Data-Driven Mobile Applications Based on AppSheet as Support in COVID-19 Crisis

Nenad Petrović, Maša Radenković and Valentina Nejković

***Abstract*—In this paper, it is explored how data-driven mobile applications can contribute to battle against the COVID-19 pandemic. As proof of concept, two novel case studies are presented: indoor safety monitoring and resource planning during pandemic crisis. AppSheet platform for automated development based on Google Sheets as data source was used for implementation of corresponding multiplatform mobile applications.**

***Index Terms*—AppSheet; coronavirus; Google Sheets; mobile applications.**

I. Introduction

Since the beginning of this year, the world tackles the novel, flu-alike and highly infectious respiratory disease *COVID-19* caused by *SARS-Cov-2* virus (often referred to as *coronavirus*) [1]. It was discovered in Wuhan, China, but reached the other parts of the world in few weeks afterwards.

The most common symptoms of this disease are: fever, short breath, dry cough, fatigue, loss of taste and smell [2]. It is transmitted both directly through respiratory droplets and indirectly, via surfaces [3]. Incubation period varies from case to case, but could be quite long [4]. However, even asymptomatic persons can spread the disease [5], while death rate is quite high when it comes to older population and persons with chronic diseases [6], which makes the situation almost impossible to handle. The adoption of face masks and hand sanitizers has shown promising results in case of disease spread reduction [7]. However, the main issue is the lack of approved vaccine and medication [8].

Due to all the mentioned facts, many protection and safety measures were taken by governments in order to reduce the disease spread, such as obligatory indoor mask wearing, social distancing, quarantine, self-isolation, working from home, limiting citizens' movement and tourism, prohibition and cancellation of public events and gatherings [9]. Apart the huge number of lost lives, the pandemic has also caused catastrophic financial losses and stagnation of economy [10], leading towards overall global crisis.

When it comes to the global battle against COVID-19 and reduction of pandemic crisis consequences, it was concluded that state of the art technologies: artificial intelligence, blockchain, IoT and smartphone applications have huge potential. The main role of IoT devices is to collect sensor data about persons with goal to provide early diagnosis, enforce safety or guideline rules or help to patients and persons in isolation [11, 12]. On the other side, artificial intelligence algorithms enable extract meaningful knowledge and patterns from the collected sensor data to enable predictions of future trends or detect certain state/behavior by classifying observations [13]. Moreover, blockchain can be used to reduce the spread of misinformation during pandemics and trusted patient number tracking by ensuring authenticity or even transaction execution between the involved parties for purpose of resource exchange [14, 15, 16]. Finally, mobile applications offer interface to the services provided by previous enabling technologies for timely adoption among the targeted user groups [17, 18].

In this paper, it is explored how AppSheet[1] platform can be used for rapid mobile application development in order to support the society during COVID-19 pandemics. As outcome, two case studies making use of the previous enabler technologies are presented: 1) indoor safety monitoring app for security guards based on computer vision-enabled IoT solution 2) resource planning app for city representatives relying on AI-based predictions and blockchain for resource exchange.

II. Background and Related Work

A. AppSheet

AppSheet is an online development platform which enables easy creation and distribution of mobile, tablet and web applications starting from cloud data sources, such as spreadsheets and databases without any coding. Its first version was released as independent project in 2014, while it was acquired by Google and became part of Google Cloud[2] in January 2020. It mainly aims business use cases, such as project management, customer relationship management and personalized reporting. AppSheet analyzes the structure of provided data sources and automatically generates the views within the application. However, the user is also able to customize the generated user interface by showing or hiding particular table columns, grouping or sorting data, creating additional views and inserting computed values with respect to defined formulas. Apart from widely used textual and numeric types, it gives the ability to embed the images and videos within the user interface as well. AppSheet is free to use for up to 10 beta/test users, while monthly fee has to be paid for commercial usage and larger user base. When it

Nenad Petrović, Maša Radenković and Valentina Nejković are with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mails: nenad.petrovic@elfak.ni.ac.rs, masa.radenkovic@elfak.rs, valentina.nejkovic@elfak.ni.ac.rs).

