NEW YORK INSTITUTE OF TECHNOLOGY

New York Institute of Technology

Department of Computer Science

CSCI 457 - Senior Project

Aware City

Project Technical Report

Spring 2023

Date: May 10 , 2023

Advisor: Prof. Wenjia Li

1271003: **Brenda Encarnacion** (bencar01@nyit.edu)

1264863: **Shajid Muntaser (**smuntase@nyit.edu)

1272119: **Fredy Sorto** (fsorto@nyit.edu)

Table of Contents

Abstract:

Our React Native and Expo-based mobile application offers a user-friendly interface to help unhoused individuals find nearby resources such as shelters, food pantries, social services, and emergency services. The app is designed to be cross-platform, accessible to anyone with a smartphone, and built with npm and Node.js. Additionally, the app encourages people to find non-profit organizations they can volunteer within the community or donate to. Our goal is to provide a comprehensive and convenient destination to assist those in need.

Keywords: React Native, Expo, npm, Node JS

1.0 Introduction

Our team is composed of three members who collaborated to develop our senior project. We chose to create a cross-platform mobile application using React Native and Expo to help unhoused individuals to gain access to essential resources. Our primary goal was to provide a simple, user-friendly app that enables users to locate nearby shelters, food pantries, soup kitchens, and more. Unlike many similar projects, we focused on developing a cross-platform app accessible to anyone with a smartphone or tablet, making it an essential tool for unhoused individuals who may not have access to conventional resources. Rather than building an Android or iOS-only app, we opted to use React Native for its cross-platform functionality. Our target audience is unhoused individuals as well as anyone who needs assistance around the city. Our app provides direct hotlines for victims of crime, sexual assault, and for anyone facing substance abuse problems. Overall, our mobile app is centered on helping individuals in need by providing a simple, user-friendly application that can assist them in finding essential resources and services.

1.1 Existing System

Several mobile applications have emerged as an effective means of addressing social issues, notably homelessness. These applications have provided essential resources and services to homeless individuals, underscoring their potential as powerful tools in tackling this complex societal challenge.

One of the main mobile applications that displays this, is called “Homeless Resources - Shelter App”. Homeless Resources is a non-profit organization run entirely by volunteers. Their mission is to provide vital assistance to homeless and low-income individuals, with a particular focus on helping young people connect to the resources and services they need to improve their

lives. Their services are concentrated in several cities on the West Coast of the United States, including Portland, Oakland, Seattle, El Paso County, Santa Clara County, and San Diego County. Through their mobile application, users can easily find and filter services based on their needs, such as food, shelter, health resources, and employment opportunities. The app provides detailed information on each service, including contact information, directions, and scheduling. Users can also access a crisis line for emotional support and connect with registered providers. What sets Homeless Resources apart is its commitment to empowering both users and service providers. The app enables individuals to list or manage a service, with updates on the availability of resources like shelter beds. Service providers can also request feedback from users, ensuring that they are meeting the needs of their clients.

Another application that aims to help homeless people is “OurCalling”. The purpose of OurCalling is to assist individuals who are experiencing homelessness in Dallas Texas by providing them with access to essential resources and services such as shelters, food, healthcare, and job opportunities. This application also offers a community section where users can connect with other communities and receive emotional support from trained volunteers. OurCalling intends to develop, heal and strengthen the relationships of those in need to lead them to a healthy and sustainable lifestyle. What makes OurCalling unique is that it allows outside users the ability to report a homeless camp location by taking a picture and OurCalling’s search and rescue team will visit them and provide them with different resources.

Lastly, "WIN: What I Need" is an application that provides resources for individuals struggling with resource insecurity, particularly homeless youth, former foster youth, struggling families, struggling veterans and college students, seniors, and individuals fleeing abuse. The app empowers these populations to connect with over 2,200 free programs offering food, shelter,

healthcare, legal aid, jobs, transportation, education, government benefits, crisis support, hotlines, hygiene facilities, drop-in access centers, and overnight parking, with contact information, addresses, and websites. What sets WIN apart from other applications is its user alert notification feature, which sends users important community announcements. Additionally, it offers an "I Need Help Now" one-tap function, allowing users to connect to emergency services quickly. Finally, the app fosters a sense of community by allowing users to connect and share their experiences.

After doing numerous amounts of research, it is evident that there is a shortage of mobile applications designed to provide resources for homeless individuals. This presents a significant gap in the support infrastructure for homeless individuals in the city and highlights the need for further development of mobile applications tailored to the unique needs of this population.

1.2 Proposed System

Our proposed system is a React Native cross-platform mobile application that simplifies access to essential services and resources for individuals experiencing housing insecurity. The application includes three different tabs, each with its purpose. The home screen features emergency and direct service hotlines, making it a valuable resource for those in need while also providing information about the application. The map screen displays the user's current location and locates the nearest public services, with color-coded markers indicating different types of services, such as blue markers for homeless shelters. By clicking on the markers, users can receive directions to those services. Finally, the charity screen provides a list of non-profit organizations that people can volunteer for or donate to in the NYC community. Our main goal is to develop a user-friendly mobile application that empowers individuals to easily access essential resources and services.

1.2.1 Advantages of the proposed system

The proposed React Native-built cross-platform mobile app we developed known as "Aware City" has several advantages over current options. First of all, it makes it simpler for those who are houseless to access crucial services and resources, making it simpler for them to get the assistance they require. A useful element that can save lives in terrible circumstances is the homepages' inclusion of easily reachable emergency and direct service hotlines. The map screen also makes it easier for users to explore the city and find the resources they require by showing their present location and the location of the nearest public service. Thirdly, the charity screen provides users with an easy and convenient way to find nonprofit organizations and donate to issues that are important to them.

1.3 Software Engineering Model

The software engineering model we chose for this project was the Waterfall Model. We carefully considered the project requirements and timeline and concluded that the Waterfall Model would be the most efficient approach. Our team spent time planning and discussing the solution to the project requirements. We were able to identify the requirements, design a solution, develop a working project, test it for deployment, and implement any necessary changes. Additionally, we conducted daily scrums to ensure everyone was on the same page and to discuss updates, issues, or any roadblocks. Although the Waterfall Model has a sequence of operations, we were able to tailor it to our needs, and it provided us with a structured approach to completing tasks efficiently. Ultimately, our use of the Waterfall Model and daily scrums enabled us to complete the project on time and with success.

