FetchZo: Real-Time Mobile Application for Shopping in COVID-19 Pandemic Situation

Sudhish Subramaniam and Subha Subramaniam

Abstract  In this paper, FetchZo real-time application is developed for the shopping purpose in COVID-19 pandemic situation. Coronavirus is a serious virus in India, which spreads via close contacts. Maintaining social distancing becomes more important to avoid coronavirus infection. Shopping for daily needs is a great issue with social distancing and have come up with FetchZo application, which will locate your nearest shop and update you with many people currently present in the shop. This sustainable application provides security with login credentials for the shopkeeper as well as the customer. The shopping list also can be updated by the shopkeeper thus providing a real-time application for this pandemic COVID 19 situation.

Keywords  COVID 19 · Coronavirus · Shopping application · Social distancing

1 Introduction

In the critical situation of COVID-19, the only way to prevent coronavirus is by social distancing. Keeping this in mind an application has been developed, which would ease shopping of essential items during a lockdown situation. The lockdown has forced everyone to stay at home. This brings up the major problem of availability of essential items such as rice, wheat, medicines, and other groceries. To help people staying at home with the essential items this model has been proposed. At the initial stage, the model takes input from the shopkeepers about their stock availability. The user is entitled to use the application to get the nearest shop, having the required...
The model also calculates the number of people in each shop and hence gives the optimum shop which has the least number of people and is the closest among all.

2 Literature Review

Aarogya Setu (The scaffold for freedom from infection) is an Indian open-source COVID-19 “Contact following, Syndromic application, and Self-appraisal” computerized administration, fundamentally a portable application, created by the National Informatics Center under the Ministry of Electronics and Information Technology [1]. The application arrived at in excess of 100 million introduces in 40 days. On 26 May, in the midst of developing protection and security concerns, the source code of the application was made open. On second April 2020, India propelled the Aarogya Setu versatile application.

Application for aiding and expand the endeavors of constraining the spread of COVID19, to empower Bluetooth based contact following, use of likely hotspots and dispersal of pertinent data about COVID19. The application has more than 114 million clients as of 26th May, which is more than some other Contact Tracing Application on the planet. The Application is accessible in 12 dialects and on Android, iOS, and KaiOS stages. Residents the nation over are utilizing Aarogya Setu to secure themselves, their friends and family, and the country. Numerous adolescents likewise call Setu as their Bodyguard.

The key mainstays of Aarogya Setu have been straightforwardness, protection, and security and in accordance with India’s arrangement on Open Source Software, the source code of Aarogya Setu has now been made open-source [2, 3].

3 Overview of Fetchzo Application

The application comprises two parts, namely the shopkeeper side and the customer side. At the shopkeeper’s side, he/she can update his stock details and the model stores in the database. FetchZo algorithm runs on the database to find the exact location and of the shop [4].

At the customer’s end, the customer’s location is accessed and he/she is asked to select the required items with the specified quantity. This is again uploaded to the database, on the customer table [5].
FetchZo application fetches the location of the customer and reaches the nearest shops in map. The distance between the customer and the shops are calculated and displayed on the map [6]. The FetchZo application turns on Bluetooth of customer device during installation of the application. The Bluetooth algorithm calculates the number of people in the shop displays it on the map [7, 8].

Using the three parameters availability of the item, distance of the shop from the customer and the number of people in the shop respectively, FetchZo application displays the optimum shop on the screen [9, 10].

In a pandemic situation, people still have to stand in long queues for the items. To overcome this problem the model has a token button. Once the optimum shop is being displayed a counter token button is displayed which will be updated with runtime. It sends a message to the shopkeeper about the customer and his details and his time of the visit to the shop. The shopkeeper can check the list of required items and can be ready with the items as per the token time. This reduces manual token systems which are existing in all our Indian cities. It provides social distancing which can prevent shopkeepers and customers from infections and direct contact.

