Design and Implementation of an Android Smart Parking Mobile Application

Samar Alkhuraiji

King Abdulaziz University, Department of Computer Science, Jeddah, Saudi Arabia

***Abstract* – Smart parking is an important sector of smart city technologies. This paper presents the design and implementation of a smart parking application in a university campus to help university personnel pre-book a parking space swiftly and easily. Each parking slot is numbered with a distinctive number and has an infrared sensor. The sensors are connected to an Arduino device to indicate whether the position is vacant or busy. The application uses several programming techniques in Android Studio and SQL Database. The efficient management of the available parking spaces is demonstrated and can be extended to cover other areas and venues.**

***Keywords* – mobile application, smart parking, online reservation, sensors**

1. Introduction

It is becoming increasingly challenging to find available parking spaces, particularly at city centers, shopping malls, large department stores, airports, crowded events, and campuses. This situation is also observed at King Abdulaziz University (KAU), especially at peak hours, when most students have classes simultaneously. Students have to go back and forth several times before finally finding a parking space. Therefore, an efficient car park system is crucial for saving students time and allow them to reserve their parking slot before coming to the university.

DOI: 10.18421/TEM94-06
https://doi.org/10.18421/TEM94-06

Corresponding author: Samar Alkhuraiji,
King Abdulaziz University, Department of Computer Science, Jeddah, Saudi Arabia.
Email: salkhuraiji@kau.edu.sa

Received: 15 April 2020.
Revised: 23 September 2020.
Accepted: 05 November 2020.
Published: 27 November 2020.

© 2020 Samar Alkhuraiji; published by UIKTEN. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License.

The article is published with Open Access at www.temjournal.com

According to our survey results conducted in March 2017 on 337 students from KAU, 84% of the participants struggle to find parking spaces at the KAU. Based on the survey results, we propose a reservation mobile application for reserving parking slots before students come to their classes, built on advanced sensor and mobile communication technology.

The paper presents the design and development of a mobile application that can help KAU community find and reserve parking slots for their cars easily and thus save time before they come to classes.

The paper describes the design and development of the prototype system, consisting of a backend system and a mobile client app that runs on a smartphone. The application also interacts with a network of area sensors connected to an Arduino platform. It will enable the users to make a reservation, submit their booking, and cancel it.

The paper novel contributions are as follows:

- It presents a mobile application that can provide parking availability information as well as suggestions based on real-time sensor data.
- It provides a methodology for developing a user-friendly interface on Android handsets that can guide users in a fast and efficient way to perform online parking reservation for a specific duration.

The system integrates an IoT sensor network, an appropriate database backend, and a lightweight android mobile app that can operate under various traffic and incoming/outgoing vehicle rate scenarios, thus covering different scalability requirements.

The related state-of-the-art methods are discussed in Section 2, and the analysis, design, and architecture of the proposed system are discussed in Section 3. The prototype and mobile app development are presented in Section 4, along with illustrative screenshots and functionalities. Section 5 illustrates the test results of the mobile app. Finally, the conclusion is presented in Section 6.

2. Related State of the Art

Smart parking frameworks and related applications have already been proposed in the literature, and representative solutions include the works in [1], [2], [3], [4]. For example, in [5], a

smart parking solution for online reservations using infra-red (IR) and sensors and facial recognition for authentication of the car reservation is proposed. Different parking guidance strategies are presented here, and their constraints are highlighted. In blind search [6], the driver searches for a parking space at random because there is no parking information. If the driver finds an available space, the search ends, else, the driver continues to search for a position in the nearby parking lot until a space is found. The parking information sharing (PIS) mechanism is currently used in designing smart parking systems. The driver gets information about the available spaces at the parking lot and decides to go to the desired parking lot. As shown in [7], the application depends on first come first serve. However, during peak hours, many drivers will be flocking for the few remaining spaces. The buffered PIS (BPIS) alleviates the problem of PIS by considering a buffer for the last remaining spaces when disseminating live availability information. If the empty parking spaces are less than a threshold, the system shows that the parking lot is full. Since the system is very sensitive to the rate of incoming and outgoing vehicles, it is not easy to identify an optimum threshold for the buffer; if the buffer is too small the problem cannot be eliminated. In contrast, if the buffer is too large, the parking space utilization will be low.

