RESEARCH ARTICLE

SIAAA-C: A student interactive assistant android application with chatbot during COVID-19 pandemic

Saadeh Z. Sweidan | Sarah S. Abu Laban | Njood A. Alnaimat | Khalid A. Darabkh

Department of Computer Engineering, The University of Jordan, Amman, Jordan

Correspondence
Khalid A. Darabkh, Department of Computer Engineering, The University of Jordan, Amman 11942, Jordan. Email: k.darabkeh@ju.edu.jo

Abstract
Nowadays, smartphone applications (apps) are literally used in all life aspects, improving the quality of different services in addition to saving time and effort. However, the majority of currently used apps in different universities around the world are limited in their features and not very popular among students who may suffer to get useful information using traditional guiding methods. However, the importance of these apps is constantly increasing, especially in times of crises like the COVID-19 pandemic during which people were forced to work from home and the human interaction was reduced to its minimal degree. In this paper, we present a Student Interactive Assistant Android Application with Chatbot (SIAAA-C), which is basically a personal electronic guide that is used by students to get various and effective academic services. The app has a number of useful features including a campus map, multiple types of notifications, and most important, an efficient built in chatbot that responds to both Arabic and English written queries and covers a wide range of academic topics ranging from basic to complex. Even more, SIAAA-C was designed to have Arabic and English interfaces to be used in the University of Jordan (UoJ), which has one of the biggest campuses in the Arabic area with more than 40,000 local and foreign students. Moreover, SIAAA-C was practically tested in a real-life environment by a group of UoJ students who were selected from different faculties during the pandemic peak period and the feedbacks were very promising, based on the outcomes of an evaluation survey that was answered by 102 students after a 1-month test period.

KEYWORDS
Android, Chatbot, COVID-19 pandemic, interactive learning, universities

1 | INTRODUCTION

Being a student in a big university can sometimes be a real challenge. In fact, higher education institutes usually expect their students to show a high degree of independency and maturity by being responsible for managing their academic lives’ details and performing daily tasks by themselves like finding the location of a specific lecture, reaching any facility they want to use, contacting an instructor during the designated office hours, and selecting next semester courses and lectures’ times wisely during the registration period. As a matter of fact, most universities usually provide their students with a number of assisting tools to guide them through...
various academic situations and to answer their questions accurately [12]. Nowadays, every university has an official website to help visitors from all over the world to get any details they need about its programs, acceptance policies, fees, and other topics. Besides, the website can be used by the students to get information about the university regulations, academic staff contact info, details of different types of activities, maps for different locations inside the campus, and even answering students’ questions about different issues by a dedicated team of employees [11]. However, a university’s website may not be easy to navigate through due to its complicated and very detailed menus in addition to the usually late responses to students’ queries, which makes its usage very limited as a student assistance tool. Also, every instructor is usually assigned as an academic supervisor or counselor to a group of students to respond to their queries and provide different academic consultations to them. This assisting tool may be very useful for students but also limited to certain areas and available during certain hours only, because every supervisor is usually connected to tens of students and cannot be available to each of them all the time [13]. Even more, it is very common in most universities that senior students volunteer in guiding teams to help freshmen during the registration period and give them tours inside the campus [1]. Although these initiatives can be very useful, they are usually seasonal and may be used by some seniors to serve certain political reasons.

Smartphones have many distinguished features like their big-size touch screens, large memories, digital cameras, and their internet connectivity [15,24]. Figure 1 shows the increment in smartphone users worldwide between the years 2014 and 2020, which clearly reflects the growing importance of these gadgets in our lives. Actually, these computerized devices with their accompanying applications (apps) are being used in every life aspect you can think of to cover all age ranges and work fields [14,31,32]. Nowadays, we can find dedicated apps for online shopping, contacting people in other countries, and even learning a new language [3,4,24]. Moreover, many apps are being used to help young people to learn different skills [7,8,21,30] and people with special needs to overcome their difficulties [5,16,17,28,33]. According to that, it was only a matter of time to start seeing universities around the world developing dedicated apps to be used by their students as daily assistance tools due to their availability all the time to automate the guiding process and save human resources [18]. These apps can come in different forms ranging from a general and simple students’ guide with limited features to a more complex interactive learning tool with advanced features like smart chatbots and financial services [10]. In fact, most of currently used apps in universities are limited in their features with poor designs and inflexible interfaces, which makes them less popular among students than expected and reduces the benefits gained by adopting them.

In December 2019, a severe acute respiratory syndrome coronavirus 2 infection occurred in Wuhan, China, and then spread fast all around the world [2]. In January 2020, the World Health Organization (WHO) declared the outbreak a Public Health Emergency of International Concern and then officially called this disease as coronavirus disease 2019 (COVID-19) in February 2020 [42]. By August 7, 2020, there were more than 18 million confirmed cases of COVID-19, including 709 thousand deaths, with the absence of an efficient cure, which led the whole world to deal with a delicate situation [39]. Figure 2 shows an exponential increment of infections worldwide in the period between January 1 and July 31, 2020. Actually, this pandemic has affected
the whole world by putting it in a mandatory quarantine to reduce the human interaction to the least possible amount and control the rapid spread of this virus. As a result, the majority of human activities were forced to temporarily shut down and most people had to work from home. The field of higher education was not an exception where most universities around the world closed their campuses and replaced the traditional learning methods with online learning using different software tools and applications. Naturally, this sudden transfer caused a confusion for both administrative staff and students in most universities where a lot of regulations had to be modified according to the new situation. Moreover, the administrative staff was unable to respond to the enormous flood of questions and queries sent by students regarding different topics. As a result, providing different automated tools that respond to different queries instantly and accurately became a necessity during the pandemic period and beyond.

The University of Jordan (UoJ) is considered the oldest and largest university in Jordan with 24 schools located in two cities and more than 43,000 registered students during the academic year 2019–2020 [35]. Providing sustainable guiding services to this massive number of students, especially during the COVID-19 pandemic, may be a real challenge to the administration and may require new revolutionary solutions that are based on new technologies and automation. Actually, this challenge encounters the administrative staff in most universities around the world, especially the ones with a large number of students.