1.4 Purpose

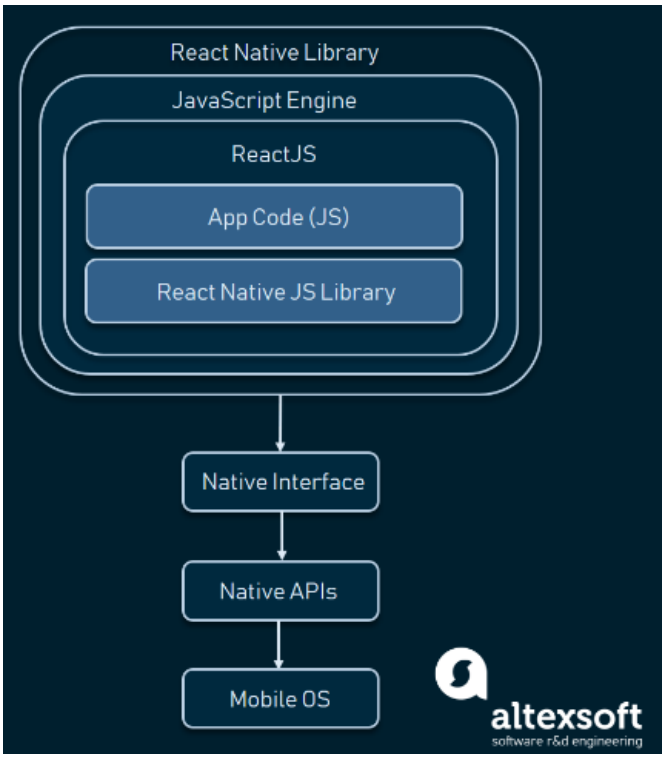
The team decided to build this application to build a one-stop shop for individuals who are in need in NYC. We noticed that there is a lack of determination from big cities like NYC and LA to help people in need, especially the unhoused members of our society. Everyone faces hard times in life, but that does not mean that they should suffer for the remaining time they have left. In an article written by Chau Lam for the Gothamist, “In 2022, the city estimated that 3,439 homeless people were residing in public spaces. As of Jan. 8 [2023], there were 67,880 homeless children and adults living in shelters.” We aim to help them find resources with more ease through the use of our cross-platform application. Ideally, we would love to share our progress with NYC initiatives like “LinkNYC” to implement our application onto their proprietary devices that are already set up all around the city. These devices are already in place with cell services and WiFi capabilities. This would allow for the quickest implementation in NYC and put the application to real use.

1.5 Project Objective/ Goals and Scope

The goal of this project is to positively impact the city we live in. Especially in a metro city like ours, there is a large population of individuals and families that suffer from a lack of housing, supply of food, clothing, and more. We strive to be the number one aggregator of such resources and change as many lives as possible by doing that. Being a cross-platform mobile application, we have created a wide range of possible users of the app, which allows for many more to have access to such a resource. Through future partnerships, our goal is to integrate the application onto public devices that can be accessed by anyone who needs assistance. We hope to be the one-stop shop for such individuals and make an impact where it matters the most.

1.6 Technologies and Tools

As mentioned previously, we opted to build a cross-platform application to provide ease of use to any users of the application. As for the front-end framework of the application, we have implemented React Native to build it. React Native is an open-source framework developed by



Meta Platforms, Inc, which primarily uses React and JavaScript. We opted for this option because this allows us to create an Android and iOS application using a single codebase. Within React Native, we chose to build with the tools and services provided by Expo. Expo allows developers to build and deploy apps much more quickly, rather than starting from scratch. Their pre-built components allowed for ease of creating and testing the application. Examples of such components include but are not limited to; location, status-bar, and linking. Using Expo also allowed the team to test the application on their mobile device with the ease of QR codes as well as using Android/iOS emulators on our computers.

Regarding the emulators, for Android testing, we used Android Studio to simulate an Android device and Xcode on our MacBooks to simulate an iOS device. This is set up on our computers and by using Expo, we were able to easily import our codebase onto a functional device. Features like these allow for ease of build without needing complicated setups and without requiring writing any native code for the front end.

As for the backend of the application, we have used Node.js which works in tandem with frontend frameworks like React or React Native. Node.js uses JavaScript to execute server-side code. This gave us the advantage of using a single programming language like JavaScript for both the server-side and client-side development of the application. Such an architecture allowed us to streamline the development process and allow for ease of collaboration as well. We used Node Packet Manager (NPM) to manage and share third-party libraries, tools, and packages. NPM is installed within Node.js by default, providing access to several open-source packages to build certain applications. NPM makes it very easy to install, manage and update packages for Node.js and React Native as well. By the use of the command line interface (CLI) in VSCode, we were able to execute such actions. We were able to add functionalities to our React Native application by using NPM.

An example of such a functionality is the Google Maps API that was implemented in our application. We have a built-in “Map Screen” in our application that allows users to check their current location and find the closest services like homeless shelters, social services, and more. We have used several math functions that use latitude and longitude, written in JavaScript, to show users the top 10 closest resources to their current location as well. This allows for a more focused number of resources provided to the user, rather than overwhelming anyone with the vast amount of resources available on the application.

1.7 Users

The users of our cross-platform mobile application who are houseless individuals or in need of resources and services are the target audience. Our application's major goal is to give them quick and easy access to necessities like shelter, food, medical care, and other public

services. We anticipate that by using this application, houseless persons will be able to quickly and easily access the assistance that they need.

We want to serve those who are houseless as well as the case managers and social service providers that work with this demographic. These experts may help their clients find the resources and services they need by using the application's user-friendly design.

In the end, our application can help contributors and volunteers who want to help those who are houseless. Our application can help people become active in their local community and make a difference by offering a simple way to donate to nonprofits that assist this group.

1.8 Motivation

We aim for this mobile application to make a positive impact on those individuals who need help and people who need service. The motivation behind developing this mobile application is because we are New York City residents and we can see the growth of houseless individuals in the streets. As students from NYIT, we have hands-on experiences of encountering houseless individuals commuting to and from campus. So we were aware that it can be difficult for these people to get the resources and services they need to get themselves in a better state. The purpose of this mobile application is to make it simpler for people who are homeless and others who are having trouble getting help. We believe that our mobile application can effectively be the first step in making sure these individuals get the help they need.