[1] https://www.appsheet.com/
[2] https://cloud.google.com/appsheet

comes to more sophisticated features, the free version of AppSheet supports embedding value prediction and optical character recognition (OCR), while premium features include sentiment analysis, anomaly detection, data clustering and many other commonly used machine learning techniques. There are some limitations, such as minimal number of data rows needed for application creation (which is 4) and prediction training (25 observations). In order to run software created using AppSheet on mobile devices (Android or iOS), it is necessary to download and install the client application from corresponding online store which serves as container for user-created applications. However, active internet connection for their usage is required. In Fig. 1, an overview of typical application creation workflow (starting from Google Sheets[3] as data source) using AppSheet is given.

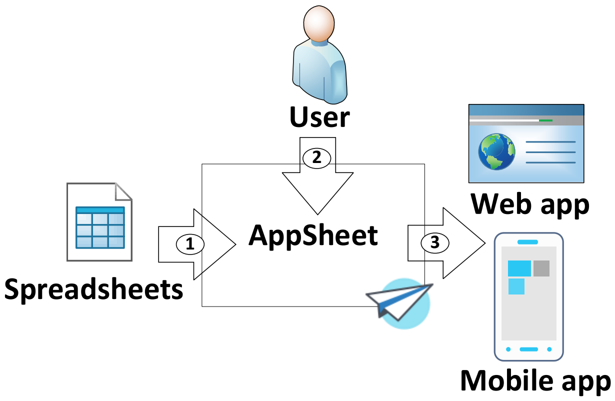

Fig. 1. Overview of typical AppSheet workflow: 1-Import data sources 2-User customizations 3-Generated output.

AppSheet-based mobile applications have been approved as effective solution across many industrial areas so far. In [19], AppSheet was used for real-time online monitoring of on-shelf availability in case of fast-moving consumer goods (FMCG). On the other side, in [20], it was used for water management application that calculates plant water consumption within given area. When it comes to healthcare, in [21], AppSheet was leveraged for development of electronic record keeping system at a pediatric clinic.

In this paper, AppSheet is adopted for COVID-19 support mobile application development, as it enables rapid prototyping and delivery of data-driven mobile applications, which is crucial in pandemic situation. As unexpected changes (such as dramatic rise of new disease cases) might occur quickly, it is of utmost importance to provide support services and respond to them timely in order to reduce the possible consequences [18]. Moreover, another advantage of AppSheet is multiplatform support without additional efforts and modifications. Finally, cross platform mobile development frameworks such as Xamarin require solid programming skills, while AppSheet enables easy application creation by domain experts.

B. *Mobile support applications in COVID-19 crisis*

In [22], a mobile application that allows people who are in self-isolation to ask for help and enable a fast, secure support by the community within 15 miles radius was presented. It was implemented using Flutter framework and supports both Android and iOS. On the other side, the AI-enabled mobile application presented in [23] aims to reduce the spread of COVID-19 disease by integrating the capabilities of user tracking, notification of the infected persons' employment/education institution and spread prediction based on the recorded movement locations. Furthermore, the work presented in [24] presents the computer vision approach which detects if face mask is worn properly using the image taken by smartphone camera. In [25], a mobile app for preliminary COVID-19 diagnosis based on cough samples was shown. In [26], an approach for on-device COVID-19 screening using snapshots of chest X-ray targeting smartphones was introduced. Finally, [27] discusses how mobile applications leveraging AI and AR can be used to measure athlete's performance without any physical contact with coach during the pandemic crisis. In Table I, overview of relevant COVID-19 support mobile applications with respect to their use cases is provided.

TABLE I
COVID-19 SUPPORT MOBILE APPLICATIONS

Application	**Use case**
C-19 C	Help in self-isolation
BreakTheChain	Spread prediction
CheckYourMask	Mask detection/check
AI4COVID-19	Cough-based diagnosis
COVID-MobileXpert	Chest X-ray screening

However, in this paper, the focus is on COVID-19 pandemic-related use cases not covered by the existing literature, mainly resource planning and indoor safety monitoring, relying on the work from [16] and [28].

III. CASE STUDIES

A. *Indoor COVID-19 Safety Monitoring Security Guard App*

During the COVID-19 pandemic, it is very important for institutions and organizations to respect the COVID-19 safety rules and guidelines in order to reduce the disease spread, especially indoors [29].