In this paper, we present a Student Interactive Assistant Android Application with a Chatbot (SIAAA-C). Actually, we were initially motivated by the fast spread of smartphones among university students and the enormous popularity of their apps in all life fields. Moreover, the importance of presenting an efficient personal electronic guide to help the students through different academic life situations has encouraged us to proceed with this study. Besides, the pandemic itself has increased the necessity of presenting apps similar to ours, as the whole world has adopted the online learning path that is here to stay. Finally, our biggest motive was to try to fill the obvious gap of not having any efficient Arabic chatbots in all Arabic universities apps and websites. SIAAA-C aims to present a bilanguage chatbot, which can be very useful to both local and foreign students and help the majority of them to interact with it using their mother tongue, thus making the received answers more understandable to them. The major contributions of this study are summarized as follows:

1. Providing a useful tool through which a student can find his/her personal information and current course schedule.
2. Using the included google map of the university campus to reach different facilities and locations easily.
3. Using notifications to remind students about upcoming lectures, alert them about absences, and even inform them about scheduled exams.
4. Providing an interactive query tool in the form of a chatbot that imitates a complete human textual conversation with a student in both Arabic and English languages to serve both local and foreign students regarding different academic needs like finding places to eat in, locating parking spots, buying books, getting useful feedbacks about different courses, and getting academic staff contact information.
5. Suggesting a convenient new semester course schedule for each student using the chatbot depending on the academic situation and courses’ prerequisites.

The rest of this paper is divided as follows, Section 2 discusses a number of related methods and similar apps and compares their features with SIAAA-C. Section 3, however, covers the overall app design in detail including the chatbot design and its training process in addition to a number of code samples. Then, Section 4 presents a comprehensive run scenario that explains all the features included in SIAAA-C with real-life examples. Next, Section 5 tests the app practically and then reviews the evaluation survey outcomes to assess it in a statistical way. Finally, Section 6 concludes this study.
In the context of interactive learning environments, several tools have been developed all over the world covering different educational areas. Nowadays, the concept of a university app has become very common and is being widely used in higher education community. These apps are expected to spread fast and become very essential for every student, especially during the COVID-19 pandemic that has affected all human activities and has given an additional push to the automation process in all fields. We present here a number of well-known university apps that are similar to SIAAA-C, show their features, and then compare them with our work.

2.1 San José State University App (SJSU)

SJSU is a public university located in downtown San José, CA, USA, which has a dedicated app that is shown in Figure 3. This app is used by students and has many interesting features like providing a list of upcoming events in the campus and providing a link to Google Maps to help reaching the locations of different faculties and other facilities. Besides, this university presents a useful library chatbot to help students find and borrow books in an efficient way [25].

The comparison between San Jose app and chatbot with our work has led to the following conclusions:

1. **Notifications**: SJSU does not notify students about their upcoming lectures, exams, or events, unlike SIAAA-C that provides this essential feature.
2. **Chatbot features**: SJSU main library’s chatbot is only concerned with borrowing books. However, SIAAA-C chatbot is used to respond to different students’ academic queries.
3. **Chatbot accessibility**: SJSU chatbot is not a part of its application and can be accessed through its website. However, SIAAA-C has a built-in chatbot.

2.2 University of Sharjah Application

University of Sharjah is located in the United Arab Emirates and has one of the biggest campuses in the Arabic area. Its application is mainly used to direct the student to the right web page in the official website along with a number of services that can be shown directly in the app such as News, Events, and Maps [23]. Moreover, the University of Sharjah has a web-based chatbot to answer students’ queries, as shown in Figure 4.

The following points conclude the comparison between the University of Sharjah app and chatbot with SIAAA-C.

1. **Notifications**: The University of Sharjah application does not notify students about their lectures, exams, or events, whereas SIAAA-C has this feature.
2. **Chatbot language**: Both chatbots respond to Arabic and English queries.
3. **Chatbot accessibility**: The University of Sharjah chatbot is considered as a part of its website page and is independent from the app itself. However, SIAAA-C comes with a built-in chatbot.
4. **Chatbot features**: The University of Sharjah chatbot is only concerned with admission-related queries. However, SIAAA-C’s chatbot is a more general academic one.
2.3 Princess Sumaya University for Technology

Princess Sumaya University for Technology is located in Amman, Jordan, and considered as one of the most advanced Jordanian universities in the field of technology. Moreover, its app, which is shown in Figure 5, includes many interesting features, making it one of the best student assistant apps used in Jordan. Its features include showing the current courses’ and exams’ schedules, registration information, and even the financial details for each student [20].

Upon comparing this app with ours, we can conclude the following:

1. **Notifications**: Princess Sumaya University application sends notifications to remind students about important events, similar to SIAAA-C, which also sends lecture and exam reminders.

2. **Chatbot**: Princess Sumaya University application does not have a chatbot for students’ queries, unlike SIAAA-C, which has it as an essential feature in the app.

3. **Maps**: Unlike SIAAA-C, this app does not provide a map to help students reach different locations in campus.

Table 1 shows a detailed comparison between SIAAA-C and the three applications we previewed in this section regarding a number of major features. As the table shows, our work suggests a complete solution that offers many useful features to the students in both Arabic and English languages with an interactive chatbot that acts like a real guide and answers different types of queries.

3 | SYSTEM DESIGN AND DEVELOPMENT

3.1 Software tools

SIAAA-C was designed and built using Android Studio as an Integrated Development Environment (IDE) and JAVA as a programming language in addition to using Android Firebase to build the system’s database. Android Studio is considered the official IDE for most Android applications including ours due to its remarkable features [41]. The instant run button, for example, allowed us to make small changes in the code and test them immediately, saving time and effort during the coding process.
phase. We have also found that the intelligent editor, however, is very useful for modifying code with the hints it shows. Moreover, Android studio provides a robust and flexible system build process [37]. It is optimized to run on different Android devices including tablets, smart TVs, in addition to smartphones of course [6,26]. However, Java is a machine-independent programming language that was developed by Sun Microsystems in 1995 [9]. Actually, it is considered the official language for Android app developers all around the world for many reasons, including the fact that it runs on a virtual machine so that there is no need to recompile it for different smartphone types [36]. Moreover, Java codes are easy to secure, which is obviously a required feature in most apps with the increasing demands to protect the users’ information [9]. On the basis of all these aspects, JAVA was the logical choice as the programming language used in SIAAA-C.

Android Firebase is considered as a cloud service provider that allows Android app developers to obtain organized data for mobile apps. Actually, Firebase can be used to build Android, iOS, and even web-based applications with storage and real-time data, because it is easy to use and allows quick data reading and writing [29]. In fact, Data are stored as JavaScript Object Notation and synchronized in real time to every connected client [19]. However, the Firebase Realtime Database can be accessed directly from a web browser or mobile device, and hence there is no need for an application server, which was very convenient to us in our work [19]. Moreover, data and security validation are available through the Firebase Realtime Database Security Rules, which are expression-based rules that are executed when data are read or written [27].