1.9 Timeline

- February 8, 2023 - February 22, 2023:
    - We have defined the analysis for our project.
    - Designed our product architecture and completed our UX/UI design.
- February 22, 2023 - March 9, 2023:

    - Created a carousel to display the name and a short description of the organizations that offer volunteering opportunities so outside users can know where to find places to help the community.
    - Managed to link the carousel with the markers on the MapScreen.js screen file. So as the user is swiping through the carousel it will animate to that marker and center it on the screen.
    - Added a Loading.js file that is just basically a component for whenever we need things to load such as the MapScreen when asking users for location permissions.
    - Started color coordinating the markers such as
        - Homeless shelters = Blue (represent safety and security)
        - Food pantry = Green (represents food supplies given)
        - Soup Kitchen = Yellow
        - Social Services = Purple
        - Donation Services = Orange
        - Emergency Services = Red
- March 9, 2023 - March 30, 2023:
    - Improved the charity page by removing the carousel and making the volunteering organizations in a flat list with text and images.
    - Successfully adding the map to the application.
    - Proper navigation through the application.
    - Hotline numbers now have a link to lead to the phone app.
- March 30, 2023 - April 14, 2023:
    - Add more locations on the map.

    - Code the map filters into mapscreen.js.
    - Sorting markers by location.
    - Fixing the overall design of the application for font and colors.
- April 14, 2023 - April 28, 2023:
    - The “visit” button works on the charity page, so now when clicked it takes the user to each of the organization's websites.
    - Created a website to display our overall application progress and individual contributions.
- April 28, 2023 - Present
    - Cleaned up the user interface making it much easier to navigate through.
    - Added more information to the carousel card such as title, description, address, hours, and phone number.
    - Made the carousel much more interactive for the users to easily access the information of their nearest service.
    - Sorted the data coordinates based on the user's location.

2.0 Analysis of the System

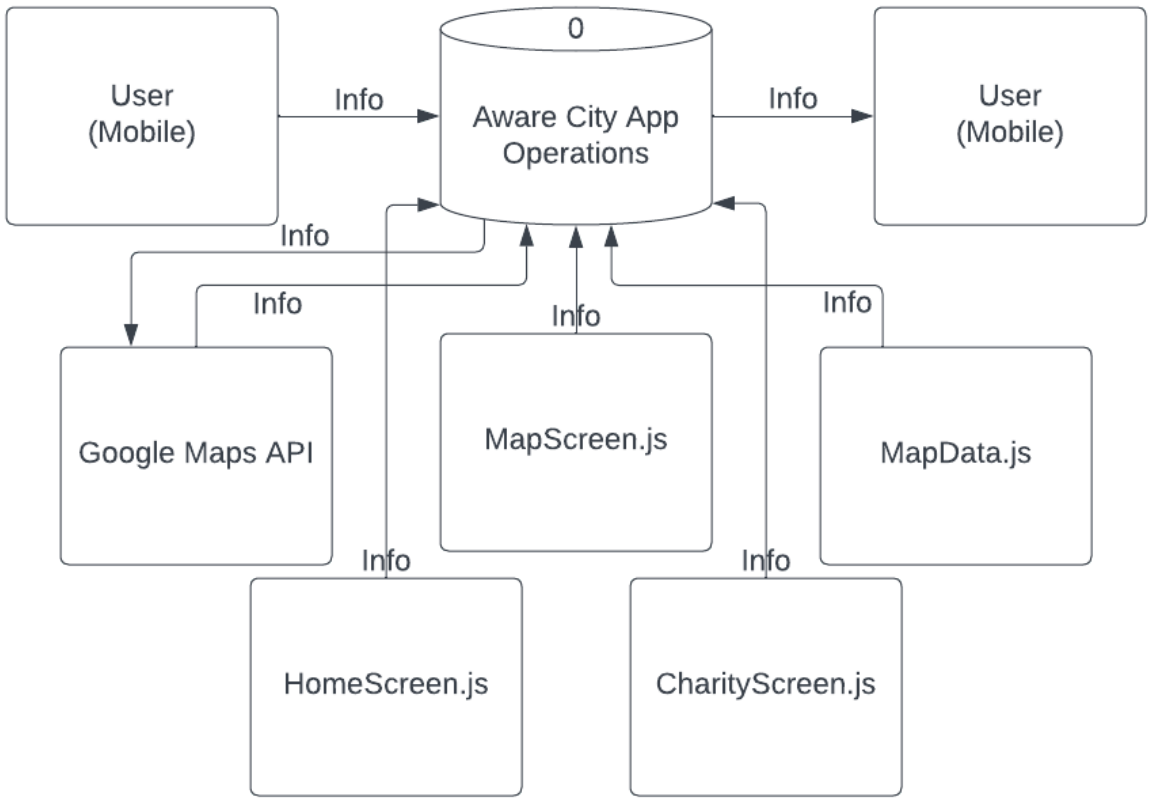
2.1 Activity List

Steps:	**User:**
1	User will be asked for permissions to access their location.
2	Once the users accept the locations permissions they will direct to the home screen which consists of many things.
3	If the user clicks on the hotlines container a

	pop-up will appear on the screen with many emergency hotlines.
4	If the user clicks on the questions container there will be frequently asked questions about the application answers.
5	If the user clicks on the charity container they will be redirected to the charity screen.
6	If the users click on the about us container they will get more information about the applications and the developers.
7	On the map screen, there will be a map displayed with many markers which are color-coded.
8	There will be a carousel card that the user can interact with that displays the nearest 10 services. Within the card, it contains information such as title, description of service, address, hours, and phone number.
9	The address and phone number on the carousel are all linked within the phone. So if the user decides to click on the address they will get directions from google maps and if the user decided to click on the phone number they have the opportunity to call the service being displayed
10	On the map screen there is also a featured button. Which contains all the color-coded services. The users can control what service they want to see on their screen.
11	In the charity tab, we provide information about the local non-profit organizations that can be beneficial for people who want to donate or volunteer.