As shown in [6], the driving distance in the blind search is the worst because it is an open-loop system where the driver searches for parking without any information. The PIS and BPIS are better than blind search because the search for parking space is based on available information. However, rather than a search based system, a policy based on the prior reservation is the best as it guarantees parking space without the search process.

There are several commercial parking guidance applications available in the market. The Mawgif application [8] is used for paying parking fees in Saudi Arabia (national parking company). It assists in locating available parking spaces around the city and is an alternative to using automated parking ticket issuing and payment machines located near parking lots. The driver is charged based on the identified parking position and actual parking duration. The application requires electronic account creation and online payment. The interface of the application is shown in Figure 1.

Figure 1. Mawgif application interface [8]

Parking Panda [9] system allows the drivers to search for and reserve parking at many parking lots and garages in several cities across North America and allows the drivers to compare parking options and prices and guarantees a parking spot. The interface of the application is shown in Figure 2. The application allows reserving a parking spot in advance, online accessing and redeeming parking reservations, booking a space at daily and monthly rates, and paying safely and securely using credit card. Best parking [10] is an online parking search engine, with the most accurate data for any parking application. It helps find the cheapest and most convenient parking facilities in 105 cities and 115 airports across North America. It also allows for booking at some of the parking places. The Parking Spot [11] covers 39 locations at 21 airports, and helps drivers to find the right parking space and make reservations without prepayment. The interface of the application is shown in Figure 3.

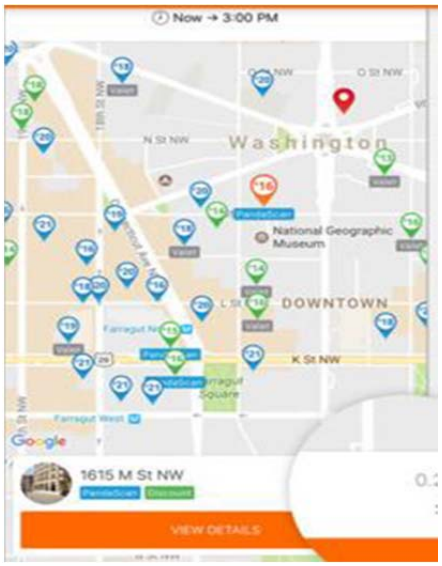


Figure 2. Parking Panda application interface [9]

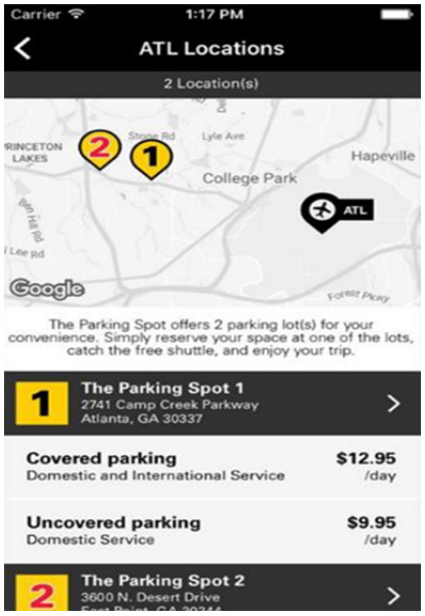


Figure 3. The Parking Spot application interface [10]

Table 1. provides a detailed comparison between existing and proposed mobile applications.

3. System Analysis and Design

The analysis of the proposed system based on data gathering techniques to identify how the application advises users on the optimal path and minimum cost, the definition of the functional and non-functional requirement, and the required skills to build the application is considered in this section. The proposed system consists of two actors:

i. The *user* is the driver of the car who is booking a parking space.
ii. The *admin* controls the proposed system components.

The system allows the user to perform the following tasks:

(a) Log in to the application using personal information (university id, password) and to logout. Further details are illustrated in the screenshots of Figure 7.
(b) Manage the car information (license plate number) by adding new car information or updating them.
(c) Manage a reservation either by making a new reservation or by editing, deleting, or updating an existing one. Each reservation has a start time and end time.
(d) Report a violation by typing the license plate number.

