



Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

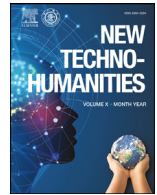
Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.



ELSEVIER

Contents lists available at ScienceDirect

## New Techno-Humanities

journal homepage: [www.elsevier.com/locate/techum](http://www.elsevier.com/locate/techum)

## A comprehensive easy-to-use techno-human solution for the resource-constrained nations in the fight against communicable disease like the coronavirus pandemic

Omkar Manjrekar<sup>a</sup>, Rathin Biswas<sup>a,\*</sup>, Aditya Agrawal<sup>b</sup>, Nilesh Barandwal<sup>c</sup>, Abhishek Gupta<sup>d</sup>, Pranjali Chaudhari<sup>d</sup>, Sharad Mishra<sup>a</sup>, Kedar Anavardekar<sup>a</sup>, Vikrant Fernandes<sup>a</sup>, Kavi Arya<sup>a</sup>

<sup>a</sup> Indian Institute of Technology Bombay, Mumbai 400076, India

<sup>b</sup> Dr. Akhilesh Das Gupta Institute of Technology & Management (Affiliated to Guru Gobind Singh Indraprastha University), New Delhi 110053, India

<sup>c</sup> Pune Institute of Computer Technology, Pune 411043, India

<sup>d</sup> Sardar Patel Institute of Technology, Mumbai 400058, India

## ARTICLE INFO

## Keywords:

COVID app  
Public health  
Contact tracer  
Self-assessment  
Infection tracker

## ABSTRACT

Following the advent of the novel Coronavirus, the governments and authorities introduced various mobile applications in the fight against COVID-19. However, these are either restricted by the developer country or not comprised of certain key features, due to which, a vast population from the low- or middle-income country, remains underprivileged from associated benefits. The purpose of this study is to explore and demonstrate how to build a comprehensive and easy-to-use application for mobile health service delivery especially fabricated to fight communicable diseases like COVID-19. Mostly open-source technologies are used to build a distributed scalable client-server application. Then the application is evaluated with the stress-test and usability-test. Besides COVID-19 advisories and guidelines, the study demonstrates complete conceptualization and development of an IoT-based mobile application. It comprises three key features that bring comprehensiveness. One is the 'contact logger & tracer' that enables users to maintain a list of contacts who get notified if any of them reported COVID-19 positive in the last 14 days. Two is the 'self-assessment' which helps users to predict the chances of COVID-19 infection using a scientific rationale. Three is the 'infection tracker' that guides users to identify infected hotspots to plan the route accordingly. The usability-test affirmed that the application is easy-to-use. Further, this study demonstrates how to construct such an easy-to-use application mostly with open-source resources. It can help the needy authorities or groups from resource-constrained countries to adopt and develop such applications quickly. Further research on the post-implementation effect will add value to this study.

## 1. Introduction

## 1.1. Literature study

The pandemic COVID-19 has shaken the world as it claimed millions of lives so far (Battineni et al., 2020) and it has surpassed most of the predictions. The international authorities and the national governments across the world have initiated or adopted several measures to deal with the COVID-19 and associated issues. As the internet plays an important role in the modern world, the adoption of internet-based web/mobile applications became one of the major equipment for the managing authorities in providing efficient medical assistance or health services (Brownstein et al., 2009; Yusif et al., 2020). There are many apps developed by various nations for mass surveillance and epidemic

or pandemic control (Calvo et al., 2020). For example- the 'Hamagen' (Winer, 2020) app by the Israeli Government alerts and advises users for self-quarantine if they have crossed an infected person's path. In China, an app is made with which authorities may decide someone's access to public space by collecting their health status and location data (Mozur et al., 2020). 'Clearview AI'- the startup in the USA was built to monitor infected people and enforce isolation (Grind et al., 2020). Despite so many apps developed by various nations or authorities or agencies, there are two major problems that were observed- accessibility and lack of comprehensiveness. Accessibility means the unavailability of a COVID-related app to various nations other than the nation which developed that particular app, due to the policy by the 'Google Play Store' or 'Apple App Store'. Comprehensiveness means the lack of all the necessary features in one single app. The combination of both these problems results in a limited or a lack of mobile health services and advisories to the public. This is further limited by 'Google Play Store' or 'Apple App Store' policy, as it doesn't allow hosting COVID-apps from various developers other than the apps backed by the corresponding Governments

\* Corresponding author.

E-mail address: [r.biswas@iitb.ac.in](mailto:r.biswas@iitb.ac.in) (R. Biswas).

<https://doi.org/10.1016/j.techum.2022.05.002>

Received 14 August 2021; Received in revised form 1 May 2022; Accepted 15 May 2022

Available online xxx

2664-3294/© 2022 The Author(s). Published by Elsevier Ltd on behalf of Shanghai Jiao Tong University. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

Please cite this article as: O. Manjrekar, R. Biswas, A. Agrawal et al., A comprehensive easy-to-use techno-human solution for the resource-constrained nations in the fight against communicable disease like the coronavirus pandemic, *New Techno-Humanities*, <https://doi.org/10.1016/j.techum.2022.05.002>

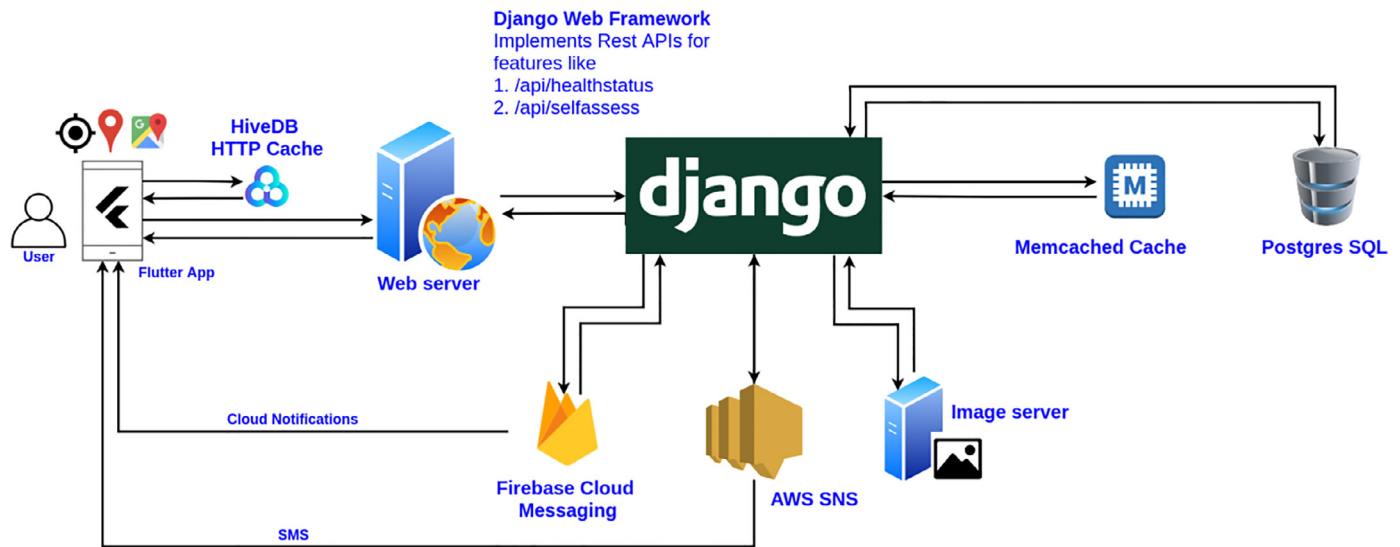


Fig. 1. System design.

following authenticity and restricting the flooding of apps. As a result, there is a chance that many comprehensive apps are unable to reach the users.