### 3.2 System overall design

Developing a successful application that is concerned with helping university students on daily bases forces a number of essential requirements including accessibility, dependence, and ease of use. Actually, smartphones allow users to access different apps at any time and from different places in the case of Internet connection availability [34]. This fits completely with our app description, which was designed to be used at any time by UoJ students inside or outside the campus. Moreover, Android is the most popular operating systems among smartphone users in Jordan, based on different studies [33]. As a result, our obvious choice was to build SIAAA-C as an Android smartphone app to reach the majority of UoJ students. However, our app is expected to be very dependable, as it is related to critical academic information.

| Features                      | SIAAA-C | San Jose State University application and chatbot | The University of Sharjah Technology application and chatbot | Princess Sumaya University for Technology application and chatbot |
|-------------------------------|---------|---------------------------------|---------------------------------|---------------------------------------------------------------|
| Notifications                 | Yes     | Does not provide notifications  | Does not provide notifications  | Does not provide notifications                                |
| Chatbot language              | English and Arabic | English only | Website-based | Website-based |
| Chatbot accessibility         | Built-in the app itself | Dedicated to library queries only |                               |                                                                 |
| Chatbot features              | Answer to all academic queries |                               |                               |                                                                 |
| Chatbot                      | Yes     | Yes                             | Yes                             | No                                                            |

**TABLE 1** Comparing SIAAA-C with three related apps
and includes many important and sensitive features like reminding students regarding lectures’ and exams’ dates and times. As our app is meant to be used by university students, we had to carefully consider every aspect in the app’s design to create a user-friendly and attractive interface. For example, the colors and button sizes were selected to deliver a feeling of comfort and relaxation. Moreover, the screens are simple, direct, and easy to use. We think that our graphical user interface is very convenient for all students who may have different skill levels regarding high technology and smartphones usage. Figure 6 shows the general design of our app where a student can use android smartphones with Internet connection to access SIAAA-C and get benefits from the features included in it. The access to all the students’ personal details in addition to related courses’ and instructors’ information is gained through a Firebase database that does not require using any dedicated servers and does not cause any delay. Finally, the student will be able to send different types of queries to a built-in interactive chatbot, which will be discussed in detail soon.

Firebase database stores data in the form of nodes [40]. Figure 7 shows a Firebase screen with a number of node types used in our database like Chatbot, Questions, Schedule, and Users. Other used nodes include courses, absence records, maps, and many more. A Users node contains a specific student’s personal information, as the figure shows, like name, current grade point average (GPA), ID, email, faculty, specialty, counselor name, and current schedule.

### 3.3 Chatbot flow diagram and training

One of the most essential features included in SIAAA-C is the interactive chatbot, which allows a student to have a complete dialogue with SIAAA-C in a humanized way and to get answers to different academic questions or queries. To build an efficient chatbot, we had to store all the expected students’ questions in the database along with their expected answers to match them with any incoming queries.

Figure 8 shows a flow diagram that explains the procedure followed by the chatbot in response to different students’ queries. When a student sends a query to the chatbot, it will be checked first to find if it exists as a stored question in the database or not. When no matching question is found to a specific query, the chatbot sends back a message to the student, declaring that it does not have an appropriate answer for that query, which in turn will be stored in the database to be processed later. However, if an incoming query has a matching question stored in the database, it will be treated based on its type as a static or nonstatic query. A static query does not require any variables or additional entries and has the same answer for all students like, for example, asking about specific instructor office hours.

However, a nonstatic query means that the same matching question may have a different response to each student based on his/her personal information and may require the chatbot to ask a number of additional questions to gather more detail about the academic situation before giving an accurate answer or solution. Actually, the nonstatic query in SIAAA-C is related to suggesting a new semester course schedule for a student based on his/her academic situation. As a response to this query, the chatbot will ask the student to enter his/her current GPA, academic year, and the number of successfully passed credit hours to suggest the most convenient schedule based on these factors. First, if a student’s GPA...
FIGURE 8  SIAAA-C chatbot flow diagram
is below 2.0, the chatbot will suggest a schedule that includes repeating any previously failed courses, if existing, and taking a number of basic elective courses with a maximum load of 15 credit hours or less to raise the student's GPA and avoid any possible academic penalties in the future. Second, if a student's GPA is 2.0 or more and the number of the successfully passed credit hours is less than the suggested number by the indicative plan for his/her current academic year, the chatbot will then suggest a schedule with the maximum allowed number of credit hours in one semester, which is 18 in UoJ regulations, for example. Finally, for a student with a GPA over 2.0 and a normal number of passed credit hours, the chatbot will simply suggest a course schedule based on his/her current case by following the indicative plan and the current academic year.

Training the chatbot itself was the most difficult task through the whole SIAAA-C developing process, as the expected queries should cover many academic topics in addition to the fact that one query can be written in different formulas that may have the same meaning. To build a useful and realistic chatbot, we tried to collect most of the expected queries with the help of UoJ students by asking them about the academic and daily questions they usually have. After a 1-month data collection period through visiting different UoJ faculties personally and publishing a related survey online, the students' preferences were as the following: instructors' office hours and contact info, courses study resources, courses outlines, suggested schedule for next semester, and the location of different places inside the campus. Next, we have processed the collected questions by classifying them into smaller categories, removing any possible redundancies, and finally writing an answer for every question and store it in our database.

However, the matching process between incoming queries and stored questions was designed based on a mathematical comparison as follows: Every stored question in the database has a total of 100 points that are divided on the basis of the number of key words in it to calculate the word weight based on this equation: (word weight = 100/number of key words). Key words in a stored question include all of its important words and exclude a number of conjunction words like The, And, A, An, Or, in addition to other words. If a stored question has five key words, for example, then each of them will have a weight of 20 points.

The matching process for any incoming query includes comparing it with every same language question stored in the database to find the number of matched words divided on the basis of the total number of the stored question words and calculate its accuracy rate as a percentage value between 0% and 100%. The stored question with the highest accuracy rate will be matched with the incoming query, and its answer will be used by the chatbot as a response message. For example, if an incoming query has an accuracy rate of 80% for one stored question and the others have lower percentages, then the answer of that particular question will be used as a query response.