2.2 Context Level Diagram

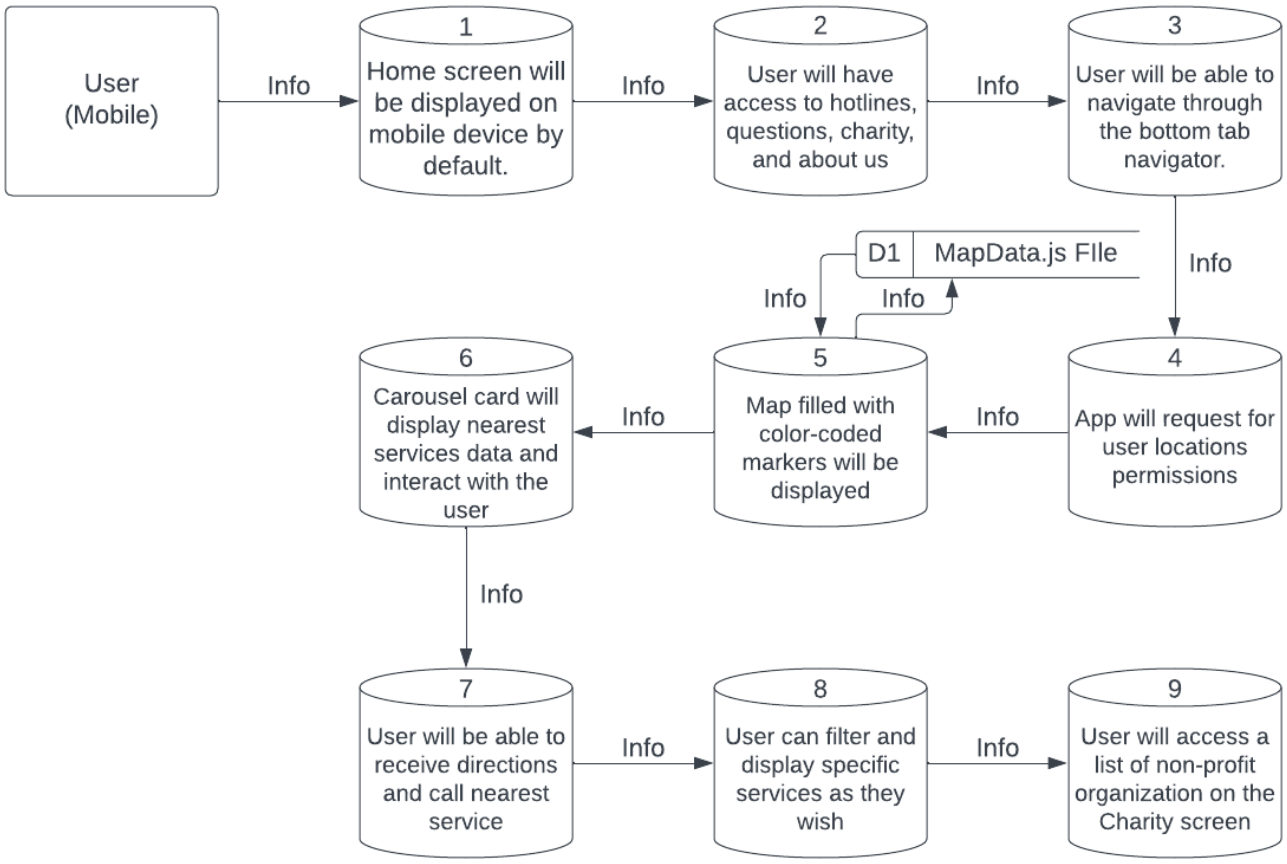
The Context Level Diagram presented below visually depicts the intended functioning of our Aware City mobile application. The application is intended for a specific audience and relies on various components, including the Google app API, MapData.js, HomeScreen.js, MapScreen.js, and CharityScreen.js files, to provide input for Aware City operations. The MapData.js file provides necessary data about services available in the community. Meanwhile, the other files will display the application's user interface, allowing users to interact with it.



2.3 Data Flow Diagram

The Data Flow Diagram presented below illustrates the navigation process of the Aware City mobile application. The application is accessible to users regardless of the device type they use. Once the user opens the application, they are directed to the home screen, where they can

access emergency hotlines, frequently asked questions, charity, and about us sections. By selecting the map icon on the bottom tab navigator, the user is redirected to the map screen, where they can view color-coded markers displaying specific provided services. The carousel card displays the provided information, which the user can interact with to receive directions or call the service provider. The map screen also allows the user to filter out which services they want to see displayed on the map. The charity tab lists non-profit organizations along with a brief description of what they offer. Additionally, users can visit the website of these organizations to donate or volunteer.