After careful scrutiny, it is found that most of the apps mostly focus on the COVID-19 advisories and guidelines, and either these apps have not considered comprehensive contact tracing or haven't considered it at all. Contact tracing is essential to monitor and contain the spread of a communicable disease or pandemic such as COVID-19 and it is also necessary to get a complete list of the identity of the people who came in contact with someone who just got infected without a moment of delay. Some of the apps are used to track the spread of coronavirus and collect self-reported information (Mayor, 2020), but the conceptualization of getting chances of COVID from those self-reported responses, are found either not inclusive or not backed by the proper literature/research. Another paucity is found in the visualization of information. For example, one can be cautious or may find an alternate route while traveling towards the destination just by keeping an eye on a map-based visualization tool. These are the crucial limitations observed that might improve mobile health services. However, it must respect users' privacy (Cho et al., 2020; Owoh & Singh, 2020).

### 1.2. Purpose

The purpose of the study is to explore how to build a comprehensive and easy-to-use application for mobile health service delivery especially fabricated to fight communicable diseases like COVID-19.

## 2. System design

As the mobile application aims to protect us from communicable diseases like COVID-19, it has been named as "Rakshak", which is basically a Sanskrit or Hindi word that means "The Protector". A user of Rakshak accesses the system using a mobile application that is developed in Flutter. Flutter (Napoli, 2019) is a reactive cross-platform application development framework developed by Google. It considerably reduces time (Olsson, 2020) to develop apps that are User Interface (UI) based and don't require heavy on-device processing or use hardware such as Bluetooth or Accelerometer. If still required the native features can be programmed using the platform channels feature.

The system design is demonstrated in Fig. 1. The app utilizes some device Application Program Interfaces (APIs), the Rakshak ReST API, and third-party APIs from Google like Maps, Places Autocomplete,

Firestore (Moroney, 2017), and Firebase analytics (Moroney & Anglin, 2017) to implement features like choropleth maps, address search, reverse geocoding, notifications, and analytics. A RESTful API or ReST API (Fielding & Taylor, 2002) is an application programming interface (API) that uses HTTP verbs to perform operations on data. Rakshak ReST API is implemented using Django ReST Framework. Caching with and without the support of the Rakshak backend is leveraged to reduce network bandwidth and battery consumption.

The Django (Forcier et al., 2008; Holovaty & Kaplan-Moss, 2009) backend relies on PostgreSQL (Drake & Worsley, 2002) and its extension PostGIS (Strobl, 2008) for storing and querying the data. It uses a dedicated image server to transfer images of optimum quality and size. SMS service of AWS was used to send OTP, FCM to deliver cloud notifications and in Contact Tracer implementation. Memcached (Nishtala et al., 2013) is used as the in-memory database which is an open-source, high performance, distributed memory object caching database which makes use of the LRU replacement policy by default, making it extremely suitable for our use case.

### 2.1. Mobile app architecture

Clean Architecture (Boukhary & Colmenares, 2019) was adopted to build the mobile app as it enables separation of business logic and UI, and provides flexibility in the long run. Dependency injection (Prasanna, 2009) was done using get\_it (Burkhart, 2021), an open-source service locator library. The architecture contains 4-5 layers each dealing with a different aspect of the application.

Fig. 2 demonstrates the various layers of mobile app architecture- (1) Presentation layer that is responsible for the UI of the application, (2) Domain layer that includes the business logic of the application and provides the services to the presentation layer, (3) Data layer that communicates with on-device databases and/or with backend servers and provides services the domain layer, (4) Core, which consists of Classes that are useful to any application in general such as data validation, caching, cloud messaging (notification service), HTTP implementations, etc., and (5) External, which are 3<sup>rd</sup> party implementations, used to implement an abstract class or interface in Core so that the external library can be changed without changing code that depends on it.

### 2.2. Requirements

The self-explanatory Table 1 briefly explains about the functional and non-functional requirements.

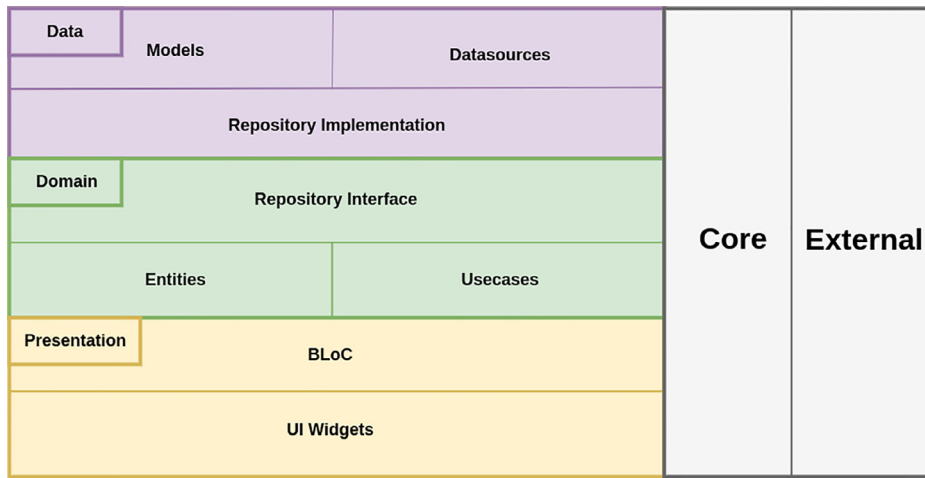


Fig. 2. Layers of mobile-app architecture.

Table 1  
Functional and non-functional requirements.

Type	Feature	Explanation
Functional	Login	OTP based login using a phone number
	Health Status	A simple questionnaire which a user answers to update his/her health status.
	Self-Assessment	A series of questions to assess possibility of getting infected by COVID-19
	Contact logger	A form-based contact logger that lets a user add details of a person who he/she has come in contact with. The receiver gets an SMS or a notification when they are added by someone and they can approve or deny the association.
	Contact Tracer	Contact tracer notifies every contact of a person who identifies himself as COVID positive or symptomatic through our Health Status questionnaire.
Non-functional	Choropleth Map	Choropleth Map enables users to easily visualize complex data on Map.
	Security	A user can be added to the contact logger without revealing his contact information using QR Code.
	Caching	The system uses Memcached as the backend for its caching mechanism to improve response time of the system.
	Ease of use	The application UI is easy to use, evident from the system usability test. Specially designed widgets are used to reduce a user's effort such as selecting a user from phone contacts, auto-complete addresses, pin location on maps, and OTP features to add a new contact.