Additionally, the system allows the admin to view statistics about car drivers, in addition to displaying current available parking spaces. In terms of non-functional requirements, the system availability will be continuous (24/7), the response time when logging in will be less than 8 s. The application will provide ease of access and usability to allow a native user to use the application efficiently without any difficulties.

The system architecture is shown in Figure 4. It is based on a software system that links hardware and

Table 1. Comparison of related mobile apps

Criteria		Competitor Application				
		Mawgif	Parking panda	Best Parking	The Parking Spot	Proposed System
Design	platform	IOS / Android	IOS/ Android	IOS / Android	IOS / Android	Android
	website	yes	yes	yes	yes	no
Payment	Free	yes	yes	yes	yes	yes
	Payment way	Credit card	Credit card	Credit card	--------	--------
Services						
	Account?	yes	yes	yes	yes	yes
	User Activites:					
	booking	no	yes	yes	yes	yes
	Compare parking	no	yes	yes	yes	no
	pay	yes	yes	yes	no	no
	search	yes	yes	yes	yes	no
	System activities :					
	navigation	yes	yes	yes	yes	no
	notification	yes	yes	yes	yes	yes
	Technology used :					
		-------	barcode	barcode	barcode	Arduino

software. The architecture of the system is based on a four-layer model, the Users-Android application interface-Server-Parking area. Users can make a booking through an android application interface that is linked to the server to send and receive information about users. The server acts as an intermediary between the users and area parking. The IR sensor devices are responsible for area parking (hardware) and sending/receiving parking information to the server through the Arduino platform [12], [13].

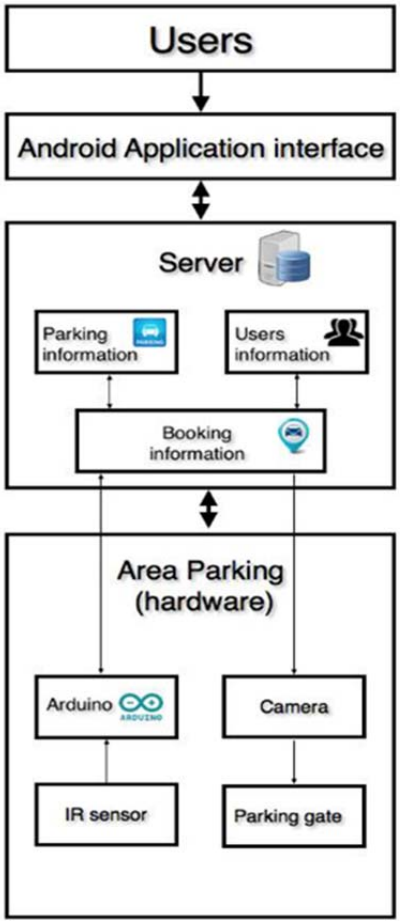


Figure 4. System architecture

Figure 5. shows a message sequence diagram outlining the interactions between objects and actors.

4. Prototype and Mobile App Development

The parkig prototype that was developed for testing is illustrated in Figure 6.