Only dmart, Star and Supermarkets are surviving which attracts large queues and crowds. By installing this FetchZo application, small shop vendors can also survive without fear. This will help in maintaining social distancing in the shops and avoiding the crowd. Surely the application will be very helpful to both the customer and the shopkeeper keeping the COVID-19 pandemic in mind.

Figure 1 depicts the data flow diagram of FetchZo application. It initially filters nearby shops having a stock from the customer side. It counts the number of people present in the shop and compares it with other shops. It selects the shop with a minimum number of people. It takes distance as a parameter and compares it with the existing shops. This FetchZo application gives social distancing as the highest priority in this COVID-19 situation. It takes shop distance from the customer side as the second parameter and displays the optimum shop.
4 Implementations of Fetchzo Application

As shown in Figs. 1 and 2 the application flows in two parts, customer side and shopkeeper side.

At the shopkeeper’s side (Fig. 2), the application starts with the Bluetooth scan. It detects the devices around it and redirects the page to the shopkeeper login. Here it checks if the account is pre-existed, if yes then login else it creates an account. Once the shopkeeper has logged in, he/she has to enter the stock details which is updated to the database. Now the location of the shopkeeper’s device is automatically tracked and updated to the database.

At the customer’s side (Fig. 3), the application starts the Bluetooth scan and stores the nearby devices. After the login procedure, it displays the essential items required during the lockdown. It asks for the quantity of the item selected. Once an item is selected the items with quantity gets added to the cart. The items are uploaded to the database, the algorithm checks for the items in the nearby shops of the customer’s location. The shops with the essential items needed are filtered. Now the algorithm

![Shopkeeper side flowchart](image-url)
checks for the number of people present in each shop. It also checks the distance of the shop from the customer’s location. With the number of people in the shop having priority, the optimum shop is displayed in the application. The token number is generated and the optimum time to visit the shop is also assigned [11, 12].

Fig. 3 Customer side flowchart
4.1 FetchZo Algorithm

- Detecting the nearest shop with the given location
- Calculating the least number of people in the shop
- Providing a token for avoiding the queues

4.2 Detecting the Nearest Shop with the Given Location

Shops register in the application with their respective locations and the details. The shop owners will have to upload their stock details. At the customer side, the user will register his location and his details, this data will be recorded in the database. In the database with the entries of both the tables. Those shops are filtered out which have the required items. The location of the customer is compared with the location of the filtered-out shops in the shop owner’s table. By this, the nearest shop can be found.

4.3 Calculating the Least Number of People in the Shop

This is done with the help of Bluetooth Low Energy (BLE). The shop owner’s phone becomes the host and it detects the number of people inside the shop using Bluetooth. This information gets uploaded in the database and hence it helps in evaluating the number of people in his shop.

4.4 Attaching a Token

To avoid long queues in the shops a counter button is made which is connected to the database. With the increase in the counter buttons, the model allows a time starting from morning 8 o clock. The application can have its online payment system (example UPI) which will help the customer to do online payments, keeping digital India and social distancing in mind. Once the online payment is done after verification of available items, it becomes convenient for the customer to reach the shop and collect shopping items. This ensures little contact and maintains social distancing.
5 Results and Snapshots

5.1 At Customer Side

Snapshots of the FetchZo application in real-time are shown below. Figure 4 depicts application initial page. This page gives the option to login as a customer or the shopkeeper. Snapshots of the application as shown below Fig. 5 depicts the cart view of the customer.

Snapshot of the application in Fig. 6 shows the application, detecting the location of the customer. Figure 7 shows the application, displaying the number of shops around the customer.

![Fig. 4 Login page](image-url)
5.2 Shop Keeper Side

The stock details of the owner are being updated in the database.

Figure 8 shows the login page at the owner’s side. As shown in Fig. 9, application asks to add the items to the owner’s side.