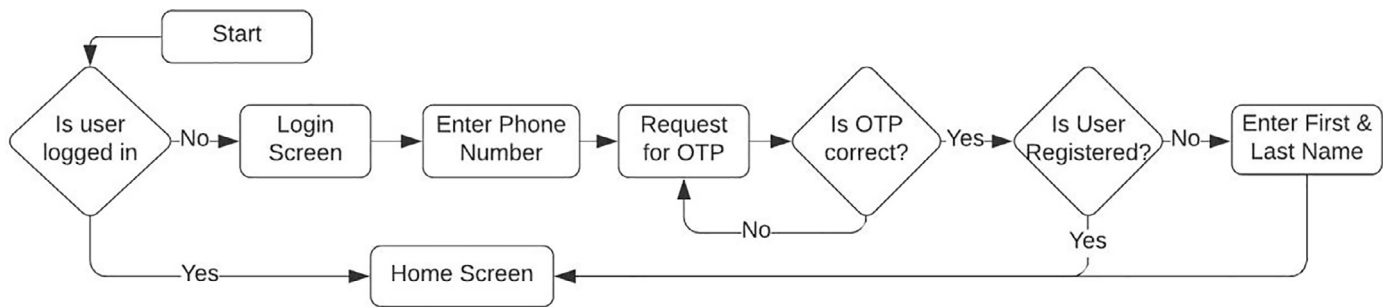


Fig. 3. Flowchart for login.

### 3. Functionalities

#### 3.1. Login

In order to log in to the application (Fig. 3), the user is required to enter their phone number. If the phone number entered is correct, then an OTP request is sent to the user for authentication. The user then enters the OTP in the OTP field provided. The server then checks whether the OTP submitted is correct or not. In case there is a mismatch in the OTP submitted by the user, then the user can ask for a different OTP or if the user wants to change the phone number then they can go back and change it. After successful OTP verification, the server checks if the verified phone number is already registered earlier to the database or not. If it is already registered, then the user is redirected to the Home Screen and can start using the application. If not, then the user has to enter basic information such as first and last name so that a new account can be created for the user on the backend server. Once the details are

filled by the user and the account set up is completed, the user gets logged in to the application and is directed to the Home Screen and now can access all the features of the application.

#### 3.2. Contact logger & tracer

##### 3.2.1. Contact logger

Contact logger is a feature implemented where a user A adds a contact and logs information about user B when they come in contact taking minimal information of user B, thereby keeping in mind the data privacy. The contact logger form is designed in such a way that asks for minimum data as possible. The contact logger form fields (mandatory and optional) can be seen in Fig. 4. A user can be added in the logs by two different ways, i.e, by manually filling the contact logger form or scanning the QR code of the contacted user.

- Manual: User 'A' manually enters the contact information of contact 'B' such as Name and contact number.

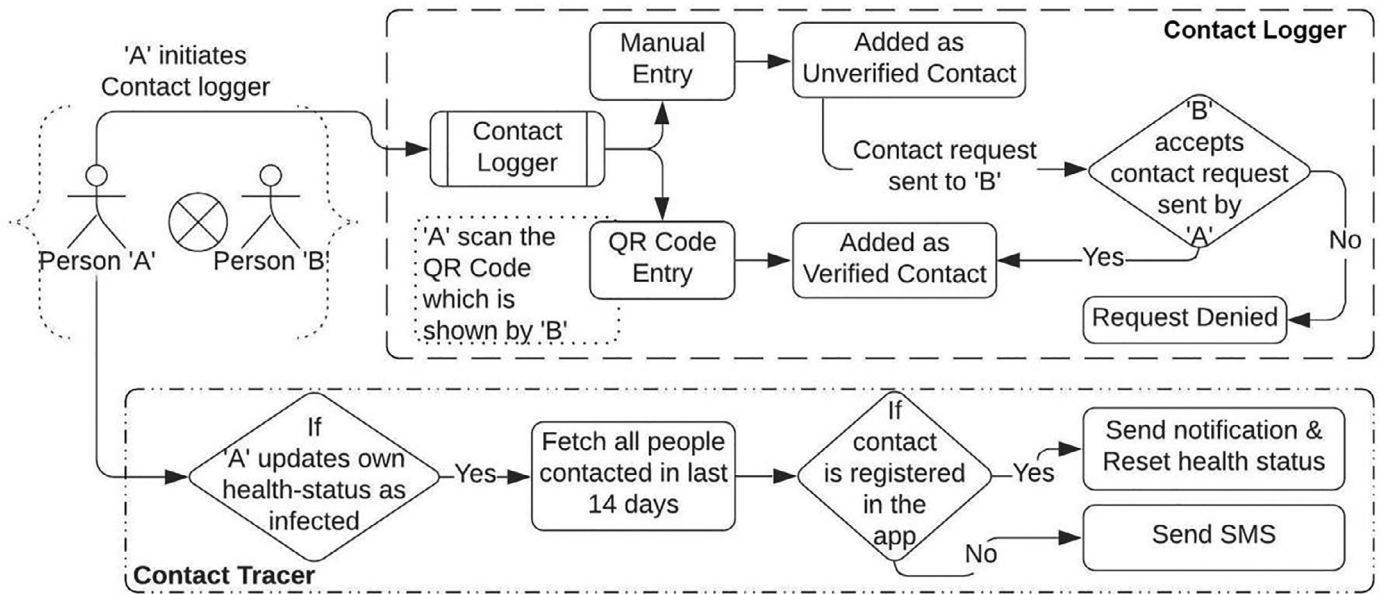


Fig. 4. Contact logger and tracer.

- QR: User 'A' can scan the unique QR code of user 'B' containing the encrypted name and contact information of the other user while keeping the personal details unrevealed.

The date of contact can be selected by the date picker dialog box. The place of contact is dynamically filled as the real-time location of the users which can be precisely altered (if required) by moving the pin/marker on google maps. Additional fields like attaching an image, a note, email id is optional. An added contact person receives a notification/SMS. Once the person accepts the request confirming the requested user has met him/her recently, the contact is added with a verified token.

### 3.2.2. Contact tracer

The user needs to take a health assessment test every 14 days since it may take 2-14 days (CDC, 2021) to appear symptoms after exposure to virus or when they've come in contact with someone who's already tested positive for the virus. If a user is found to be infected, all the people added by them to their contact log within the last 14 days (Fig. 4) will be sent a push notification and would be prompted to retake the health test. SMS about coming in contact with an infected person is sent to people not registered on the app.