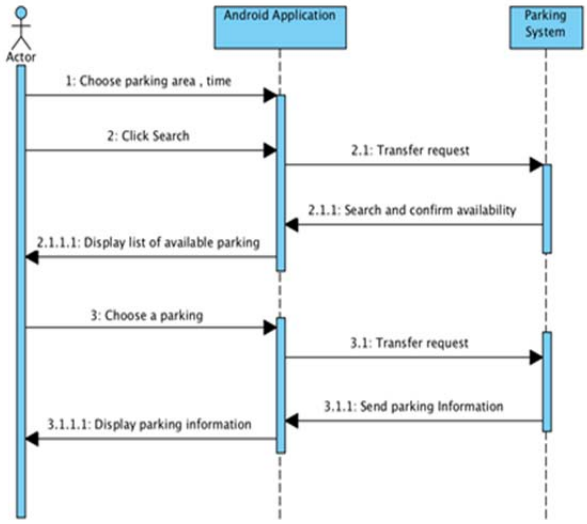


Figure 5. Message sequence diagram

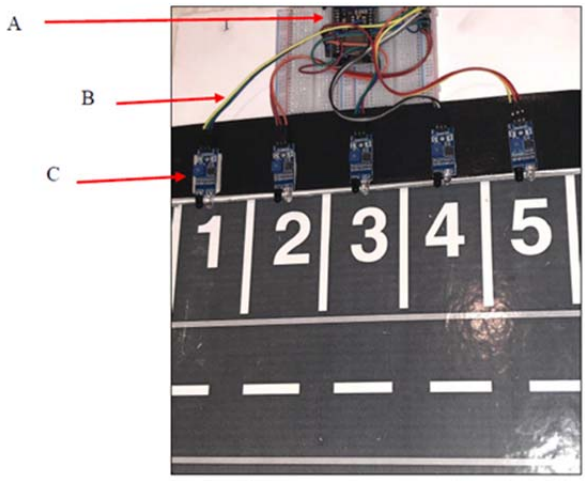


Figure 6. Developed prototype

The Arduino platform (A) reads the output from each IR sensor (B), via its wired connection (B), and sends it to the database. The IR sensors are embedded on a small digital board. The prototype includes five sensors to send information to the Arduino via their output interface ('0' and '1' values corresponding to parking spots availability).

The mobile app interface screens are shown in Figure 7.

For the mobile app implementation, Android Studio v 2.2.3 was considered. The software development was performed using the ASP.NET Microsoft Visual Studio framework [14], while the database was implemented using Microsoft SQL Server and SQL Server Management Studio (SSMS) [15]. Finally, the Arduino IDE platform was ported and connected to the internet via NodeMCU ESP8266 Wi-Fi using the open-source code to manage the information collected from the parking IR sensors [13].