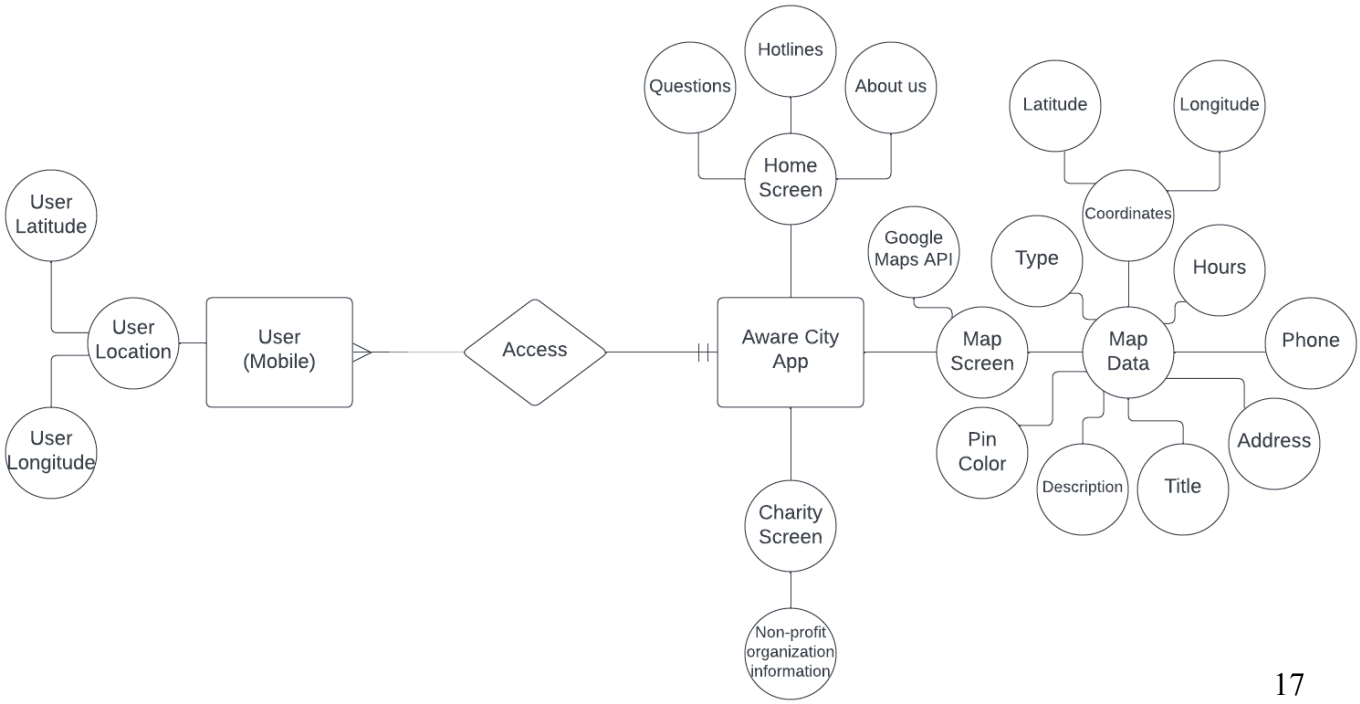


3.0 Data Structure Design

All of the data for our React Native mobile application is sourced from a file called "MapData.js." This file provides information on services available in New York City. To store this information, we chose to use an array that includes the coordinates, title, description of services provided, address, hours, and phone number of each service. The array also includes the color of the marker, which differentiates the services. We did not use a database for this application because none of us had experience adding a database to a React Native application. However, we did consider converting the array into a JSON file, which would make data retrieval easier and reduce time complexity. The image below illustrates how we stored the data.

3.1 Entity Relationship Diagram

The figure below shows the Entity Relationship Diagram that was created when we were working on building the Aware City mobile application. This diagram shows the relationships between our entities along with the attributes connected with them.



4.0 Functionality and Implementation

As we are building a cross-platform application, it is built to be functional on both Android and iOS devices. This is possible because we are using React Native and this framework allows cross-functionality using a single codebase. Another requirement for this application to function is having a WiFi connection so it receives the latest and most up-to-date updates from the developers. This is important because the markers that are used for the resources will be regularly updated as many locations may change information like times open and new locations will be added as we receive more data over time. By the use of Google Maps API, we built the map functionality in the app without having to write any code regarding the map from scratch. On the home screen, we were also able to implement direct hotlines to assist people in need as fast as possible.

This application can be used on anyone's devices, once downloaded from the Google Play Store or the Apple App Store. We intend on allowing our application to be implemented onto proprietary public devices like "LinkNYC" for easy and fast access to the public. We believe that this would be the quickest way to market this product and put it into the hands of the people who may need it the most.

4.1 Features

As for the features of this application, we had many ideas but focused on three main implementations that we thought would best serve the people in need in New York City. These three functions are built-in emergency hotlines, a map screen with live locations, and charity options for people who would like to donate using either their time or money.

The emergency hotlines are available on the home screen of the application. As you click on the "Hotlines" button, it will show a pop-up on the application which displays several hotlines

for different people who need assistance. The hotlines include but are not limited to; homeless youth, victims of crime, people with disabilities, and sexual assault victims. These hotlines may not seem important for many of us but there is a large population of people who are not aware that such resources and services exist. Our goal is to spread awareness of the existence of such aid in New York City, as per the name of our application, Aware City.

The second main feature we have added to the application is the map screen. This map screen is what allows us to provide available resources to the unhoused population and anyone who is looking for other resources like food pantries, soup kitchens, social services, and donation centers. By using colored markers, we were able to coordinate the different types of services that we wanted to showcase on the application. Users can filter each resource on the map by the use of our filter switches that add and remove markers from the screen. For example, if a user would only like to view the available social services around their current location, they can switch the social services filter on and be able just to see all the social services that are located on the map.

The last main function that was implemented on the application is the Charity page. On the charity page, we added a list of top organizations in New York City that users can donate to. The donations are not limited to a monetary amount but also include organizations that need your time. You can choose to donate cash to your choice of organization or decide to volunteer at one of these locations. If you click on your choice of organization, you can navigate their website and find more information on when, where, and how an individual can volunteer at locations like homeless shelters, food pantries, and donation centers. These types of locations are very much in need of help as they are limited in resources and funds. So we made sure to add organizations to the application that need such aid from the public.