### 3.3. Health status

The 'health status' feature allows users to quickly tag themselves towards an indicative health condition by answering a few questions. The basis behind this feature lies in whether a person has COVID-19 symptoms or not. For having any COVID-19 symptom, the person will be diagnosed with a set of 1-6 "Yes/No" questions to be tagged as either of 'Infected', 'Suspected', 'Symptomatic', 'quarantined' and 'Cured' following the logic demonstrated in the Fig. 5. The person will be tagged as "Asymptomatic or Low chance of COVID" if no symptom is observed. Fig. 6 shows some of the screenshots of this feature. In such a case, the person may traverse through 'self-assessment' features (discussed in the next section) in-depth assessment for chances of getting infected by COVID-19.

### 3.4. Self-assessment

A person can take self-assessment test before opting for a physical medical check-up if that person is not sure about own health status or

unable to determine the same in absence of any COVID-19 symptom. This self-assessment test comprises a series of questions composed of a set of parameters, and their corresponding attributes (Biswas et al., 2021a), found from the available literature and guidelines provided by World Health Organizations (WHO) and the Center for Disease Control (CDC). An expert opinion survey was conducted with a structured questionnaire, to determine the weight distribution among the components (parameters and attributes) of the self-assessment test, so that the system can generate a percentage or chance of getting infected by COVID-19 once a person takes the self-assessment test and provide inputs or answers to it. Here, the experts are the general physicians or specialist doctors who are practicing mostly pulmonary or respiratory illness since long time, and actively looking after COVID-19 patients. The components of the self-assessment test comprising 12 parameters and their corresponding attributes are shown in the Fig. 7.

### 3.5. Map-enabled infection tracker

This section discusses an important set of features for users to easily visualize complex data on Map. The working of this feature is demonstrated in Fig. 8.

A user can view the state-level or a district level choropleth map (Fig. 9) showing COVID statistics fetched from the server. The user can switch to a "hotspot map" to visualize if there is any hotspot in the place nearby. A search with autocomplete option is provided so that a user can search a place by its name and view the hotspots there. This has been done using Google Maps API for Places Autocomplete (Google, 2020a). Another feature allows users to commute from a source location to destination location choosing the path with the least number of hotspots. This is done using Google Maps API for roads, directions, and distance matrix (Google, 2020d, 2020b, 2020c) and experience is similar to using Google Maps.

## 4. Evaluation

### 4.1. Stress testing

Stress test on the API was performed using Apache Jmeter (Halili, 2008). The test server configuration consisted of a core i5 processor, 2GB RAM running ubuntu server 20.04 with python application

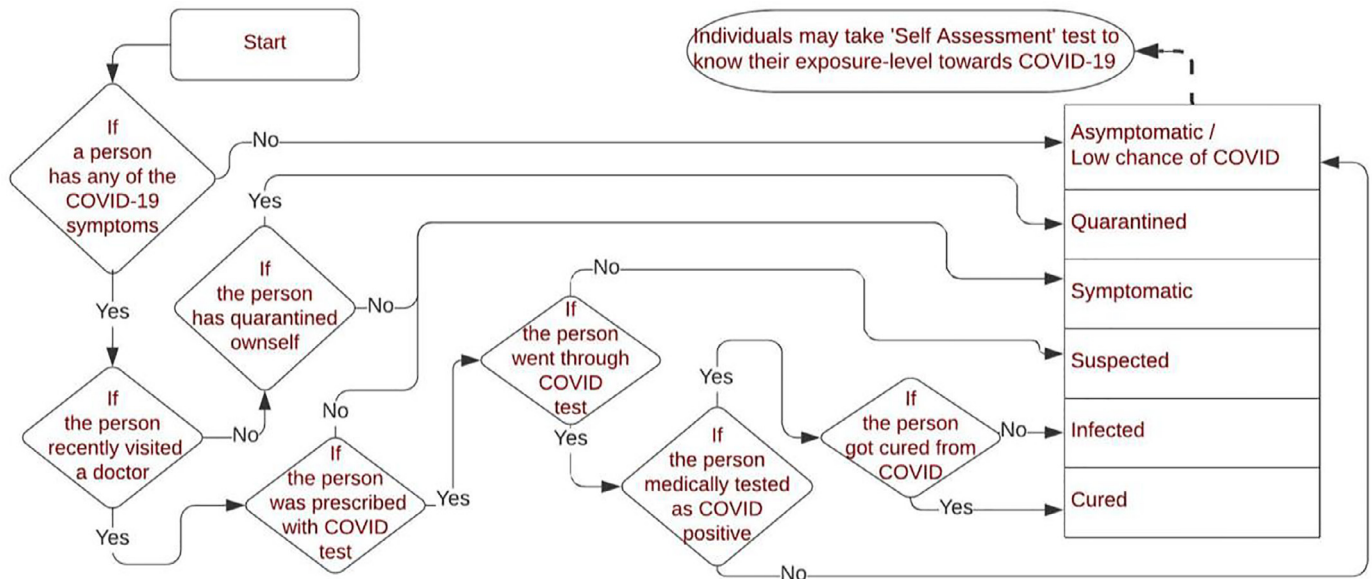


Fig. 5. Working algorithm for health status.