An exact match will happen when the accuracy rate for one stored question is 100%. However, if an incoming query's highest accuracy rate is less than 50% for all stored questions, then no match will happen and the chatbot will send a message “I don't have an answer for this question” and store the query itself in the “Chatbot”

**FIGURE 9** Questions node screens with (a) Arabic stored questions and (b) English stored questions
node along with its sender, in addition to the receiving date and time. The 50% value was selected based on practical observations by noticing that the queries with this accuracy rate or less matching questions do not usually receive logical answers to them. Next, when the Chabot database node is checked by the app administrator, each of the queries with no match will be processed by forwarding it to the convenient employee to be answered. Then, all of the newly processed queries will be added to the database Questions node in the form of stored questions along with their answers, so that the chatbot will be able to respond accurately to them the next time they are being received as queries. This feature is called the context saving and it adds a lot of reliability to the chatbot. It is worth mentioning here that we have tried in our work to cover all the expected queries the students may have through the intensive data collection period. However, there is always a chance for receiving a query with no match. We believe that with the continuous usage of SIAAA-C, almost all possible queries will be answered accurately and the chance of getting a no match query will be very rare. Actually, our latest

**Code 1  Maps activity**

```java
1. public void onMapReady(GoogleMap googleMap) {
2.   mMap = googleMap;
3.   //Add a marker in Sydney and move the camera
4.   LatLng sydney = new LatLng(32.015693, 35.869588);
5.   mMap.addMarker(new MarkerOptions().position(sydney).title("University of Jordan"));
6.   mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 18));
7.   //Check Permission
8.   if (Build.VERSION.SDK_INT > Build.VERSION_CODES.M) {
9.       if (checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
10.          //Set my location button enabled
11.             mMap.setMyLocationEnabled(true);
12.             //Check and update the user location
13.             LocationManager manager = (LocationManager) getSystemService(LOCATION_SERVICE);
14.             if (manager!= null) {
15.                 manager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5, 5, new
16.                     LocationListener() {
17.                         @Override
18.                         public void onLocationChanged(Location location)
19.                         LatLng newLocation = new LatLng(location.getLatitude(),
20.                             location.getLongitude());
21.                         //mMap.addMarker(new MarkerOptions().position(newLocation).title("Me"));
22.                         mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(newLocation, 18));
23.                     });
24.                     @Override
25.                     public void onStatusChanged(String provider, int status, Bundle extras) {
26.                         }
27.                     @Override
28.                     public void onProviderEnabled(String provider) {}
29.                     @Override
30.                     public void onProviderDisabled(String provider)
31.                     {}};
32. } else {
33.     Toast.makeText(this, "Manager is null", Toast.LENGTH_SHORT).show();};
34. } else {
35.     requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
36.         LOCATION_CONST);}}
```
tests gained a 92% correct response rate for incoming queries in both languages after receiving more than 250 irredundant questions. Figure 9a,b show two screens for the database Questions node with a number of stored questions in both Arabic and English, respectively. As we can notice in the figure, each of the Arabic stored questions is shown along with a red-bordered English translation that is added for the reader of this paper, which is the approach we adopted in all Arabic content figures.

Finally, code sample 4 shows how the chatbot answers a student’s query. When the chatbot gets a question, it determines first if it is written in Arabic or English by comparing the characters with Unicode. Then, the chatbot calculates the accuracy of the question as explained earlier and compares it with the stored questions in the database. Finally, the answer of the stored question with the highest accuracy is sent to the student as a response to the query.

4 | SIAAA-C RUN SCENARIO

In this section, we will present a detailed app description in the form of a realistic run scenario to show all the features included in SIAAA-C. Upon a successful logging in process by a student called Ahmad, for example, the home screen will appear, as shown in Figure 11a. This screen has four main categories: Schedule, Campus map, Chatbot, and Settings. Selecting any of these categories will lead naturally to a new screen with more details related to it. If Ahmad selects the settings category, for example, the Settings screen in Figure 11b will appear, which in turn has four subcategories to select from.

Selecting the personal information subcategory in the Settings screen shows the student’s Personal Information screen. Figure 12 shows Ahmad personal information that includes his full name, Student ID as given by the admission and registration unit, and the faculty in which the student is enrolled in, such as the school of engineering in this case. Moreover, this screen also shows the official UoJ email of the student along with his current GPA. The screen also shows that Ahmad is a student at the computer engineering department and that the instructor who is assigned to him as an academic supervisor or counselor is Dr. Rami Malik.

SIAAA-C, like other apps, has the feature of sending different types of notifications to the users to inform them about urgent news and remind them about important dates. Figure 13 shows the Notifications screen, which includes four different types of notifications a student can select to receive or not. As we can notice from the figure, the first type of notification is mandatory and cannot be disabled due to its importance, unlike the other three that can be selected by the student if he/she likes, with a strong recommendation to select all of them. Ahmad in this scenario decides to enable both the exams reminder and the lectures reminder and ignore the absence reminder. The first type of notification, which is Announcements, includes miscellaneous announcements that are officially declared by the student’s department head and cover different academic topics. This type of notification may include a wide range of

3.4 | Code samples of important components

SIAAA-C was written in JAVA where the total system was divided into smaller parts, which in turn were also divided into even smaller parts. This approach of developing a new app is very practical, as it saves effort and time during the programming phase. In fact, building one small component at a time has helped us to easily test our codes for different types of errors in a short time. Moreover, we were able to eliminate redundant codes by reusing the same component multiple times with a minimal required modification. We present in this section four code samples that represent some of the most important components in our app with a general description of each of them.

Code sample 1 shows the process of taking the current location for a student using google maps services by adding two variables (X, Y) to represent the accurate location. The default map location is UoJ campus center, which usually reduces the time needed to locate any landmark or building inside or near the campus. The current location is automatically updated every 5 min and every 5 m walked to guarantee accuracy. Moreover, the student can zoom the map for a more detailed vision. Naturally, these services will be available when the user agrees to give the permission to access his/her current location.

Code sample 2 describes how the chatbot creates a new semester course schedule in response to a related query made by a specific student. First, the chatbot asks the student about current GPA, Year, and the number of successfully passed credit hours. Then, the gained answers for the previous questions are used to select the approach of selecting courses, as explained earlier.