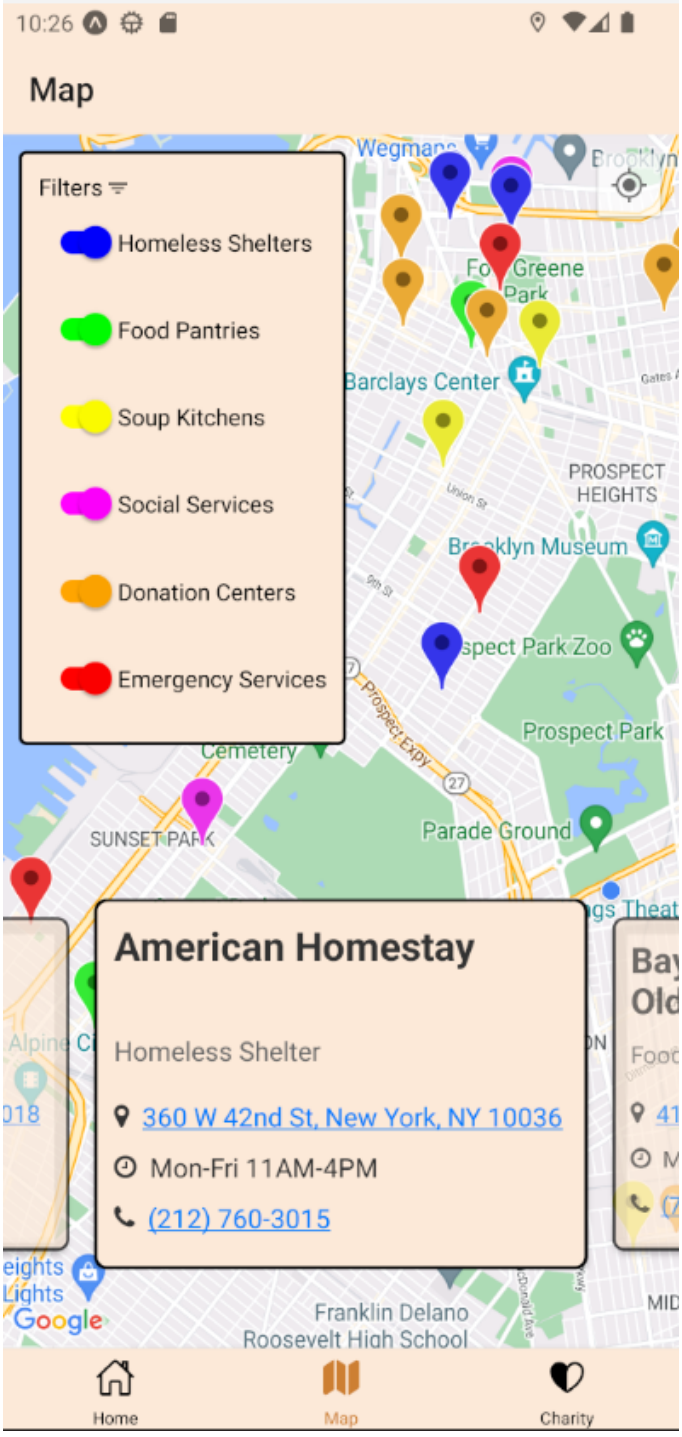
4.2 File Structure

The file structure that was followed for this project looks like the default structure of a freshly imported Expo app with a Node.js folder in the root folder. Some additional folders you may see are “src” and “components” which include buttons and dynamically loaded components on the application. Some other important folders we had to often visit and code on are the “routes” folder and the “assets” folder. As you can tell from the names, those folders included JavaScript code to handle the app screen routing and storing assets like pictures and logos, respectively. The majority of our time was spent in the “screens” folder which includes files like Home.js, MapScreen.js, and Charity.js. These were the main screens/pages of the application that the user interacts with and all the features are displayed on. And more importantly, we have the root folder where the App.js and MapData.js are found in. These two JavaScript files are very important because without App.js, there would be no possible navigation features on the application and MapData.js is our local database for all the markers used as resources on the map.

4.2.1 Example 1 (UI)

The following is an example of the user interface of the application from MapScreen.js. This screenshot was taken while the user is looking for assistance in Brooklyn. A few things you may notice on this screen are the filtering system, a carousel, a live location, and the bottom tab for app navigation. The filtering system was built to allow users to seamlessly navigate the data to find the services they’re looking for. Currently, all the switches are on, which allows us to show all the markers on the map. For instance, if only the “Social Services” switch was turned on, it would only show the Social Services markers around the user’s live location. The live location is represented using the light blue glowing dot on the map. For this, it's required to ask

the device for location permissions to be enabled on the first instance of opening the application. The carousel displays additional information about each marker. It is treated as a card that lists the title, description, address, hours, and contact number of the organization. The address and contact number are both clickable links. Clicking on the address will lead the user to the directions to the organization and clicking on the number would open the native phone app of the device to call the organization. And of course, the hours are displayed so the individuals in need will know when they can visit their destination. Finally, we have the bottom tab of the application which allows for simple navigation to other pages and features. On the bottom tab, you will see three icons; homepage, map, and charity. What we are looking at here is the map screen selected, while the homepage icon would take the user to the homepage to check out our other features and the charity page would allow the user to visit various organizations they can donate and/or volunteer at.



4.2.1a Code Snippet

4.2.1b Code Explanation

The code snippet above is taken from the MapScreen.js file, the screenshot of which was explained in section 4.2.1. This snippet presents the code that was used to display the filtering system we built for the application and how the data for each category correlates with the data that is displayed per marker. The code from lines 247-253 shows how each useState variable for a filter finds the correct data to show when selected or not selected. Each of them is mapped to its corresponding “type” from MapData.js where all the data is identified. And once the data is filtered by the selection of categories, it is then mapped to the markers that display all the information as shown in lines 256-262. Each marker contains information like the title, color,

description, contact number, and more. All of these are imported from MapData.js and assigned to local variables.

4.2.2 Example 2 (UI)

This image is a demonstration of our charity page tab. In this screenshot, you can see how there are five organizations. The purpose of this tab is to allow users to help out in the community by volunteering with top organizations in New York City that share a common interest in providing homeless individuals or those in need with a variety of services and resources. Each section displays the name of the organization, the logo, and a short description of what activities or impacts you'll be doing if you end up volunteering for those organizations. Along with that, you will also see a "Visit" button, which when clicked, directs you to the organization's website to view any further information. Each of those websites has a volunteering page on which you can register to volunteer whether by yourself or with a group and it also has a donation page for you to make any distributions.