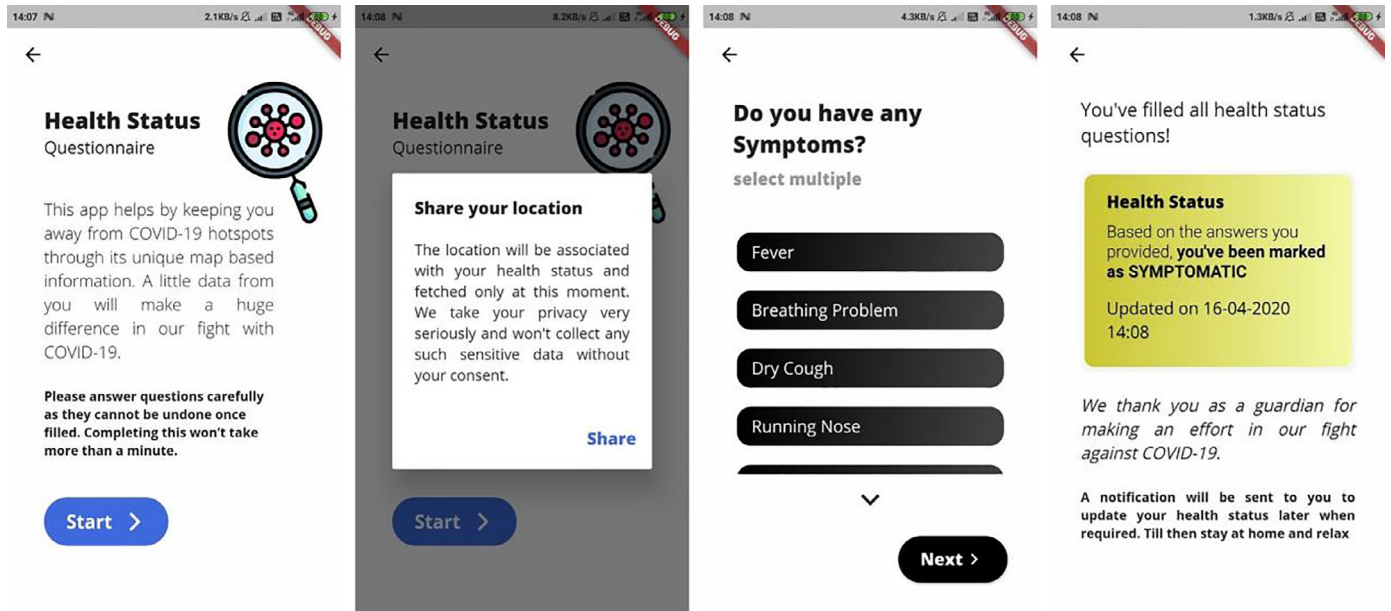


Fig. 6. Sample screens of health status.

served through gunicorn running 8 workers. The test was performed under 2 conditions, one while using cache (Memcached) and one without cache. The following observations were recorded on GET requests on the me/ endpoint which is the most frequented.

From the observations mentioned in the Table 2, it is inferred that writing a custom serializer with ReadOnlyField provided average response time gains of 227% compared to ModelSerializer. In the test, it is also inferred that using a cold cache improved average response time by 683% compared to not using a cache

4.2. Usability testing

4.2.1. Testing procedure

System Usability Scale (SUS) is one of the highly reliable, robust, and versatile tools used to measure users' satisfaction with system usability (Brooke, 1996). This is also being used in testing mobile apps

(Biswas et al., 2021b). The SUS is adopted for our study as part of the evaluation, where users or participants are asked to rate 10 statements (five positive and five negative) on a 5-point Likert Scale, after performing certain tasks on the system or mobile application. Literature suggests to have at least 20 users or participants to identify 95% of the usability (Faulkner, 2003). In this study, a total of 41 evaluators took part. They were asked to perform certain tasks in the application such as registration, health-status reporting, map-visualization, etc., before they provide responses for SUS statements. Finally, the usability-score is estimated from the recorded responses following the SUS calculation rules.

4.2.2. Usability analysis

The overall rating provided by the evaluators, along with the SUS score estimated from their recorded response at the individual level, are demonstrated in the Fig. 10. The aggregated SUS score for our mobile

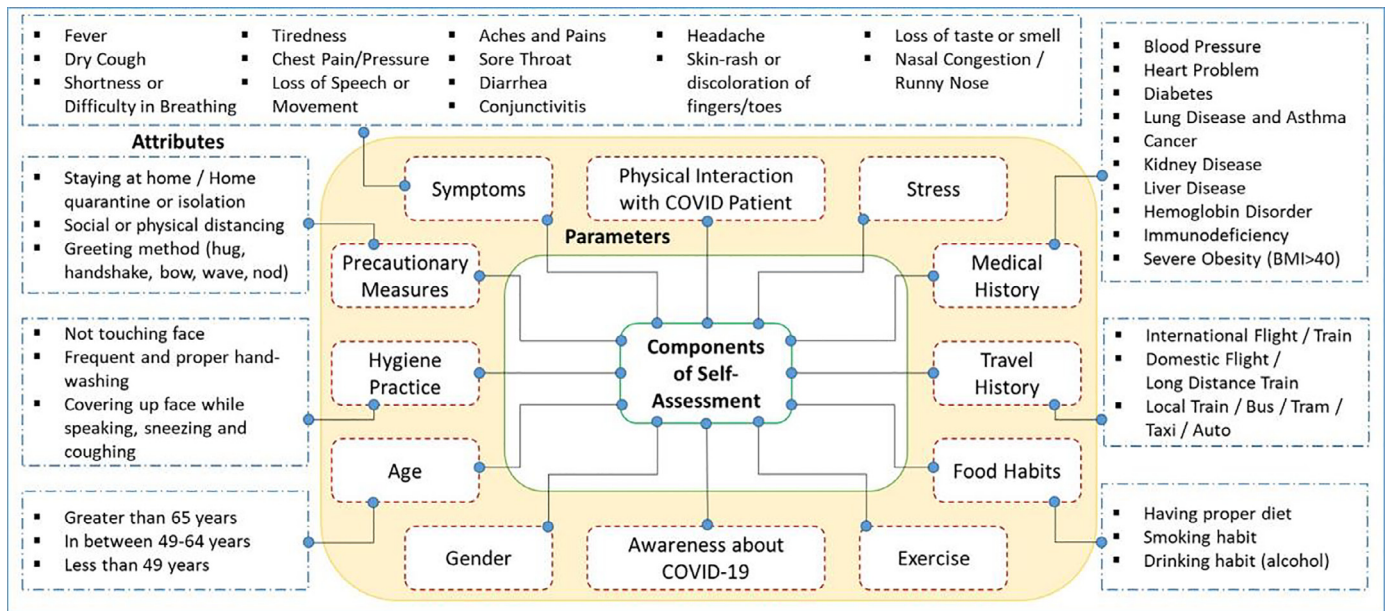


Fig. 7. Components of self-assessment- parameters and attributes.