Code sample 3 shows the process of viewing the course schedule for current semester. The application uses a student username to reach the database node User, which has all his/her information including the current semester course schedule. The viewed schedule contains full details about each course, as shown by Figure 10.
Code 2  schedule activity

1. if (title.equals("Year")) {
2.  scheduleInputTextYear = input.getText().toString();
3.  createDialog("GPA", "Please enter your gpa");
4. } else if (title.equals("GPA")) {
5.  scheduleInputTextGPA = input.getText().toString();
6.  createDialog("Hours", "Please enter your taken hours");
7. } else if (title.equals("Hours")) {
8.  scheduleInputTextHours = input.getText().toString();
9.  //Generate the schedule
10.  if (Double.parseDouble(scheduleInputTextGPA) >= 3.0) {
11.   if (Integer.parseInt(scheduleInputTextHours) > 90) {
12.     for (SemesterSchedule schedule: wantedToUser) {
13.       if (schedule.getNumber() >= 3) {
14.         toOutput.add(schedule);}}}
15.   else if (Integer.parseInt(scheduleInputTextHours) >= 60 &&
16.     Integer.parseInt(scheduleInputTextHours) <= 90) {
17.     for (SemesterSchedule schedule: wantedToUser) {
18.       if (schedule.getNumber() == 1 || schedule.getNumber() == 4) {
19.         toOutput.add(schedule);}}}
20.   else if (Integer.parseInt(scheduleInputTextHours) >= 40 &&
21.     Integer.parseInt(scheduleInputTextHours) < 60) {
22.     for (SemesterSchedule schedule: wantedToUser) {
23.       if (schedule.getNumber() > 1 && schedule.getNumber() <= 2) {
24.         toOutput.add(schedule);}}}
25.   else if (Integer.parseInt(scheduleInputTextHours) >= 20 &&
26.     Integer.parseInt(scheduleInputTextHours) < 40) {
27.     for (SemesterSchedule schedule: wantedToUser) {
28.       if (schedule.getNumber() > 0 && schedule.getNumber() <= 1) {
29.         toOutput.add(schedule);}}}}
30.  else if (Double.parseDouble(scheduleInputTextGPA) >= 2.0 &&
31. Double.parseDouble(scheduleInputTextGPA) < 3.0) {
32.  if (Integer.parseInt(scheduleInputTextHours) >= 90) {
33.  for (SemesterSchedule schedule: wantedToUser) {
34.  if (schedule.getNumber() > 2 && schedule.getNumber() <= 3) {
35.     toOutput.add(schedule);}}}
36.  else if (Integer.parseInt(scheduleInputTextHours) >= 60 &&
37. Integer.parseInt(scheduleInputTextHours) < 90) {
38.  for (SemesterSchedule schedule: wantedToUser) {
39.  if (schedule.getNumber() > 2 && schedule.getNumber() <= 3) {
40.     toOutput.add(schedule);}}}
41.  else if (Integer.parseInt(scheduleInputTextHours) >= 40 &&
42. Integer.parseInt(scheduleInputTextHours) < 60) {
43.  for (SemesterSchedule schedule: wantedToUser) {
44.  if (schedule.getNumber() > 1 && schedule.getNumber() <= 2) {
45.    toOutput.add(schedule);}}}
46.  else if (Integer.parseInt(scheduleInputTextHours) >= 20 &&
47. Integer.parseInt(scheduleInputTextHours) < 40) {
48.  for (SemesterSchedule schedule: wantedToUser) {
49.  if (schedule.getNumber() > 0 && schedule.getNumber() <= 1) {
50.    toOutput.add(schedule);}}}
50. toOutput.add(schedule);}}}}
51. else {
52. //Less than 2.0
53. if (Integer.parseInt(scheduleInputTextHours) >= 90) {
54. for (SemesterSchedule schedule: wantedToUser) {
55. if (schedule.getNumber() == 0 | | schedule.getNumber() == 1) {
56. toOutput.add(schedule);}}}}
57. else if (Integer.parseInt(scheduleInputTextHours) >= 60 &&
58. Integer.parseInt(scheduleInputTextHours) < 90) {
59. for (SemesterSchedule schedule: wantedToUser) {
60. if (schedule.getNumber() > 1 && schedule.getNumber() <= 0) {
61. toOutput.add(schedule);}}}
62. else if (Integer.parseInt(scheduleInputTextHours) >= 40 &&
63. Integer.parseInt(scheduleInputTextHours) < 60) {
64. for (SemesterSchedule schedule: wantedToUser) {
65. if (schedule.getNumber() > 1 && schedule.getNumber() <= 0) {
66. toOutput.add(schedule);}}}
67. else if (Integer.parseInt(scheduleInputTextHours) >= 20 &&
68. Integer.parseInt(scheduleInputTextHours) < 40) {
69. for (SemesterSchedule schedule: wantedToUser) {
70. if (schedule.getNumber() > 1 && schedule.getNumber() <= 0) {
71. toOutput.add(schedule);}}}}
72. String toMessage = "";
73. int number=1;
74. int counter = 0;
75. for (SemesterSchedule s1: toOutput) {
76. toMessage = toMessage.concat(number + "," +"Course Number: " + s1.getCourseNumber()+
77. "\n"+ "Course Name: " +s1.getCourseName()+ "\n" +"Days: " + s1.getDays()+ "\n"+"Time: " +
78. s1.getTime() + "\n"+ "Credit hours: " + s1.getCreditHours()+ "\n" +"\n"");
79. sendRes(toMessage, counter + +);
80. number + +;}}});
81. builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
82. @Override
83. public void onClick(DialogInterface dialog, int which) {
84. dialog.cancel();}});
85. builder.show();
86. //If it does exists on the database
87. //Or it have an accuracy of 60 >
88. private void sendRes(String message, int counter) {
89. if (counter < 2) {
90. //Toast.makeText(this, String.valueOf(counter) + " = " + message,
91. Toast.LENGTH_SHORT).show();
92. return;}
93. //Toast.makeText(this, "Sending Message", Toast.LENGTH_SHORT).show();
94. //Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
95. newChat = new Chat(message, new Date().toString(), "chatbot");
96. chats.add(newChat);
97. reference.child("Chatbot").child(auth.getCurrentUser().getEmail().split("@")[0]).
98. setValue(chats).addOnCompleteListener(new OnCompleteListener<Void>() {
99. @Override
100. public void onComplete(@NonNull Task<Void> task) {
101. adapter.notifyDataSetChanged();}});}
announcements like informing the students about lecture cancellations, exams date changing, social events, and different types of deadlines. However, the Exams Reminder sends two notifications for each exam, one before 7 days of the exam day and the other one on the same day. Moreover, the Lectures Reminder sends a 10-min early notice before any lecture begins. Finally, the absence reminder sends an alert to the student if he/she was counted as absent in a specific lecture with the number of total absences in the course ten minutes after that lecture ends.