After displaying the image, it gives the option to enter the item and the quantity of the item to be added in the shopkeeper database. Figure 10 depicts the image of this. Figure 11 shows the items added and shown on the application page. As shown in Fig. 12a, all the items in the cart are uploaded to the database. These are the images from the application developed using Android Studio. The software is user friendly and allows the user to develop android applications.

As depicted in Fig. 12a, b the login information and authentication of both the customer and owner are saved in the database.
Figure 13 shows the saved page of the location. The location of the owner is being analyzed and prepared for comparing with the location of the customer. Figure 14 depicts the calculation of the distance between the shops and the customers.

6 Conclusion

As the COVID 19 situation has become very fierce and the situation is getting complex day by day, would like to present this FetchZo application model so that it will ease the situation and help people in shopping. In future, the model will be surely useful for the day-to-day essential items. This will not only increase the economy but also would prevent the spreading of coronavirus and will help to avoid long queues in the shops. Getting the required items, finding the nearest shop, avoiding contact with people hence following social distancing, avoiding cash payment, these are some of the major advantages of the application. It can be concluded that FetchZo model(application) will surely help people in the lockdown situation of COVID-19.
Fig. 7 Displaying the nearby shops with distance

This application will make shopping easier and also will help India fight against COVID-19 by social distancing and avoiding contact.
**Fig. 8** Login page of Owner

**Fig. 9** Add Items
Fig. 10  Adding Items-Owner Side

Fig. 11  Items-Owner Cart
Fig. 12  a The Items of the cart are uploaded to the Database, b Customer and Owner Database

Fig. 13  Saved Location Details of FetchZo Application
Fig. 14 Calculation of distance between the shops and the customers

References

1. nic-delhi/AarogyaSetu_Android, nic-delhi, 27 May 2020, Retrieved 27 May 2020
2. Jump up to:a b“Govt launches ‘Aarogya Setu’, a coronavirus tracker application: All you need to know”. Livemint, 2 April 2020. Retrieved 5 April 2020
3. Aarogya Setu Mobile Application. MyGov.in. Retrieved 5 April 2020
4. Sridhar, S., Smys, S.: A hybrid multilevel authentication scheme for private cloud environment. In: 2016 10th International Conference on Intelligent Systems and Control (ISCO), pp. 1–5, IEEE (2016)
5. Palanisamy, R.: The impact of privacy concerns on users’ trust, attitude and intention of using a search engine: a conceptual model and research issues. Int. Acad. Res. J. Business Manage. 2(3), 31–43 (2013)
6. MohdAnis, T.B., Subramaniam, S.: Operating system: the power of android. Int. J. Sci. Res. (IJSR) (Online) 03(11), 1914–1919 (2014)
7. Marcario, G., Torchiano, M., Violante, M.: An invehicle infotainment software architecture based on google android. In: SIES, pp. 257–260. IEEE, Lausanne, Switzerland July 2009
8. Nogueira, L., Pinho, L.M.: Time-bounded distributed qos-aware service configuration in heterogeneous cooperative en-vironments. J. Parallel Distrib. Comput. 69(6), 491–507 (2009)
9. Higuera–Toledano, M.T., Issarny, V.: Java embedded real time systems: an overview of existing solutions. In: Proceedings of the Third International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2000), pp. 392–399. IEEE Computer Society, Washington, D.C., USA (2000)
10. Guerra, R., Schorr, S., Fohler, G.: Adaptive resource management for mobile terminals— the actors application. In: Proceedings of 1st Workshop on Adaptive Resource Management (WARM10), Stockholm, Sweden, April 2010
11. Maia, C., Nogueira, L., Pinho, L.M.: Experiences on the implementation of a cooperative embedded system framework. CISTER Research Centre, Porto, Portugal, Tech. Rep., Jun ’10
12. Ahmad, M.S., Musa, N.E., Nadarajah, R., Hassan, R., Othman, N.E.: Comparison Between Android and iOS Operating System in terms of Security. School of Computer Science, Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia 2013 8th International Conference on Information Technology in Asia (CITA)