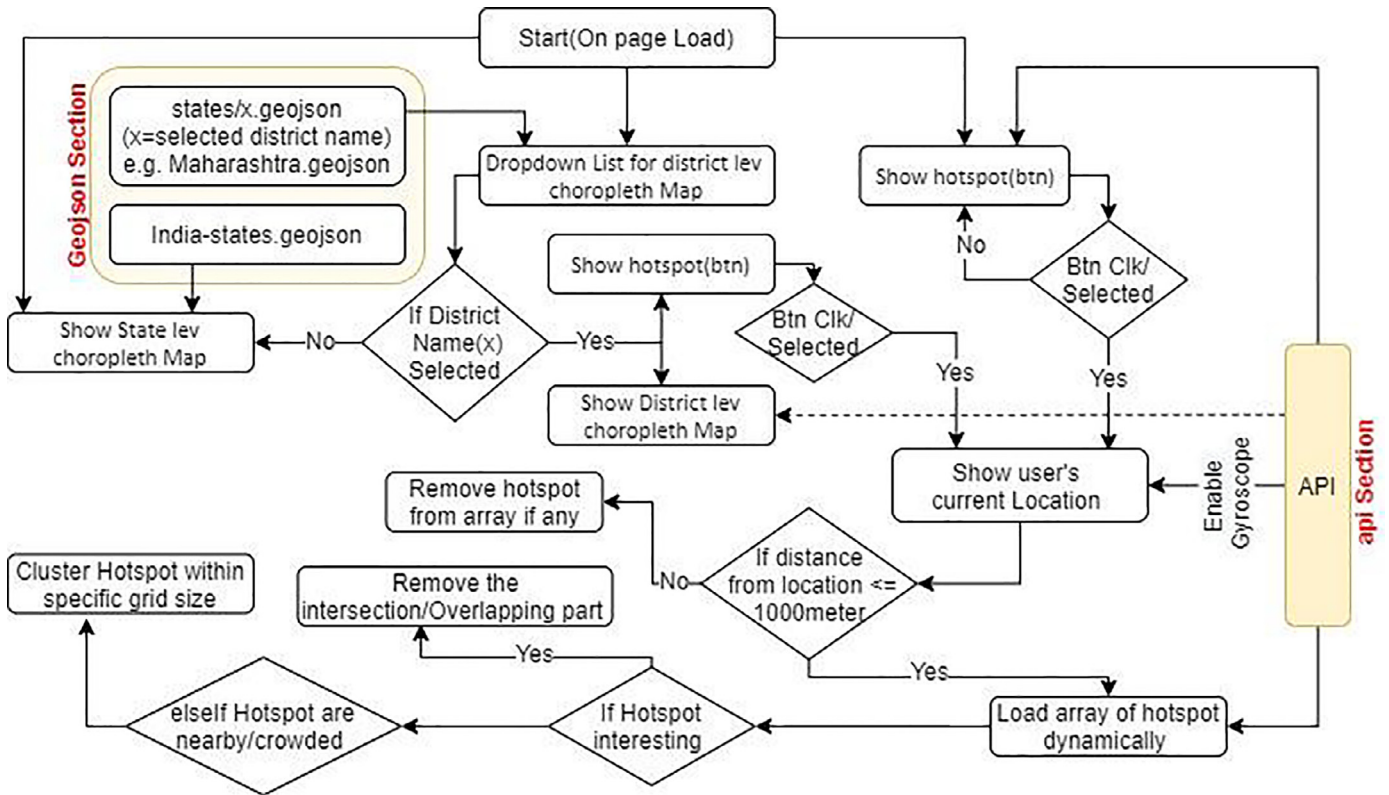


Fig. 8. Working of infection tracker.

Table 2  
Outcome of stress test.

	Custom Serializer with read-only fields		ModelSerializer in Django Rest Framework		
	Without Using Cache		Using Cache	Using Cache	
Number of Users	1000		1000	20000	20000
Ramp-up Time	10		10	10	10
Average Requests	99.5		99.6	749.7	717.8
Response Time (ms)	Average		82	12	203
	Minimum		5	3	2
	Maximum		657	165	1079
Errors	0		0	0	0

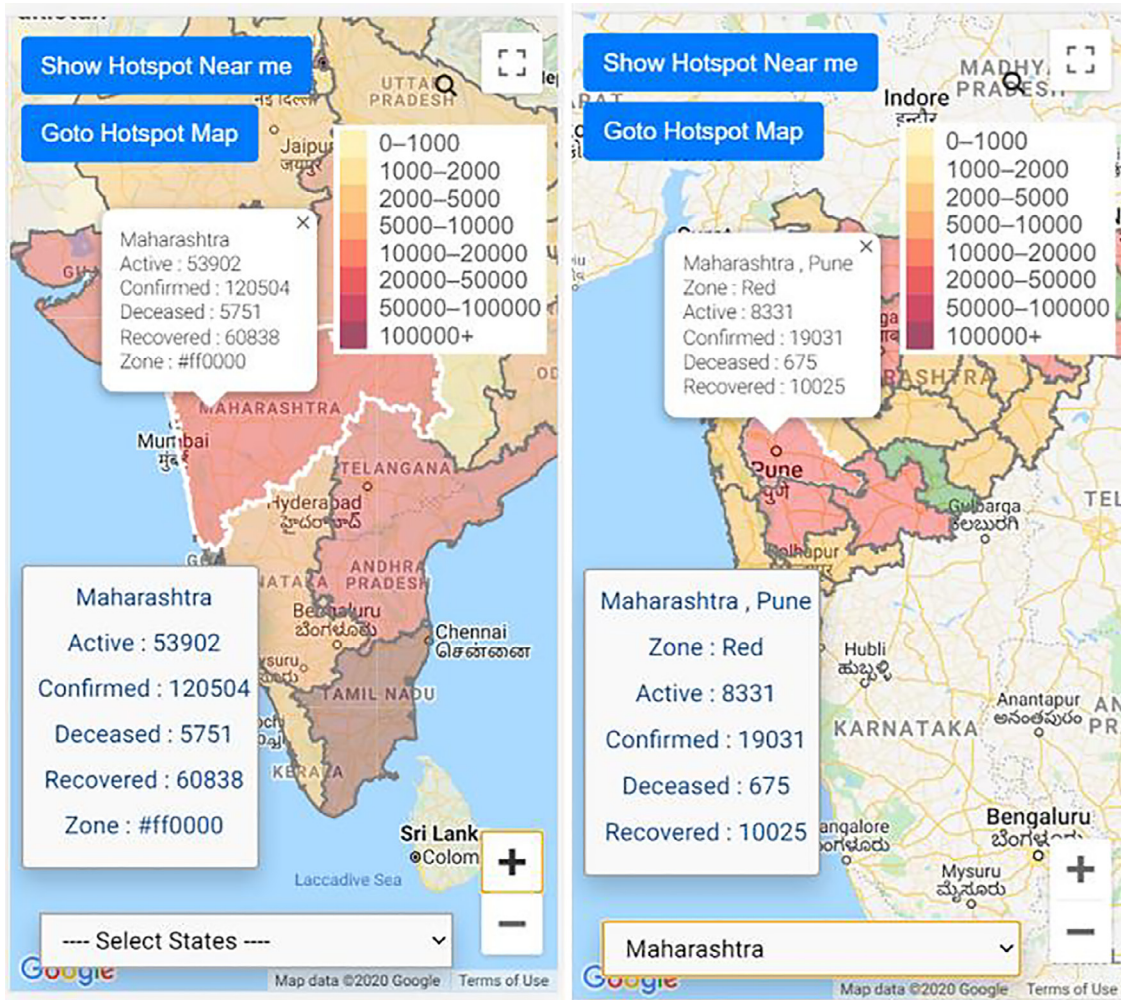


Fig. 9. Choropleth map at the state and district level.