4.2.2a Code Snippet

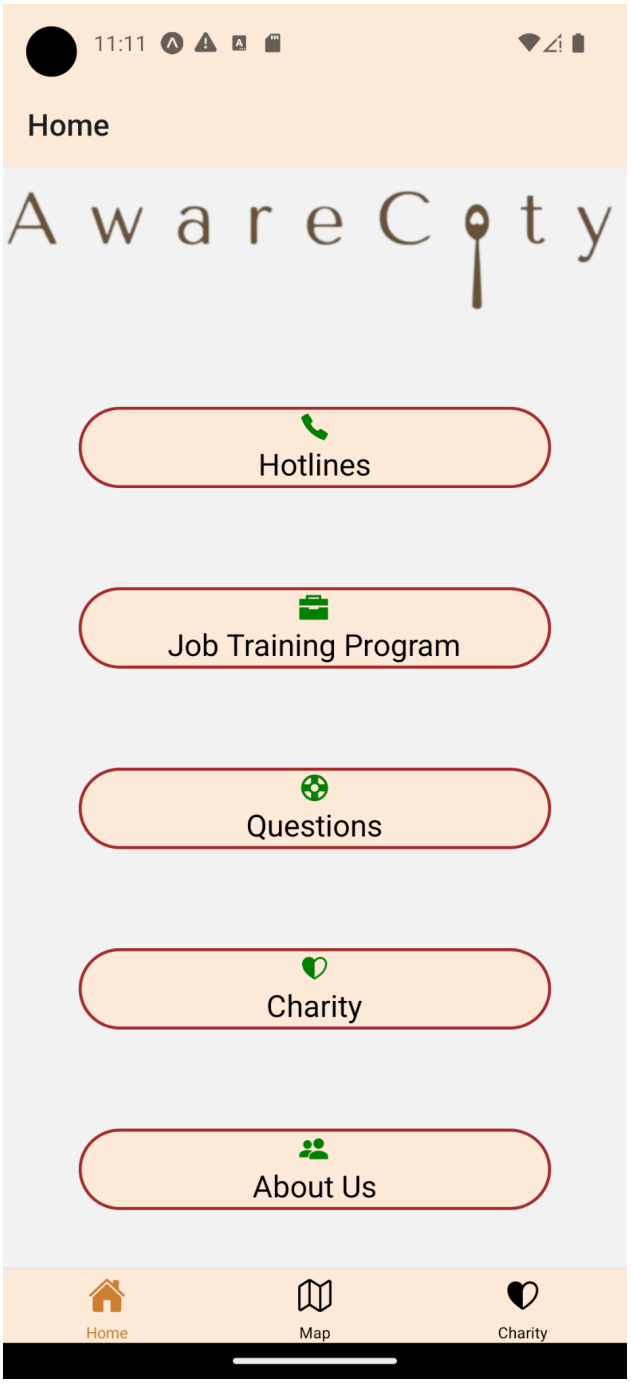
```
  return (
    <View style={styles.listItem}>
      <Image source={{ uri: item.photo }} style={{ width: 60, height: 60, borderRadius: 80 }} />
      <View style={{ alignItems: "center", flex: 1 }}>
        <Text style={{ fontWeight: "bold" }}>{item.name}</Text>
        <Text>{item.mission}</Text>
      </View>
      <TouchableOpacity style={{ height: 50, width: 50, justifyContent: "center", alignItems: "center" }} onPress={handleVisitPress}>
        <Text style={{ color: "green" }}>Visit</Text>
      </TouchableOpacity>
    </View>
  );
}

export default class App extends React.Component {
  state = {
    data:[
        {
            "name": "The Bowery Mission:",
            "mission": "Serve a meal, tutor and mentor clients as well as offer a special or professional skill to impact someone's life.",
            "photo": "https://pbs.twimg.com/profile_images/877222620554346496/J09Hv6KW_400x400.jpg",
            "link": "https://www.bowery.org/"
        },
        {
            "name": "Ali Forney Center:",
            "mission": "Provide nutritious meals, assist youth with clothing needs, facilitate workshops and host career days.",
            "photo": "https://fundforsharedinsight.org/wp-content/uploads/2017/08/2017-r1-grantee-aliforneycenter-1080x675.jpg",
            "link": "https://www.aliforneycenter.org/"
        },
        {
          "name": "Coalition for the homeless:",
          "mission": "Making a difference in  someone's life by making local stops and providing clients with hot meals.",
          "photo": "https://www.coalitionforthehomeless.org/wp-content/uploads/2015/12/CFTHLogo_InSquare.jpg",
          "link": "https://www.coalitionforthehomeless.org/"
        },
        {
            "name": "Women in Need:",
```

4.2.2b Code Explanation

The code snippet above displays the charity tab of our application. In the screenshot, you can see a few different non-profit organizations that contribute to providing assistance to those in need and making an impact in the community overall. For each section, there is an organization listed consisting of its name, image of its logo, their mission with a brief description of what you will be doing if you were to volunteer, alongside a link that if pressed, will direct you to the organization's website in case you want to get further information, sign up to volunteer or even donate. The list becomes scrollable after there is enough information for organizations listed on the tab.

4.2.3 Example 3 (UI)



The image displays the main home screen of the Aware City mobile application. Here, users can interact with different functionalities provided by our home screen mobile application. For example, if users wish to view the different hotlines available on our application, they can click on the "Hotlines" button. Each button on the home screen serves a different objective. The "Job Training Program" button assists users in locating job opportunities, while the "Questions" button displays answers to frequently asked questions about the mobile application. The charity button redirects users to the charity screen, which contains a "Flatlist" component of all the non-profit organizations in the NYC community. Lastly, the "About Us" button allows users to learn more about the purpose of the application and the developers who created it.

4.2.3a Code Snippet

```
<TouchableOpacity onPress={() => Linking.openURL('https://www.coalitionforthehomeless.org/our-programs/first
    <View style={styles.box}>
        <View style={styles.inner}>
        <Ionicons name="briefcase-sharp" size={20} color="green" />
            <Text style={styles.text}>Job Training Program</Text>
        </View>
    </View>
</TouchableOpacity>

<TouchableOpacity onPress={() => setModalVisible2(true)}>
    <View style={styles.box}>
        <View style={styles.inner}>
        <Ionicons name="help-buoy-sharp" size={20} color="green" />
            <Text style={styles.text}>Questions</Text>
        </View>
    </View>
</TouchableOpacity>

<TouchableOpacity onPress={() => navigation.navigate('Charity')}>
    <View style={styles.box}>

        <View style={styles.inner}>
        <Ionicons name="heart-half-sharp" size={20} color="green" />
            <Text style={styles.text}>Charity</Text>
        </View>
    </View>
</TouchableOpacity>
```

4.2.3b Code Explanation

In the code snippet, we can see how we implemented some of the buttons on the home screen. One of the primary components used in building the home screen application was React Native, specifically the "TouchableOpacity" component that converts containers into buttons. We also utilized the "View" component extensively, as it is simpler to divide it into individual containers. The "View" component is similar to the "div" element in HTML. To display icons on the button, we imported the Ionicons library and simply called the icon's name we wanted to show. Finally, we incorporated the Linking component, which enables the application to redirect users to the website for accessing first-step job training.