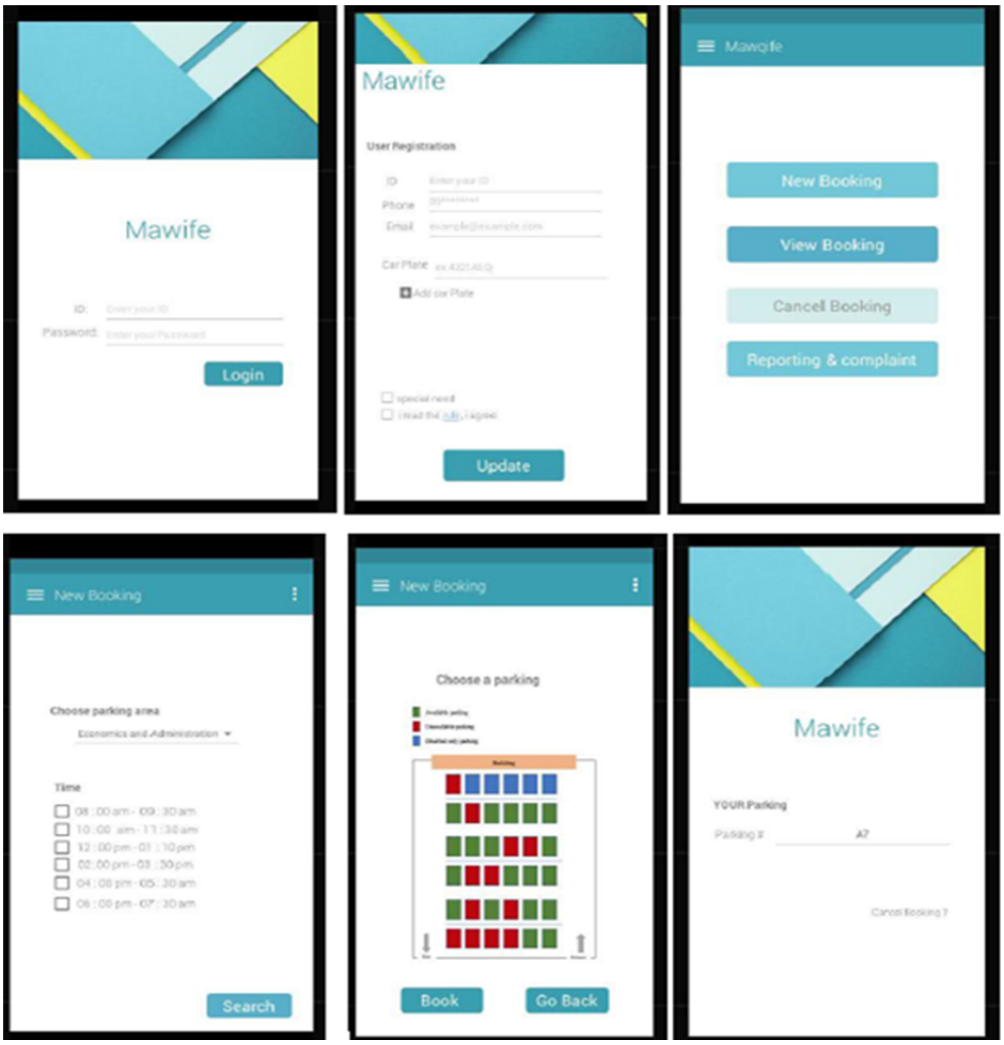


Figure 7. Mobile app interface screens

5. System Assessment

The 'black box' testing technique was used to check if the system functionality meets user expectations and to ensure that the application was working correctly. Ten users interacted with the system interface, to test the following five test scenarios:

1. Log in to the application using University ID, as shown in Figure 8.
2. Create a new booking by choosing time, as shown in Figure 9.
3. Provide all booking history, as shown in Figure 10.
4. Cancel booking, as shown in Figure 11.
5. Reporting a complaint, as shown in Figure 12.

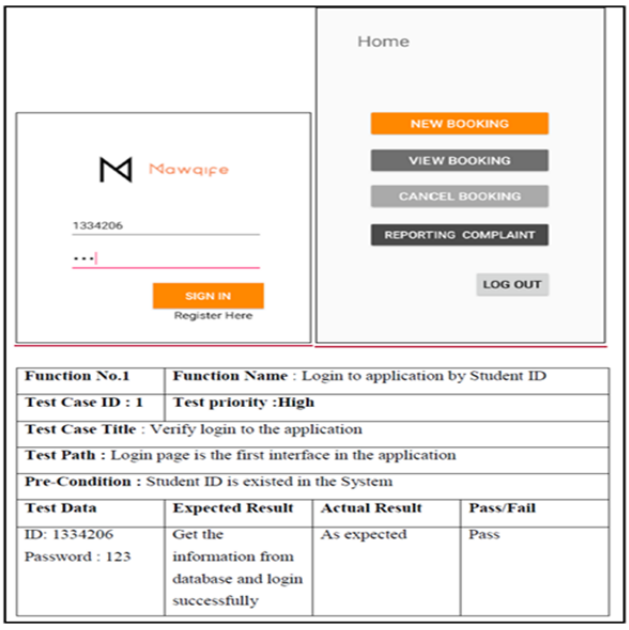


Figure 8. Function test 1

Function No.2	**Function Name:** Create a new booking by choosing time		
Test Case ID : 2	**Test priority** :High		
Test Case Title : Verify Create a new booking			
Test Path : Go to Home Screen → New Booking→ Choose time →Choose green parking			
Pre-Condition : Login to the application			
Test Data	**Expected Result**	**Actual Result**	**Pass/Fail**
Time :8:00-9:30	Provide all available parking from 8:00 to 9:30	As expected	Pass
Press park 1	Park 1 will be reserved	As expected	Pass

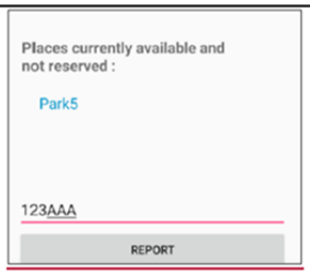
Figure 9. Function test 2

Function No.3	**Function Name:** Provide all booking (History)		
Test Case ID : 3	**Test priority** :High		
Test Case Title : Verify Provide all booking			
Test Path : Go to Home Screen → View Booking			
Pre-Condition :Have a previous booking			
Test Data	**Expected Result**	**Actual Result**	**Pass/Fail**
	Get the information from database and provide it	As expected	Pass

Figure 10. Function test 3

Function No.4	**Function Name: Cancel Booking**		
Test Case ID : 4	**Test priority** :High		
Test Case Title : Verify Cancel Booking			
Test Path : Go to Home Screen → Cancel Booking			
Pre-Condition :has a previous booking			
Test Data	**Expected Result**	**Actual Result**	**Pass/Fail**
Press the Cancel button	Delete the booking	As expected	Pass

Figure 11. Function test 4



Function No.5	**Function Name:** Reporting		
Test Case ID : 5	**Test priority** :High		
Test Case Title : Verify Reporting			
Test Path : Go to Home Screen → Reporting Complaint			
Test Data	**Expected Result**	**Actual Result**	**Pass/Fail**
Car plate: 123 AAA	Send the information	As expected	Pass

Figure 12. Function test 5

From Figures 8. to 12., it can be seen that the system passed all tests, and the mobile app operated according to the expected specifications and requirements.

