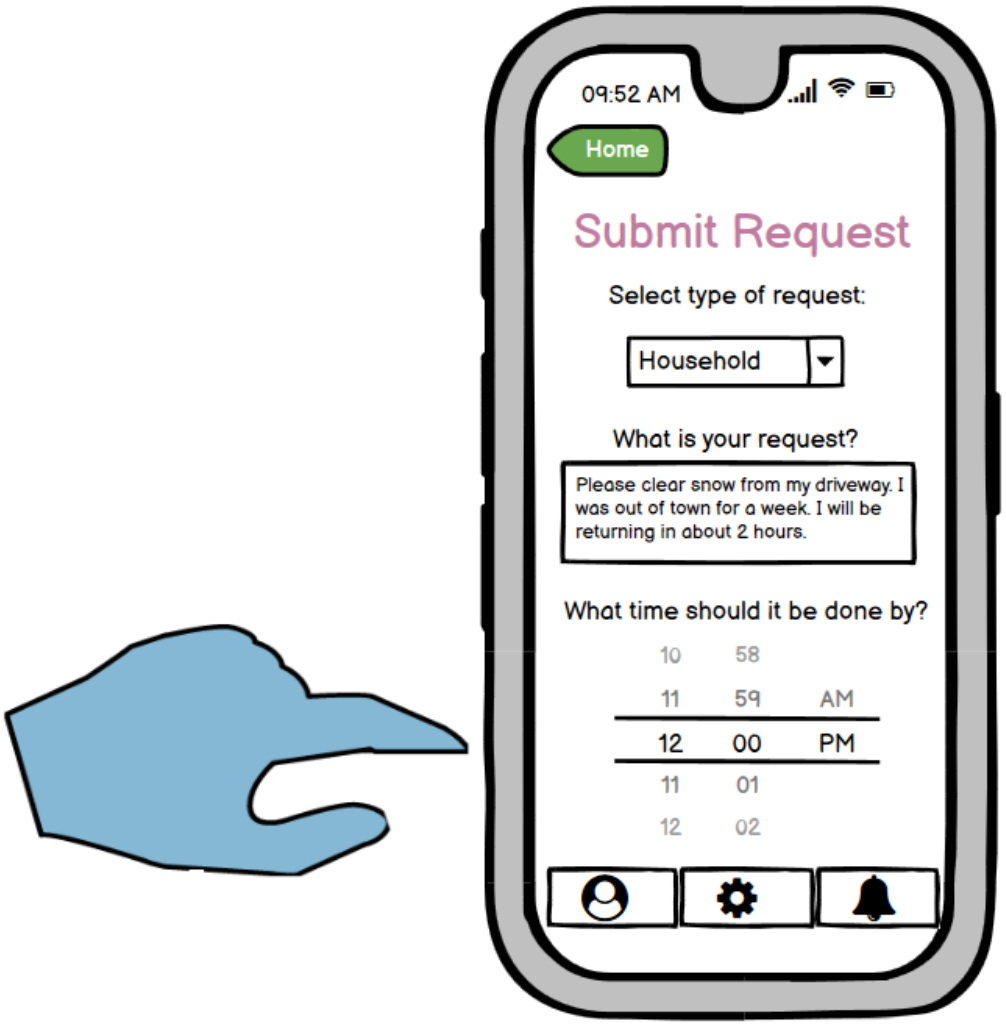
Joe travelled south on a road-trip vacation for a week to escape the harsh winter in Ottawa. When he is returning home, he hears in the news that Ottawa has been seeing heavy snow for the past few days. Joe realizes that he would not be able to park his car onto his driveway as there would likely be a mountain of snow blocking him. He would need to ask somebody for help before he arrives home.

**Step 2:**




Joe opens the *WAOBF* app. Since he needs support, he chooses the Requestor role. His GPS says that he will be home in 2 hours 15 minutes. Joe figures that it should be enough time for a support person to see his request and complete it before he arrives.

**Step 3:**




Joe fills out the *Submit Request* form. The type of request is household-related. He makes sure to put down exactly which task he is requesting for. He also puts in extra details such as being “out of town for a week” and “returning in about 2 hours” which lets the support person know how much work they should expect and how much time they have for it. The current time is 9:52am for Joe and he wants the task completed by 12:00pm.

**Step 4:**


Alex, who is using the *WAOBF* app with the Supporter role, sees Joe's request among the list of requests from his neighbourhood at 10:05am while he is taking a walk. Alex toggles ON the button beside Joe's request indicating that he will be working on it. Alex can open Joe's profile to see his home address so he knows where he must go.

**Step 5:**

Alex finishes clearing Joe's driveway just as Joe arrives. Alex must have been working on it for almost 2 hours. Joe comes home to a clear driveway and also sees mountains of snow on both sides of his driveway which indicated how much effort Alex put into the task. Joe is very satisfied with Alex's hard work and he gives Alex a payment of $50 through the app as well as a 5-star review. Alex also gives a 5-star review to Joe for being understanding and compensating him well for his work.

# Appendices

This is my rough sketch on MS Paint to plan my ideas for the prototypes before I started implementations on Balsamiq.