Figure 14 shows an example screen of four notifications that were sent to Ahmad at different times on May 5, 2020. First, we can also notice that an exam reminder was received an hour earlier to remind the student of a Computer Organization exam he had to attend at 13:00 on the same day. Moreover, another exam reminder was received before 10 min as an early notice of an upcoming Networks exam on May 12, 2020 to urge Ahmad to start planning and studying for it. Also, a lecture reminder about the Circuits course was received 10 min before the start of the lecture. Finally, the last received notification 1 min ago was an announcement about a Computer Organization lecture cancellation for the current day. Ahmad did not receive any absence notifications, as he had not enabled this option at first place.

The Settings screen also allows the student to change both the interface and notification languages from
English, which is the default, to Arabic and vice versa. Selecting Arabic may be more convenient for local and Arab students who may sometimes prefer to have an Arabic interface, specially the ones who have some weakness in the English language and may find it a challenge to interact with a full English interface, which in turn is more convenient for foreign students. Figure 15a, for example, matches Figure 9a and shows the four categories of SIAAA-C main screen in Arabic. Moreover, Figure 15b shows an x Arabic notification that declares a Digital Electronics lecture cancellation. These screens were taken from a student’s account who had chosen Arabic as a main language. However, the chatbot itself can be interacted with using both English and Arabic queries at the same time, regardless of which language is chosen by the student. Finally, The Useful Links option in the settings screen is simply a collection of useful links that may be useful for students like the official UoJ website and a direct link to the registration page.

However, if Ahmad selects the Schedule category in the main screen, the details of his current semester registered courses will appear, as shown in Figure 10. As we can notice, Ahmad’s current schedule consists of five courses and one laboratory, with 16 credit hours in total. Moreover, each registered course details include the name and number of the course, the days and times of its lectures, the name of the instructor, its assigned classroom, and the number of this course’s credit hours of course.

The campus map category in the main screen will lead to a Google map screen, similar to the one appearing in Figure 16. The screen by default shows the map of UoJ with its main gates and landmarks appearing on it. Moreover, Ahmad will be able to view his current location through the button appearing at the top right corner of the map. Even more, he can also use this map to find the directions to any place he wants inside the campus of UoJ and the surrounding area starting from his current location.

The last and most important category in the main screen is the Chatbot, which allows the students to send different queries and receive responses to them by creating a complete dialogue with it in a similar way as talking with a human. As mentioned earlier, the chatbot responds to both Arabic and English queries to cover both local and foreign students and is trained to respond correctly to a large group of questions regarding different academic topics. Figure 17 shows two screens with sample queries sent by Ahmad to the chatbot and the responses he received to them. As we can notice, our chatbot is built as a part of SIAAA-C, unlike most of
other apps where the chatbot usually uses the Facebook messenger to interact with the user. Figure 17a shows an example of two queries written in English by Ahmad to ask about Dr. Mohammad Yazan office hours and

```java
Code 4 Chatbot activity

1. public static boolean textContainsArabic(String text) {
2.     for (char charac: text.toCharArray()) {
3.         if (Character.UnicodeBlock.of(charac) ==
4.             Character.UnicodeBlock.ARABIC) {
5.             return true;}};
6.     return false;}
7. 
8. private void checkIfTheresAnswer(final String message) {
9.     // If arabic or English
10.    String child = "English";
11.    if (textContainsArabic(message)) {
12.        // Toast.makeText(this, "Arabic", Toast.LENGTH_SHORT).show();
13.        child = "Arabic";
14.    } else {
15.        child = "English";
16.    }
17.    reference.child("Questions").child(child).
18.        addListenerForSingleValueEvent(new ValueEventListener() {
19.            @Override
20.            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
21.                questions.clear();
22.                for (DataSnapshot snapshot: dataSnapshot.getChildren()) {
23.                    Question question = snapshot.getValue(Question.class);
24.                    for (String questionWord: question.getQuestionName().split(" ")) {
25.                        for (String messageWord: message.split(" ")) {
26.                            if (questionWord.contains(messageWord)) {
27.                                question.addAccuracy();}}}
28.                questions.add(question);
29.                Question oldQuestion = null;
30.                for (Question question: questions) {
31.                    if (oldQuestion == null) {
32.                        oldQuestion = question;
33.                    } else {
34.                        if (oldQuestion.getAccuracy() < question.getAccuracy()) {
35.                            oldQuestion = question;}}
36.                    if (oldQuestion != null) {
37.                        chats.addAll(oldQuestion.getChatArrayList());
38.                        reference.child("Chatbot").child(auth.getCurrentUser().getEmail().split("@")[0]).
39.                            setValue(chats);}}
```

FIGURE 11 (a) SIAAA-C main screen with four main categories and (b) settings screen

FIGURE 12 Student Ahmad’s personal information screen
After matching each query with the stored database questions, as explained earlier, the chatbot responds by giving the office hours of the first instructor and the email of the second one. However, Figure 17b represents two queries written in Arabic in which Ahmad asks about the nearest bookstore to his current location and the location of classroom 001 in the computer engineering department. The chatbot answers both queries in Arabic as being asked, giving Ahmad the names of two near bookstores inside and outside UoJ campus in addition to the exact location of the required classroom.

It is worth mentioning here that these queries have all been compared with the questions stored in the chatbot database and matched with the highest accuracy rate for each of them, as explained earlier. In the case of having no match to any query, a message will inform the student of having no answer and the query will be saved to be processed later. Figure 18 shows two queries asking about the email and office location for an instructor called Mousa Kamel. One of the queries was written in English and the other one in Arabic and both did not have any matching stored questions, and hence the response messages said “Sorry. I don’t have an answer to this question” and “أعتذر. لا أملك إجابة لهذا السؤال.” in both languages. Actually, the reason of not having a match is that either Ahmad has written a wrong name or the instructor’s information has not been added to the database yet. In case of the second reason, the next time the database is checked for unanswered queries, this and other unanswered queries will be processed and added to the database. As a result, the next time a student asks about Dr. Mousa Kamel information, a matching stored question will be found and a valid response message will be sent.

Actually, the reason of not having a match is that either Ahmad has written a wrong name or the instructor’s information has not been added to the database yet. In case of the second reason, the next time the database is checked for unanswered queries, this and other unanswered queries will be processed and added to the database. As a result, the next time a student asks about Dr. Mousa Kamel information, a matching stored question will be found and a valid response message will be sent.

The previous examples represent static query samples and have the same response to all students. However, asking the chatbot to suggest a new semester course schedule is a nonstatic query, which depends on the student’s situation and may require gathering additional information before giving a valid response, as explained earlier. Here we will preview three examples for three students with different academic situations and how the chatbot responds to each of them.