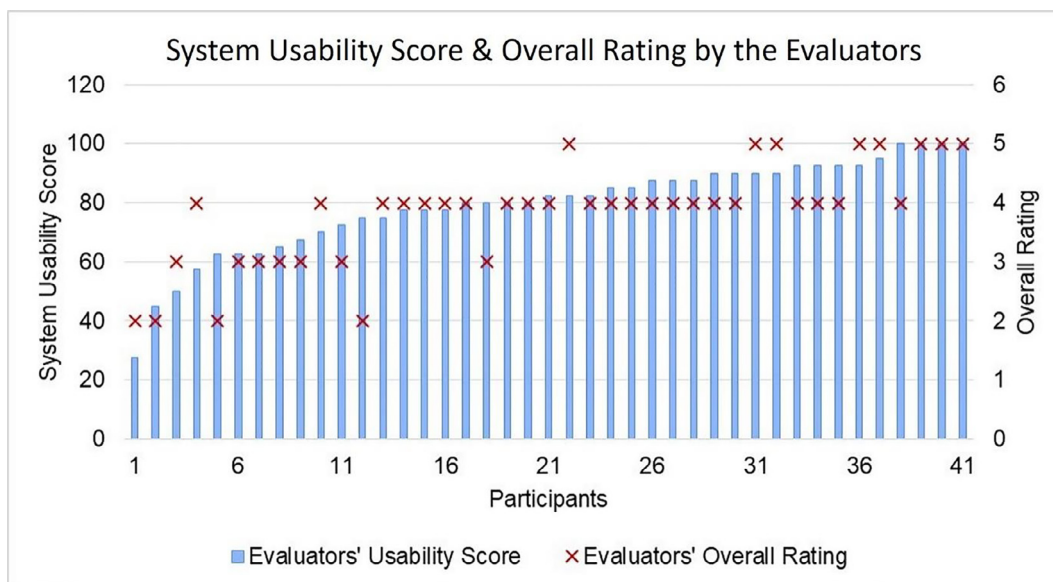


Fig. 10. System usability score in comparison with overall rating.



application is 79.26 on a scale of 100. As per the literature, our application can be classified as 'Good' as per (Bangor et al., 2009) and 'Grade B' as per (Sauro, 2011). This testing asserts that the application is useful and ready to use.

## 5. Discussion

### 5.1. Comprehensiveness, usefulness & applicability

The application 'Rakshak' is developed not only to provide the COVID-19 advisories and guidelines but also to embed the necessary key features in one place that makes it comprehensive- (1) Contact logger & tracer, (2) Self-assessment, and (3) Infection tracker. 'Contact logger & tracer' helps to get the complete list of the identity of the people without a moment of delay, who came in contact with an infected person in the last 14 days. Users add details of the contacted user in the app in multiple ways. The app then sends a request to the contacted user via notification/sms for verification. On verification, the contacts in this list are notified if the user reports themselves as positive in the 'health status' questionnaire. 'Self-assessment' helps individuals to know their chances of getting COVID-19 from their self-reported responses. These responses are collected against a set of questions consisting of a number of criteria and their corresponding attributes, composed from the available scientific literature and recommendations by the national and international authorities. 'Infection tracker' using google map API helps people to track infected-zones nearby so that they may opt for an alternate route while traveling towards the destination, just by keeping an eye on a map-based visualization-tool. While the stress-test demonstrates the system's capability of handling user requests, the usability-test further confirms that the application is easy-to-use.

The proposed system is not capable to combat any generic communicable diseases. It only aims to fight the communicable diseases, which have similar kind of properties like COVID-19, such as common cold, sore throat, etc. Health assessment, contact tracing and map-enabled infection tracking are among the key instruments to fight communicable diseases like COVID-19 at large scale. The present study demonstrated how those instruments can be bind together to make a comprehensive app fighting other communicable diseases like COVID-19. However, certain specific requirements need to be especially articulated for particular disease. For example, common cold and sore throat are communicable diseases, which properties are somewhat similar to COVID-19. If the application needs to be modified for common cold, then contact-tracing for the last 14-days might not be required.

### 5.2. Practical implications

Due to the nation-specific privacy and google restriction policies, many of the resource-constrained countries are unable to get the benefit of having an easy-to-use comprehensive mobile-application that can help in managing communicable diseases like COVID-19. The present study explores how to build a comprehensive and easy-to-use application for mobile health service delivery especially fabricated to fight communicable diseases. This will help the needy authorities from those resource-constrained nations to develop an app using open-source resources. This dynamic nature of the app is the main advantage. As the Covid-19 is a novel virus, attributes, parameters may change over time across nations, the authorities should be flexible enough to add or modify those details according to the need. In this way, the authorities can provide a human-centric technological solution to their marginalized people. The people will have a handy tool with which they can find their chances of getting infected and point out if there is any infected area on their travel path so that they can avoid/re-route. The application can also help an infected person to warn those who came in contact in the last four days.

### 5.3. Privacy & security

The application gives extreme importance to user privacy and security. The application uses HTTPS protocol to communicate with the backend which in turn enforces HSTS (Hodges et al., 2012) for an added layer of security. The application provides One Time Password (OTP) based login with focus on minimizing the attack vectors by invalidating the OTPs based on time and by enforcing limits on API calls. The application also minimizes the risk of remote code execution and injection by sanitizing all user queries. The application also enforces object-level access permissions (Armando et al., 2014) across the system to mitigate the risk of Broken Access Control (OWASP, 2017). The contact logger functionality provides a feature to anonymously add contacts with a QR code which is encrypted with AES 256 encryption to protect user identity. The user location history is stored without any reference to the user by using an SHA 256 hash which is used to verify a user's claim to update the location history object in the future. The system enforces a strict tamper resist mechanism on the questionnaire to prevent users from arbitrarily updating their health status. This is done by providing a single endpoint to fetch and update answers, thereby minimizing the attack surface.

## 6. Conclusion & future scope

After the emergence of the novel Coronavirus, several national and international authorities launched various mobile applications as one of the measures to fight the COVID-19. While COVID-19 advisories and guidelines are the most common services that are taken into consideration, many of these available and accessible apps are not comprehensive enough in absence of some key features. In this study, a comprehensive and easy-to-use application is conceptualized and developed towards mobile health service delivery, especially fabricated to fight communicable diseases like COVID-19. It comprises of three major functional capabilities- 'contact logger & tracer' to get the list of contacts for the last 14 days of a very new COVID-patient, scientific rational based 'self-assessment' to know the chances of getting COVID-19 at the individual level, and 'infection tracker' to visualize infected zone for alternate route-planning. Other than its functional capabilities, the stress test and usability test further affirm that the app is useful and ready-to-use. This will assist the managing authorities of resource-strapped nations in providing a human-centric technical solution to their marginalized citizens to fight against communicable diseases. Further studies on exploring other key features and post-implementation impact can be carried out to add value to this study.

### Funding sources

The study presented here is a part of 'Ministry of Human Resource Development (MHRD), Govt. of India' funded project, under National Mission on Education through Information and Communication Technology (NMEICT).

### Funding statement

Details are omitted due to double-blind peer-review process.

### Declaration of Competing Interests

The authors report no conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to thank all the medical experts and physicians for their active participation in the survey and providing valuable feedback. They also thank all the members of “e-Yantra” for their support.