This case study builds upon the work from [28], making use of IoT-based computer vision system that consists of Raspberry Pi devices equipped with camera. Each Raspberry Pi is located at a different location inside a building (main entrance or specific room entrance, for example) and is used to observe whether the passengers respect COVID-19 indoor safety guidelines: wearing masks and social distancing. For that purpose, computer vision algorithms are performed on-board in order to process the images coming from camera stream. In case that violation of safety rules is detected, then the Google Sheets API for Python[4] is used to append new data

[3] https://www.google.com/sheets/about/

[4] https://developers.google.com/sheets/api/quickstart/python

to the corresponding spreadsheet used as a log of rule violations at different locations within the building. Moreover, the camera image where violation was detected is uploaded to server and its URL is also included within the log together with the timestamp of current time. Apart from that, the corresponding security guard for a given location is determined using the remote call to the edge server and procedure described in [28]. This information is also included in the spreadsheet log, so the security guards can be notified which of the reported rule violations are under their responsibility. In order to keep track of the violations, security guards use mobile application developed using AppSheet, based on the data available within the violation log spreadsheet. In Fig. 2, overview of this case study is given.

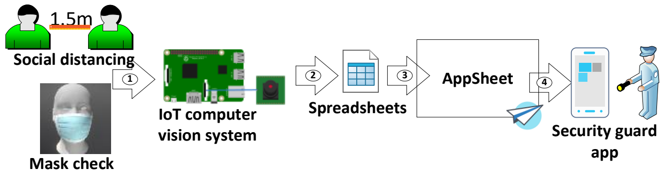

Fig. 2. COVID-19 indoor safety mobile application overview: 1-Camera stream analysis 2-Insert rule violation record 3-Update app 4-Notify security guard.

In Listing 1, the role of IoT system in context of security guard mobile application for violation log creation is described as pseudocode.

```
Input: image, sheet_id, location
Output: violations
Steps:
distance_violation:=SocialDistanceCheck(image);
mask_violation:=MaskCheck(image);
if(distance_violation==false and mask_violation==false)
    violations:="No violation";
else
    if(mask_violation==true)
        violations:=violations+"No mask or worn improperly";
    if(distance_violation==true)
        violations:=violations+"Social distancing not satisfied";
    image_url:=UploadImage(image);
    guard_id:=RemoteCall.GetFreeGuard(location);
    t:=Now();
    WriteToSheet(sheet_id, image_url, violations, location, t, guard_id);
end if;
return violations;
end.
```

Listing 1. IoT indoor safety monitoring system modifying log using Google Sheets API in pseudocode.

In Fig. 3, the use case diagram of this application is shown.

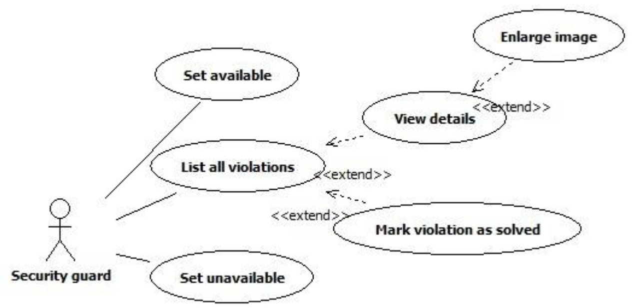

Fig. 3. Indoor safety security guard App use cases.

In Fig. 4, screenshots of security guard AppSheet mobile app run on iPhone SE is given. The first view (a) shows the list of detected violations within the building, while the second one (b) shows details for the selected rule violation. For each violation, it is possible to see the enlarged image, time/date when it occurred together with the location where it was detected and which device performed its detection. Once security guard warns the persons breaking the safety rules, the violation event can be checked as resolved and removed from pending list. Moreover, security guards can set their availability status to determine whether they will be taken as candidates by the server's algorithm [28] for patrol task allocation or no.