At the end of the semester, Ahmad was worried because his current GPA was less than 2.0, which would lead him to get hard academic penalties according to UoJ regulations. However, the number of the successfully passed credit hours was normal. When Ahmad asks the
chatbot to suggest a course schedule, it responds by asking him to enter his current academic year, GPA, and the number of successfully passed credit hours. Depending on Ahmad's answers, the chatbot suggests the schedule appearing in Figure 19, which consists of 13 credit hours only. The proposed schedule includes three previously failed courses, one elective course, and one laboratory only. The reason for suggesting this schedule is obviously to help Ahmad raise his GPA over 2.0 through replacing the low grades gained in failed courses with the repetition ones. Moreover, trying to get a high grade in the elective course is usually a doable task, which justifies suggesting it in the proposed schedule. However, reducing the number of credit hours to 13 in this schedule aims to help Ahmad concentrate efforts on raising his GPA to avoid any academic penalties. For receiving the lowest GPA, the least number of credit hours are suggested by the chatbot.

Another student with the same query but with a different academic situation will naturally receive a different course schedule suggestion, as we have explained earlier in Section 3. Hussam, for example, asks an advice for next semester course schedule because his total credit hours were beyond normal, but his GPA is higher than 2.0. As a result, the chatbot recommends the maximum allowed number of credit hours, which is 18, in UoJ regulations. Figure 20 shows Hassam's suggested schedule that consists of five courses and three laboratories. Although such a schedule may require a lot of work,
it allows Hussam to increase his total passed credit hours fast and reduce the gap with his colleagues in the same academic year.

Finally, a third student, Abeer, has a good GPA and her total successfully passed credit hours were normal as a representation of the majority of students who have no academic problems. When she asks the chatbot for a suggested course schedule based on her academic situation, it proposes a schedule based on the official department indicative plan, as appearing in Figure 21. In this case, the suggested schedule consists of 16 credit hours including five courses and one credit hour laboratory, which is a logical semester load for a normal student. In the three previously discussed examples, the proposed courses are selected by excluding the previously passed ones first and then choosing the remaining courses in the indicative plan starting from the first academic year fall semester until the suggested load of credit hours is filled.

5 | TESTING AND SURVEY RESULTS

The best way to efficiently evaluate any app is to practically apply it in a real environment in which all expected types of users are included and all possible inputs are tested to get a realistic assessment. In fact, all the features provided by SIAAA-C have been carefully chosen based on the feedbacks we gained from a random group of UoJ students whom we asked to fill a description form we prepared to define a useful students’ assistant app. Actually, these feedbacks were very useful to us during the design phase to come out with an initial version of the app. After that, SIAAA-C has been modified multiple times depending on the continuous responses gained by small test groups, reaching the final version described in this paper. Moreover, SIAAA-C was launched in UoJ for a 1-month test period during the second semester of the academic year 2019/2020. Actually, the students’ responses to SIAAA-C were very encouraging, which has motivated us to authenticate them in a statistical way in the form of an assessment survey. In fact, the survey we prepared consisted of 13 questions that measure different aspects regarding students’ assistant apps in general and SIAAA-C in particular. Moreover, the survey was distributed 2 weeks after launching the final app version to give the students a sufficient amount of time to test all the features included in it and get realistic answers to the included questions at the end of the test period. The survey was filled using MS forms by 102 students distributed over five different faculties, as revealed by the numbers in Table 2.

It worth mentioning here that our testing and assessment process was executed during the COVID-19 pandemic, which caused a complete lockdown in Jordan and most countries around the world as well. As a result,
the UoJ campus was closed for the majority of 2019–2020 second semester, which had both positive and negative sides regarding our work. On the one hand, closing of UoJ campus has affected our SIAAA-C test plans and reduced the test period length and the total number of app users along with the number of survey participants as well. On the other hand, the pandemic has created an additional motive for our work and noticeably increased the importance of providing an automated tool that is available round the clock to answer different students’ queries and reduce the enormous load on the administrative staff. Interestingly, the analysis of the survey has revealed a high degree of similarity in the answers gained by different faculties’ students. This observation may suggest that UoJ students in general have a good knowledge of technology, which makes SIAAA-C a very useful tool for the majority of them, regardless to their specialties. Moreover, this similarity may also suggest that our app is a user-friendly tool and can be used without any problems. Actually, having both Arabic and English interfaces may be a reason for the last conclusion, because language will not be an obstacle for users any more.

Table 3, however, shows the survey results from which we can conclude that the feedbacks were very positive in general and most of the users were satisfied with the features included in the app and its general design. In more detail, we can notice that four in every five students have used one university app at least before SIAAA-C according to Question 1, which shows that most of our participants have a previous experience in this area. Moreover, Question 2 reveals that less than 30% of our test group participants have said that the university apps they used before were useful to them, which reflects a low degree of satisfaction. However, the concept of chatbot is considered familiar to 63.7% of our participants, which may be justified with the limited usage of this component in many daily life apps. Questions 4 shows that the majority of students prefer to interact with a chatbot in Arabic, which implies that our choice to provide SIAAA-C and its chatbot in both Arabic and English languages was logical and justified. Moreover, Question 5 has revealed that the majority of our test
sample has considered the notifications feature in SIAAA-C as a useful addition in general with a debatable degree of satisfaction among them. One of the most pleasant results was that 74.5% of students found the course schedule suggested by the chatbot useful to them (question 6), because it is considered as one of the most important features included in this app. Moreover, Question 7 shows also a high satisfaction degree regarding static queries responses in general. Actually, we believe that with continuous SIAAA-C usage and updating the database regularly, the users will be more and more satisfied with the responses they receive for different queries.

In a similar way, Question 8 shows that most students are very satisfied with both the campus map included in the app and the chatbot responses to different locations queries. However, Question 9 reflects mostly positive reviews about the app’s design in terms of used colors and fonts in addition to its categories, which in turn reflects positively on the general app evaluation. Question 10 suggests that SIAAA-C is a user-friendly app with more than 90% of students having no problems at all while using it during the test period. Interestingly, Question 11 suggests that the importance of SIAAA-C...
and similar apps has remarkably increased during the pandemic based on the opinion of 88.23% of participants. This result was not a surprise to us, as we planned from the beginning to present a human interaction-free tool that is able to provide instant and accurate services to students and reduce the load on administrative staff in UoJ. In fact, the previous result may be strongly related to question 12 result, which shows that more than 80% of the survey participants have preferred SIAAA-C chatbot responses to different queries over the human ones in terms of time and accuracy during UoJ campus closing period due to COVID-19 pandemic.