5.0 Software Testing

Throughout the development process, the team had several modes and levels of testing for the application. Some forms of testing that we have implemented are alpha, unit, compatibility, and integration testing. All forms of testing have a different purpose and

configuration that is being tested. Alpha testing is when the application is being tested by the developers as it is being built. This is typically before any public release or demonstration. Such testing can be carried out on local development environments like on our IDE which works simultaneously with either Android Studio or XCode. Both of this software allow developers to build simulations of either an Android or iOS device. Through this type of testing, we were able to focus on detecting and fixing bugs right after implementation.

Regarding implementations for the application, unit testing was performed for individual components before it was merged with the main branch. Using Github, we were able to create multiple branches that were dedicated to building separate features of the application, allowing for unit testing. This type of testing involves testing each unit (feature/page) of the application in an isolated branch, to be sure that the feature worked as intended. Once we received confirmation, we were then able to confidently merge that branch with the main. Once we merged with the main, we took it a step further than simulation testing and implemented compatibility testing.

With access to technology like Expo, we were allowed to test the application in real-time with our mobile devices in hand. Once the Expo app is downloaded on either an Android or iOS device, our local development environment can start an instance of the application through VSCode and Expo. This environment generates a QR code that can be scanned by our devices, opening up an instance of the application. This was one of our favorite features of using Expo as this was our method of compatibility testing with both operating systems in hand. Without this, we are unable to test the speed of the application, compatibility of the features, and connectivity of all aspects of the build.

By taking advantage of this mode of testing, we were able to test another level of the application through integration testing. One example of integration testing was to test the map screen features by inputting many different locations, viewing as many markers as possible, and testing the filtering system. This was done to identify any issues that may arise with the multiple integrations we had on a single screen.

6.0 Limitations

One current limitation of our application is its display on larger screens. While the user interface is designed for phone-sized dimensions and is user-friendly, it looks off and clunky when downloaded on an Android or iPad. Another challenge we faced was obtaining data on the different resources. Despite searching for an API, database, JSON file, or CSV file containing information on these services, we were unable to find anything suitable. Consequently, our team took matters into our own hands and ensured that we had accurately imported information about the various community resources. However, during the data import process, we discovered that some services did not provide phone numbers or hours of operation, which made it challenging to incorporate them into our application.

6.1 Future Enhancements and Recommendations

Some future enhancements that can be worked on are the dynamic functionalities of the application. As the application has a very long-term vision, we would aim on building an infrastructure that allows for simple updating functions and backend management systems. Onto our React Native framework, we can implement a content management system (CMS) like Sanity or Contentful that allows for dynamic data updating. We would mainly be using this to update map data from third-party software like Sanity, which would allow the developers to add and update the markers without committing or pushing any code. The markers may have

outdated information like the time and location or there will be new resources that are available in New York City that will be needed to add to the platform. The main idea behind having a CMS is to allow for easy and fast updating of data that will dynamically update our application without the need of touching the codebase. Even a non-technical individual would be able to take care of the data management.

We are also planning on allowing the users who use the mobile app to create an account so that they can have more future features. Such as being able to favorite a specific service they are interested in seeing most often. Being able to leave a review or a comment on a specific service that they just experienced. We didn't want to make the users have to create an account to access the mobile app but we would like the users to create an account with us because then we will be able to provide them with notifications about events at non-profit organizations that are relevant to their interest.

6.2 Team Members

- **Brenda Encarnacion** - Creating this application with React Native, Node Js, and Javascript, was definitely a learning experience for me as it was a new concept. I ended up learning how to use all the new technology after watching numerous videos and reaching out to my teammates when I came across something I didn't understand. The most challenging part about this process was downloading all the tools necessary to always have the most current update on our application's progress. As it was the first time that I ran most of the tools, I had a lot of bugs and spent a lot of time debugging. I am glad to have effectively learned how to push and pull projects on GitHub to Visual Studio Code, run an IOS simulator, and code with react native and javascript. My main

contribution to this application was creating the charity tab. I also contributed to providing data for all the Queens locations and resources on the map screen.

- **Shajid Muntaser -** This project was my first React Native build and it was a great experience since we opted to use Expo tools and testing. It was cool seeing our product being built in real time on our phones and simulators as we programmed the features. My main contribution was building the user interface of the application. I worked on features like the navigation bar, the page build, and the routing architecture. We aimed to build a fast and easy-to-use app, and I think we achieved that. The biggest challenge I faced was getting familiar with keywords used in React Native like <Text>, <View>, <TouchableOpacity>, and more. This syntax is very different from frameworks I've used in the past like React or Next.js.
- **Fredy Sorto** - In this project, I learned how to build my first React Native project. Using the prior knowledge I learned from developing mobile applications using Swift and building websites with React JS I was able to connect the knowledge I gained from those two to work on this React native application. My main contribution to this project was developing the UI and the functionality of the application. My main focus on this project was to have a well-designed map screen that would be user-friendly and easy to navigate through and understand. I also took a big part in making the data structure and importing the data for our application. I believe looking for an API, a database, or a JSON file with all the data of the provided services we were looking for to include in our app was one of the biggest challenges to be quite honest because we had to take the time to make sure we imported accurate data. Lastly, I truly enjoyed building this application and I hope to

continue adding more features that can be very beneficial as I think our application can leave a positive impact on the community.

Works Cited:

- Chau LamPublished Jan 11, 2023, Modified Jan 11, 2023, at 8:02 p.m. Elizabeth ShwePublished May 9. “What You Need to Know about NYC’s Upcoming Street Homeless Count.” *Gothamist*, gothamist.com/news/what-you-need-to-know-about-nycs-upcoming-street-homeless-count. Accessed 10 May 2023.

  Editor. “The Good and the Bad of Reactjs and React Native.” *AltexSoft*, 27 Feb. 2020, www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/.
- *Homeless Resources - Shelter App*, www.shelterapp.org/. Accessed 10 May 2023.
- “Homeless Ministry & Discipleship in Dallas.” *OurCalling*, 3 Nov. 2022, www.ourcalling.org/.
- “The Win App • OCLA.” *OCLA*, 8 Feb. 2022, oclawin.org/our-programs/the-win-app/.