6. Conclusions

This paper presents a prototype system and mobile app implementation for managing parking spaces within a university campus to assist university personnel, students, and visitors to find a parking space in the easiest and fastest way by pre-booking via the application. The system involves the use of IR sensors and an Arduino platform that interfaced with a Microsoft SQL Database backend to provide information on parking availability and allow for parking space pre-booking on a developed android mobile app. The proposed system was shown to have passed all the tests and operated according to the expected specifications and requirements, allowing for its use in various areas and events of increasing scalability. The proposed application provides many advantages, such as knowing the daily peak time of parking at the parking lot. This information will help the university admins to know which time holds the most lectures during the day because it has the most number of students attending. The university administrators can then fix the scheduling error and distribute classes evenly during the day.

References

[1]. Lin, S. F., Chen, Y. Y., & Liu, S. C. (2006, October). A vision-based parking lot management system. In *2006 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 4, pp. 2897-2902). IEEE.

[2]. Kianpisheh, A., Mustaffa, N., Limtrairut, P., & Keikhosrokiani, P. (2012). Smart parking system (SPS) architecture using ultrasonic detector. *International Journal of Software Engineering and Its Applications*, *6*(3), 55-58.

[3]. Agrawal, K. G., Sadhani, S., Ahuja, R., Khandelwal, S., & Koul, S. (2015). Parking navigation and payment system using IR Sensors and RFID technology. *International Journal of Computer Applications*, *111*(15).

[4]. Chinrungrueng, J., Sunantachaikul, U., & Triamlumlerd, S. (2007, January). Smart parking: An application of optical wireless sensor network. In *2007 International Symposium on Applications and the Internet Workshops* (pp. 66-66). IEEE.

[5]. Thangam, E. C., Mohan, M., Ganesh, J., & Sukesh, C. V. (2018). Internet of Things (IoT) based smart parking reservation system using raspberry-pi. *International Journal of Applied Engineering Research*, *13*(8), 5759-5765.

[6]. Wang, H., & He, W. (2011, April). A reservation-based smart parking system. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 690-695). IEEE.

[7]. Kousalya, G., & Selvakumar, N.(2019). *The Real time Mobile Application for Reservation based Smart Car Parking Slot with IoT*. 5th International Conference on Latest Trends in Science, Engineering and Technology (ICLTSET'19).

[8]. Mawgif National Parking Co. Ltd Android Mobile Application. (2019). Retrieved from: https://play.google.com/store/apps/details?id=com.mawgif&hl=en [accessed: 21 February 2020].

[9]. Parking Panda On-Demand Parking Deals OS Mobile Application. (2020). SpotHero, Inc. Retrieved from: https://apps.apple.com/us/app/parking-panda/id550285323 [accessed: 21 February 2020].

[10]. ParkWhiz and BestParking Mobile Application. (2020). Retrieved from: https://www.parkwhiz.com/, [accessed: 25 February 2020].

[11]. The Parking Spot Mobile Application. (2020). Retrieved from: https://www.theparkingspot.com/ [accessed: 25 February 2020].

[12]. Arduino Platform Portal. (2020), Retrieved from: https://www.arduino.cc/ [accessed: 03 March 2020].

[13]. How to setup NodeMCU drivers and Arduino IDE, Marginally Clever Robots. (2017). Retrieved from: https://www.marginallyclever.com/2017/02/setup-nodemcu-drivers-arduino-ide/ [accessed: 10 March 2020].

[14]. Microsoft ASP.NET: A framework for building web apps and services with .NET and C#, Microsoft. (2017). Retrieved from: https://dotnet.microsoft.com/apps/aspnet [accessed: 10 March 2020].

[15]. What is SQL Server Management Studio (SSMS)?, Microsoft. (2017). Retrieved from: https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15 [accessed: 11 March 2020].