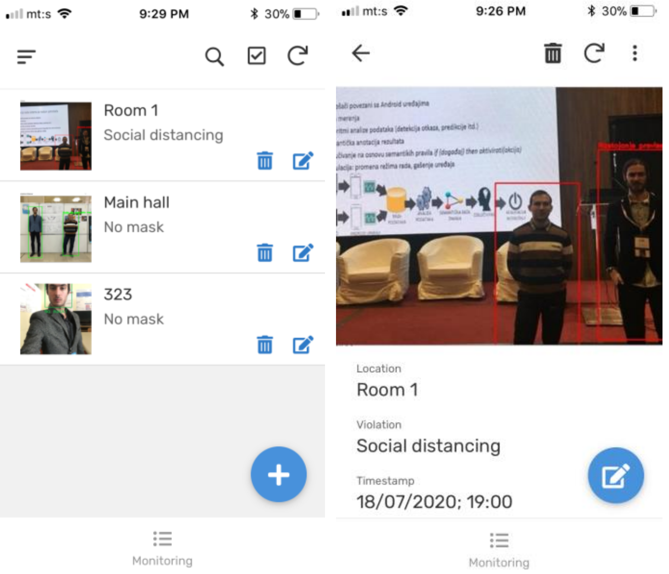

Fig. 4. Indoor safety security guard AppSheet iOS mobile app screenshots: (a) list of detected violations (b) detailed violation view.

In Table II, the structure of the underlying Google Sheet used for AppSheet application generation is illustrated.

TABLE II
INDOOR SAFETY APPLICATION GOOGLE SHEET STRUCTURE

Location	*Device*	*Violation*	*T*	*Guard id*	*Image url*	*Solved*

B. *Resource Planning Mobile Application*

In pandemic, it is of utmost importance to plan resources efficiently, but timely – either human, material resources or equipment. The protection of health and the economy of a country are linked tightly [30].

This mobile app follows up the framework for efficient city resource planning during pandemic crisis from [16]. Its target users are city administration/financial representatives. For each city, relevant resources in context of pandemic crisis are considered [16]: budget, available hospital places, test kits, face masks, ambulance vehicles and doctors. Moreover, it is possible to see the detailed information about availability of resources and their prices.

For each day, the number of predicted new COVID-19 cases and deaths are taken into account in order to determine whether the available city resources are sufficient to satisfy the current demands. For daily case predictions, several factors are considered, such as historical data about cases in the considered city, weather parameters, safety measures applied within the city (lockdown or limited movement, gathering prohibitions, restaurant/bar working hours). The first two parameters are automatically updated daily, leveraging the publicly available online data from services, such as [31], [32] and [33], while the others are entered manually by users. Moreover, the new case predictions are further used within the formulas for resource prediction as described in [16]. The prediction mechanism completely relies on AppSheet whose minimum number of observations for training is 25. Moreover, calendar view of daily predictions is also generated using AppSheet. Finally, city representatives can make offers to other cities for resource exchange. If offer is accepted, resource exchange between the cities relying on blockchain and smart contracts will be performed building upon the work presented in [16] and [34]. In Fig. 5, an overview of described applications is given.

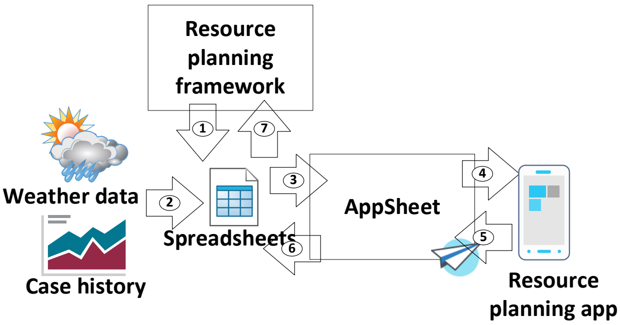

Fig. 5. COVID-19 resource planning mobile app: 1-City resource model 2-Online weather data and case history 3-Import spreadsheet 4-Generated data view 5-User input parameter 6-Predictions 7-Import spreadsheet into resource planning framework.

In Fig. 6, an overview of resource planning app use cases is given.

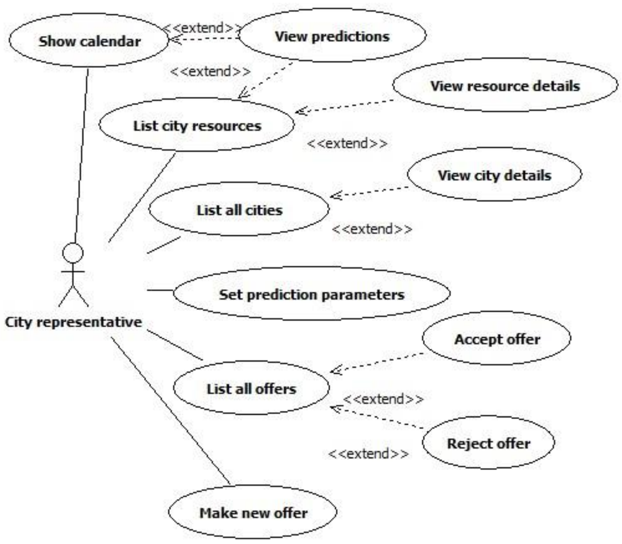

Fig. 6. City representative resource planning mobile app use case diagram.