## References

- Armando, A., Carbone, R., Chekole, E.G., Ranise, S. 2014. “Attribute based access control for apis in spring security”. In: *Proceedings of the 19th ACM symposium on access control models and technologies*, 85–88.
- Bangor, A., Kortum, P., Miller, J. 2009. “Determining what individual SUS scores mean: Adding an adjective rating scale”. *Journal of Usability Studies* 4 (3): 114–123.
- Battineni, G., Chintalapudi, N., Amenta, F. 2020. “Forecasting of COVID-19 epidemic size in four high hitting nations (USA, Brazil, India and Russia) by Fb-Prophet machine learning model”. *Applied Computing and Informatics* doi:10.1108/ACI-09-2020-0059.
- Boukhary, S., Colmenares, E. 2019. “A clean approach to flutter development through the flutter clean architecture package”. In: *2019 International conference on computational science and computational intelligence (CSCI)*, 1115–1120.
- Brooke, J. 1996. “SUS-A quick and dirty usability scale”. *Usability Evaluation in Industry* 189 (194): 4–7.
- Brownstein, J.S., Freifeld, C.C., Madoff, L.C. 2009. “Digital disease detection—Harnessing the web for public health surveillance”. *The New England Journal of Medicine* 360 (21): 2153.
- Burkhart, T. (2021). Get it - Simple service locator for dart and flutter projects. [https://github.com/fluttercommunity/get\\_it](https://github.com/fluttercommunity/get_it)
- Calvo, R.A., Deterding, S., Ryan, R.M. 2020. *Health surveillance during covid-19 pandemic*: British Medical Journal Publishing Group. BMJ 2020; 369. doi: 10.1136/bmj.m1373.
- CDC. (2021). Symptoms of COVID-19. CDC Website. <https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html>
- Cho, H., Ippolito, D., & Yu, Y.W. (2020). Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs. ArXiv Preprint ArXiv:2003.11511.
- Drake, J.D., & Worsley, J.C. (2002). Practical PostgreSQL. “O’Reilly Media, Inc.”
- Faulkner, L. 2003. “Beyond the five-user assumption: Benefits of increased sample sizes in usability testing”. *Behavior Research Methods, Instruments, & Computers* 35 (3): 379–383.
- Fielding, R.T., Taylor, R.N. 2002. “Principled design of the modern web architecture”. *ACM Transactions on Internet Technology (TOIT)* 2 (2): 115–150.
- Forcier, J., Bissex, P., & Chun, W.J. (2008). Python web development with Django. Addison-Wesley Professional.
- Google. (2020a). Google Maps platform - Web - Maps JavaScript API - Places autocomplete. <https://developers.google.com/maps/documentation/javascript/places-autocomplete>
- Google. (2020b). Google maps platform - Web services - Directions API. <https://developers.google.com/maps/documentation/directions/overview>
- Google. (2020c). Google maps platform - Web services - Distance matrix API. <https://developers.google.com/maps/documentation/distance-matrix/overview>
- Google. (2020d). Google maps platform - Web services - Roads API. <https://developers.google.com/maps/documentation/roads/overview>
- Grind, K., McMillan, R., Wilde Mathews, A. 2020. “To track virus, governments weigh surveillance tools that push privacy limits”. *Wall Street Journal*. <https://www.wsj.com/articles/to-track-virus-governments-weigh-surveillance-tools-that-push-privacy-limits-11584479841>.
- Halili, E.H. 2008. *Apache JMeter: A practical beginner’s guide to automated testing and performance measurement for your websites*: Packt Publishing Ltd.
- Hodges, J., Jackson, C., & Barth, A. (2012). Http strict transport security (hsts). URL: <http://Tools.IETF.org/Html/Draft-IETF-Websec-Strict-Transport-Sec-04>.
- Holovaty, A., & Kaplan-Moss, J. (2009). The definitive guide to Django: Web development done right. Apress.
- Mayor, S. 2020. *Covid-19: Researchers launch app to track spread of symptoms in the UK*: British Medical Journal Publishing Group.
- Moroney, L. 2017. “Firebase cloud messaging”. In: *The definitive guide to firebase*, 163–188: Springer.
- Moroney, Moroney, L, Anglin. 2017. *Definitive guide to firebase* Springer <https://doi.org/10.1007/978-1-4842-2943-9>.
- Mozur, P., Zhong, R., & Krolik, A. (2020, March 1). In coronavirus fight, China gives citizens a color code, with red flags. The New York Times. <https://www.nytimes.com/2020/03/01/business/china-coronavirus-surveillance.html>
- Napoli, M.L. 2019. *Beginning flutter: A hands on guide to app development*: John Wiley & Sons.
- Nishtala, R., Fugal, H., Grimm, S., Kwiatkowski, M., Lee, H., Li, H.C., McElroy, R., Paleczny, M., Peek, D., Saab, P. 2013. “Scaling memcache at facebook”. In: *10th {USENIX} symposium on networked systems design and implementation*, 385–398 ({NSDI} 13).
- Olsson, M. (2020). A comparison of performance and looks between flutter and native applications: When to prefer flutter over native in mobile application development.
- OWASP. (2017). OWASP top 10 - 2017: The ten most critical web application security risks. [https://wiki.owasp.org/images/7/72/OWASP\\_Top\\_10-2017\\_%28en%29.pdf.pdf](https://wiki.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf)
- Owoh, N.P., Singh, M.M. 2020. “Security analysis of mobile crowd sensing applications”. *Applied Computing and Informatics* 18 (1/2): 2–21. doi:10.1016/j.aci.2018.10.002.
- Prasanna, D.R. (2009). Dependency injection. manning.
- Sauro, J. 2011. *A practical guide to the system usability scale: Background, benchmarks & best practices*, Denver, CO: Measuring Usability LLC.
- Strobl, C. (2008). PostGIS.
- Winer, S. 2020. “Health ministry launches phone app to help prevent spread of coronavirus”. *Times of Israel*. <https://www.timesofisrael.com/health-ministry-launches-phone-app-to-help-prevent-spread-of-coronavirus/>.
- Yusif, S., Hafeez-Baig, A., Soar, J. 2020. “Change management and adoption of health information technology (HIT)/eHealth in public hospitals in Ghana: A qualitative study”. *Applied Computing and Informatics* doi:10.1016/j.aci.2019.11.007.
- Biswas, R., Arya, S., Arya, K. 2021a. “A quantifiable framework for ‘Covid-19 exposure’ to support the vaccine prioritization and resource allocation for resource-constrained societies”. *Urban Governance* 1 (1): 23–29. doi:10.1016/j.ugj.2021.11.001.
- Biswas, R., Arya, K., Fernandes, V., Shah, T. 2021b. “Find A Loo: an app for sanitation governance”. *Information, Communication & Society* 24 (11): 1586–1602. doi:10.1080/1369118X.2020.1716038.