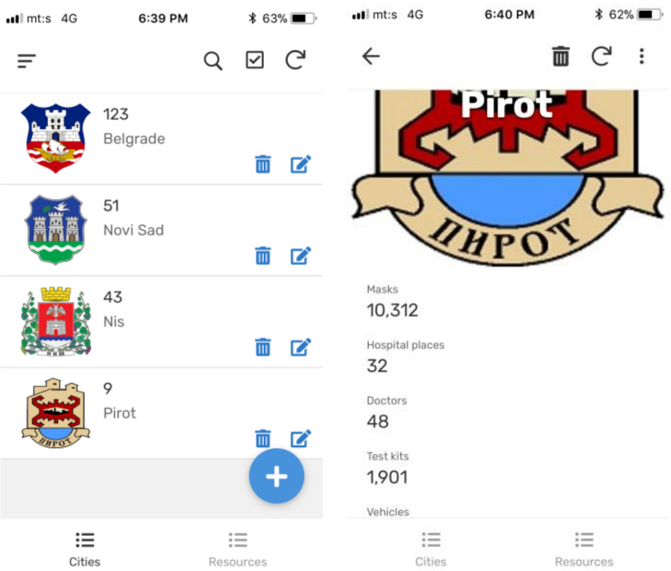

Fig. 7. Resource planning AppSheet iOS mobile app screenshots: (a) city overview (b) city resource view.

Fig. 7. shows a screenshot of city resource views taken on iPhone SE. The first screen (a) shows the list of available cities together with number of new cases that day. The second screen (b) shows summary of available resources for each of the cities. Furthermore, Fig. 8 displays the details the list of available resource types and details for each of them (b). In this case, the number of available places in hospitals for the selected city is shown.

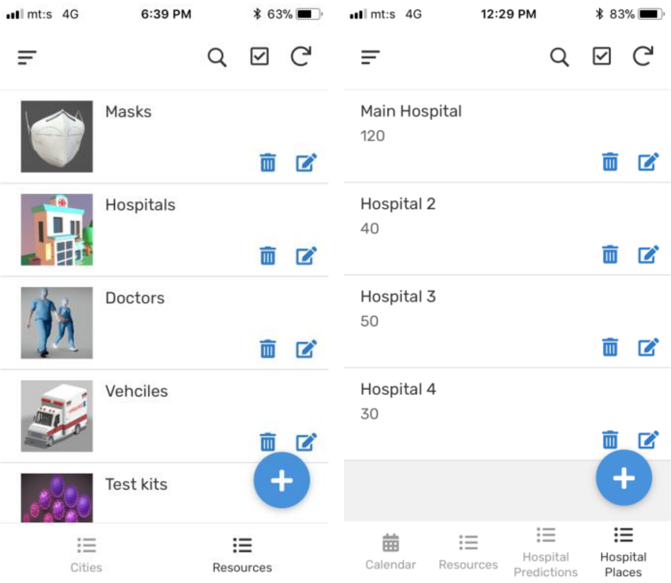

Fig. 8. Resource planning AppSheet iOS mobile app resource view screenshots: (a) resource list (b) resource details.

In Fig. 9, the screenshots of case prediction view (a) and offer list (b) are shown. When it comes to prediction view (a), users can provide the input parameters for case predictions (apart from those that are obtained from online sources automatically) that are further used to determine the daily demand of resources. On the other side, in offers view (b), user can see all the resource trading offers for the city under responsibility and accept/reject them.

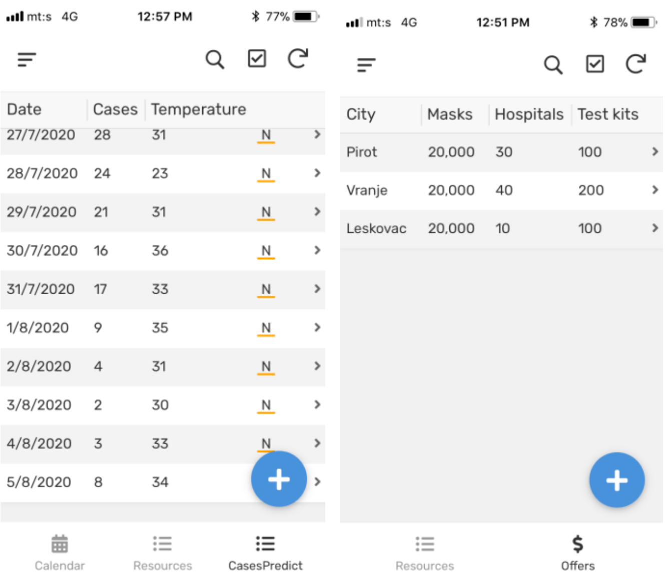

Fig. 9. (a) Case prediction screenshot (b) Offer list screenshot.

A brief summary of the main Google Sheets used for generation of resource planning AppSheet mobile application is given in Table III.

TABLE III
RESOURCE PLANNING APPLICATION GOOGLE SHEETS STRUCTURE OVERVIEW

Sheet	***Col1***	***Col2***	***Col3***	***Col4***
Resources	City [text]	Budget [num]	Hospital places [num]	Test kits [num]
Hospitals	City [text]	Name [text]	Free places [num]	Cost [num]
Test kits	City [text]	Number [num]	Type [text]	Cost [num]
Prediction	Date [dd/mm/yy]	Cases [num]	Temperature [°C]	Lockdown [y/n]
Offer	City [text]	Hospitals [num]	Test kits [num]	Price [num]

IV. EXPERIMENTS AND EVALUATION RESULTS

For purpose of evaluation, the following, the following devices were used: a laptop equipped with Intel i7 7700-HQ quad-core CPU running at 2.80GHz with 16GB of DDR4 RAM and 1TB HDD acting as Edge server, iPhone SE 2016 (dual-core, 2GB of RAM) as representative of iOS smartphones and Xiaomi Redmi Note 7 (octa-core, 3GB of RAM) as representative of Android smartphones. As active internet connection was required to run the apps, 4G network with 50Mbps download/upload speed was used.

Various aspects of evaluation were considered: application development time speed-up (compared to equivalent Android application developed in Android Studio [28]), application delivery time (how much it takes to see the application modifications without re-installing), storage required for AppSheet client, RAM memory utilization (for Android) and average prediction accuracy of AppSheet's predictor in case of new cases and resource demand predictions. In Table IV, the results achieved during the evaluation are given.

TABLE IV
EVALUATION RESULTS

Aspect	**Value**	
Development speed-up [times]	12 times	
Delivery time [s]	iPhone	Redmi
	4.12	3.26
Client app size [MB]	iOS	Android
	.ipa:59 total:96	.apk:13 total:77
RAM utilization [MB]	Case A. 132	Case B. 184
Prediction accuracy [%]	85.6%	

According to the achieved results, the proposed approach of mobile application generation using AppSheet taking Google Sheet as an input significantly speeds up the mobile application development compared to native solutions. For evaluation, the case study (A) was considered as native Android mobile app was already developed in [28]. Using AppSheet, around 30 minutes were needed, while it took around 6h to develop similar application from scratch in Android Studio. However, the native application uses Message Queuing Telemetry Transport (MQTT) protocol and libraries for message exchange, while AppSheet solution logs the violations within spreadsheet.

When it comes to delivery time, it was faster in case of Android device, as it is equipped with more powerful CPU and larger RAM, while AppSheet client is larger in case of iOS. Furthermore, resource planning tool occupies more RAM than the indoor safety monitoring application, as it has much more views and features, such as value prediction. However, indoor safety monitoring application is much heavier than native application in terms of storage required and RAM occupation, as native Android indoor safety monitoring app was just 6MB, while only around 50MB of RAM was required.

Finally, when it comes to accuracy of case predictions, it was higher than in case of [16] which was around 72%, as much more data was available (whole May, June and July) and AppSheet's prediction module is highly optimized, but still fast, as training took just several seconds.

V. CONCLUSION AND FUTURE WORK

According to the achieved results for case studies, the proposed approach of AppSheet-based COVID-19 support mobile application development has huge potential, as it provides rapid creation, fast delivery, relatively low hardware requirement requirements and solid set of features at the same time. Unlike state-of-the-art cross platform frameworks, it requires almost no coding, while it provides more advanced user interfaces, data visualization and AI-enabled capabilities compared to mobile version of Google Sheets. In one of the case studies, the prediction module was used to estimate the daily number of new cases and according to that, the city resource demand with satisfiable accuracy.

However, in future, it is planned to perform tighter integration with blockchain infrastructure in order to enable secure logging of the events that occur, case data and resource availability leveraging smart contracts. Finally, the integration with framework for smart contract verification based on semantic technology [35] is also considered. Moreover, apart from COVID-19 use cases, we plan to adopt AppSheet applications for custom fitness training generation based on user parameters (height, weight and other details) in future.

ACKNOWLEDGMENT

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

[1] V. C. Bala, P. Kumar, "Review on COVID-19: Rise of SARS-CoV-2 Pandemic Outbreak", Borneo Journal of Pharmacy, 3(Special-1), pp. 103-120, 2020. https://doi.org/10.33084/bjop.v3iSpecial-1.1412

[2] P. Zhai et al., "The epidemiology, diagnosis and treatment of COVID-19", International Journal of Antimicrobial Agents vol. 55 issue 5, May 2020, 105955, pp. 1-13, 2020. https://doi.org/10.1016/j.ijantimicag.2020.105955

[3] P. Dawson et al., "Loss of Taste and Smell as Distinguishing Symptoms of COVID-19", Clinical Infectious Diseases June 2020, pp. 1-4, 2020. https://doi.org/10.1093/cid/ciaa799

[4] Coronavirus Incubation Period [online]. Available on: https://www.worldometers.info/coronavirus/coronavirus-incubationperiod/ , last accessed: 11/07/2020.

[5] D. Oran, E. Topol, Prevalence of Asymptomatic SARS-CoV-2 Infection: A Narrative Review, Annals of Internal Medicine, June 2020, pp. 1-7, 2020. https://doi.org/10.7326/M20-3012

[6] Y. V. Chudasama, "Impact of COVID-19 on routine care for chronic diseases: A global survey of views from healthcare professionals", Diabetes & Metabolic Syndrome: Clinical Research & Reviews, Volume 14, Issue 5, pp. 965-967, 2020. https://doi.org/10.1016/j.dsx.2020.06.042

[7] T. Dbouk, D. Drikakis, "On respiratory droplets and face masks", Physics of Fluids 32, 063303, pp. 1-11, 2020. https://doi.org/10.1063/5.0015044

[8] Y. Song et al., "COVID-19 Treatment: Close to a Cure? – A Rapid Review of Pharmacotherapies for the Novel Coronavirus" [preprint], pp. 1-25, 2020. https://doi.org/10.20944/preprints202003.0378.v1

[9] V. Balachandar et al., "COVID-19: emerging protective measures", European Review for Medical and Pharmacological Sciences vol. 24 (6), pp. 3422-3425, 2020. https://doi.org/10.26355/eurrev_202003_20713

[10] M. Nicola et al., "The socio-economic implications of the coronavirus pandemic (COVID-19): A review", International Journal of Surgery, Volume 78, June 2020, pp. 185-193, 2020. https://doi.org/10.1016/j.ijsu.2020.04.018

[11] R. Vaishya, M. Javaid, I. H. Khan, A. Haleem, "Artificial Intelligence (AI) applications for COVID-19 pandemic", Diabetes & Metabolic Syndrome: Clinical Research & Reviews, Volume 14, Issue 4, 2020, pp. 337-339, 2020. https://doi.org/10.1016/j.dsx.2020.04.012

[12] R. P. Singh, M. Javaid, A. Haleem, R. Suman, "Internet of things (IoT) applications to fight against COVID-19 pandemic", Diabetes & Metabolic Syndrome: Clinical Research & Reviews, Volume 14, Issue 4, 2020, pp. 521-524, 2020. https://doi.org/10.1016/j.dsx.2020.04.041

[13] M. Nasajpur et al., "Internet of Things for Current COVID-19 and Future Pandemics: An Exploratory Study" (preprint), pp. 1-19, 2020. https://arxiv.org/abs/2007.11147

[14] D. Marbouh et al., "Blockchain for COVID-19: Review, Opportunites and a Trusted Tracking System" (preprint), pp. 1-17, 2020. https://doi.org/10.36227/techrxiv.12609344.v2

[15] T. Alam, "Internet of Things and Blockchain-Based Framework for Coronavirus (COVID-19) Disease" (preprint), pp. 1-9, 2020. https://doi.org/10.20944/preprints202007.0642.v1

[16] N. Petrovic, Dj. Kocic, "Framework for Efficient Resource Planning in Pandemic Crisis", CIIT 2020, pp. 1-6, 2020.

[17] L. C. Ming et al., "Content Analysis and Review of Mobile Health Applications on COVID-19", JMIR mHealth and uHealth. 30/06/2020:19796 (forthcoming/in press). https://doi.org/10.2196/19796

[18] D. Drew et al., "Rapid implementation of mobile technology for real-time epidemiology of COVID-19", Science Vol. 368, Issue 6497, pp. 1362–1367, 2020. https://doi.org/10.1126/science.abc0473

[19] N. Harraz, "Online Monitoring of On-Shelf Availability", International Conference on Industrial Engineering and Operations Management 2018, pp. 450-461, 2018.

[20] P. Samutrak, C. Kangvaravoot, "Application to Calculate Potential Evapotranspiration", International Journal of Applied Computer Technology and Information Systems: Volume 7, No.1, pp. 35-40, 2017.

[21] D. S. Wijesekara et al., "Developing an electronic record keeping system at a paediatric clinic in Colombo South Teaching Hospital, Sri Lanka", Sri Lanka Journal of Child Health, 49(2), pp.116–124, 2020.

[22] E. Ogunshile, K. Phung, S. Odongo, "C-19 C -A mobile application to support the community during Covid-19 lockdown", CONISOFT 2020, pp. 1-9, 2020.

[23] J. Vanathi, G. SriPradha, "BreakTheChain: A Proposed AI powered Mobile Application Framework to handle COVID-19 Pandemic", Alochana Chakra Journal, Volume IX, Issue IV, April 2020, pp. 108-114, 2020.

[24] K. Hammoudi, A. Cabani, H. Benhabiles, M. Melkemi, "Validating the correct wearing of protection mask by taking a selfie: design of a mobile application "CheckYourMask" to limit the spread of COVID-19" (preprint), pp. 1-5, 2020. https://hal.archives-ouvertes.fr/hal-02614790

[25] A. Imran et al., "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app" (pre-proof), Informatics in Medicine Unlocked, pp. 1-30, 2020. https://doi.org/10.1016/j.imu.2020.100378

[26] X. Li, C. Li, D. Zhu, "COVID-MobileXpert: On-Device COVID-19 Screening using Snapshots of Chest X-Ray" (preprint), pp. 1-14, 2020. https://arxiv.org/abs/2004.03042

[27] J. Lim, "Measuring sports performance with mobile applications during the COVID-19 pandemic", SPSR 2020, 103, v1, pp. 1-3, 2020.

[28] N. Petrović, Đ. Kocić, "IoT-based System for COVID-19 Indoor Safety Monitoring" (preprint), IcETRAN 2020, pp. 1-6, 2020.

[29] L. Morawska, "How can airborne transmission of COVID-19 indoors be minimised?", Environment International vol. 142, September 2020, 105832, pp. 1-7, 2020. https://doi.org/10.1016/j.envint.2020.105832

[30] P. Vanini, "Protection of the Population and the Economy in a Pandemic" (preprint), pp. 1-21, 2020.

[31] Statistika COVID-19 u Srbiji [online]. Available on: https://covid19.data.gov.rs/, last accessed: 29/07/2020.

[32] Zaraženo u danu [online]. Available on: https://covid19.data.gov.rs/infected, last accessed: 29/07/2020.

[33] Serbia weather conditions [online]. Available on: https://www.accuweather.com/en/rs/serbia-weather, last accessed: 29/07/2020.

[34] N. Petrović, Đ. Kocić, "Data-driven Framework for Energy-Efficient Smart Cities", Serbian Journal of Electrical Engineering, Vol. 17, No. 1, Feb. 2020, pp. 41-63. https://doi.org/10.2298/SJEE2001041P

[35] N. Petrović, M. Tošić, "Semantic Approach to Smart Contract Verification", Facta Universitatis Series Automatic Control and Robotics 19(1), pp. 21-37, 2020.