The last question in the survey was a general review area in which students are given a chance to evaluate SIAAA-C in their own words and to provide us with any useful suggestions they may have. Actually, we have gained miscellaneous responses to question 13, mostly positive, all helpful, with a number of suggestions to be considered in the future to enhance the performance of SIAAA-C and improve its overall design even more. Table 4 shows three random samples for students’ answers to question 13, reflecting different opinions about SIAAA-C.

### 6 CONCLUSION AND FUTURE WORK

Smartphones applications can be used to help people improve their lives in different fields including higher education. SIAAA-C is a personal electronic guide that was developed to be used by the students of UoJ or any other university to help them getting different academic services. The services included in the app include showing the personal information of a student along with the current schedule, enabling different types of notifications as reminders for lectures, exams, absences, and general announcements, using a map to reach different locations inside or near UoJ campus, and providing a very useful interactive query system in the form of a built in chatbot. The chatbot was built to interact with...
different queries ranging from simple questions like asking about a specific instructor’s email to more complex ones like asking for a suggested course schedule based on the student’s academic situation. Moreover, SIAAA-C has both Arabic and English interfaces and its chatbot can respond to queries written in both languages to make the app usable by local and Arabic students in addition to thousands of foreign students who are enrolled in UoJ. The app was launched to be tested by students in a number of UoJ schools during the second semester of the academic year 2019–2020 when the campus was closed due to COVID-19 pandemic and the feedbacks of most users were very positive. We think that the current worldwide pandemic has confirmed the importance and the necessity of SIAAA-C and similar apps in all higher education institutes around the world. Moreover, we have conducted an evaluation survey that was answered by 102 active users distributed over five faculties and the responses were very encouraging with a number of useful notes to take into consideration. Finally, our future work includes expanding the app by adding new languages, creating iOS and website versions, and contacting UoJ administration to adopt it officially as a student assistant app for UoJ students.

CONFLICT OF INTERESTS
The authors declare that there are no conflict of interests.

ETHICAL STANDARDS STATEMENT
All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. Informed consent was obtained from all individual participants included in the study.

PEER REVIEW
The peer review history for this article is available at https://publons.com/publon/10.1002/cae.22419.

DATA AVAILABILITY STATEMENT
Data sharing is not applicable to this article, as no data sets were generated or analyzed during the current study.

ORCID
Khalid A. Darabkh  http://orcid.org/0000-0002-0362-1975

REFERENCES
1. B. Abu-Hamour, Faculty attitudes toward students with disabilities in a Public University in Jordan, Int. Educ. Stud. 6 (2013), no. 12, 74–81.
2. Y. Bai, L. Yao, T. Wei, F. Tian, D. Y. Jin, L. Chen, and M. Wang, Presumed asymptomatic carrier transmission of COVID-19, JAMA 323 (2020), no. 14, 1406–1407.
3. J. Bourgonjon, M. Valcke, R. Soetaert, and T. Schellens, Students’ perceptions about the use of video games in the classroom, Comput. Educ. 54 (2010), no. 4, 1145–1156.
4. J. Breuer and G. Bente, Why so serious? On the relation of serious games and learning, J. Comput. Game Cult. 4 (2010), 7–24.
5. C. Checho, The Effects of Podcasting on Learning and Motivation: A Mixed Method Study of At-risk High School Students, University of Nevada, Reno, 2007.
6. C. Craig, and A. Gerber, Learn Android Studio: Build Android Apps Quickly and Effectively, Apress, 2015.
7. K. A. Darabkh, F. H. Alturk, and S. Z. Sweidan, VRCDEA-TCS: 3D virtual reality cooperative drawing educational application with textual chatting system, Comput. Appl. Eng. Educ. 26 (2018), no. 5, 1677–1698.
8. K. A. Darabkh, L. Haddad, and S. Sweidan, A modified speech recognition algorithm for people with physical disabilities, Proceedings of 2017 IEEE European Conference on Electrical Engineering & Computer Science, Bern, Switzerland, 2017, pp. 23–27.
9. A. De Jonge, Essential App Engine: Building High-performance Java Apps With Google App Engine, Addison-Wesley, 2011.
10. M. Dibitonto, K. Leszczynska, F. Tazzi, and C. M. Medaglia, Chatbot in a campus environment: Design of LiSA, a virtual assistant to help students in their university life, International Conference on Human-Computer Interaction, Springer, Cham, 2018, pp. 103–116.
11. D. M. Fuller and P. G. Hinegardner, Ensuring quality website redesign: The University of Maryland’s experience, Bull. Med. Libr. Assoc. 89 (2001), no. 4, 339.
12. A. M. Hamdan-Mansour and H. A. Dawani, Social support and stress among university students in Jordan, Int. J. Mental Health Addict. 6 (2008), no. 3, 442–450.
13. A. M. Hamdan-Mansour, J. O. Halabi, and H. A. Dawani, Depression, hostility, and substance use among university students in Jordan, Mental Health Subst. Use Dual Diagn. 2 (2009), no. 1, 52–63.
14. A. F. Khalifeh, K. A. Darabkh, and A. Kamel, Performance evaluation of voice-controlled online systems. *International Multi-Conference on Systems Signals & Devices Chemnitz*, Germany, 2012, pp. 1–6.

15. A. Kukulska-Hulme, *Will mobile learning change language learning?* ReCALL 21 (2009), no. 2, 157–165.

16. P. Leijdekkers, V. Gay, and F. Wong, CaptureMyEmotion: A mobile app to improve emotion learning for autistic children using sensors, *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, IEEE, 2013, pp. 381–384.

17. K. B. Lofland, *The use of technology in the treatment of autism*, Technology and the Treatment of Children with Autism Spectrum Disorder, Springer, Cham, 2016, pp. 27–35.

18. E. Mansour, *Use of smartphone apps among library and information science students at South Valley University*, Egypt, Int. J. Internet Educ. 15 (2016), no. 1, 30–62.

19. L. Moroney, *Definitive Guide to Firebase*, Apress, 2017.

20. PSUT, 2019, available at https://www.psut.edu.jo/

21. E. A. Qaralleh and K. A. Darabkh, *A new method for teaching microprocessors course using emulation*, Comput. Appl. Eng. Educ. 23 (2015), no. 3, 455–463.

22. Research Gate, 2020, available at https://www.researchgate.net/figure/Graphical-statistics-ofsmartphoneusers_fig1_321224582

23. Sharjah, 2020, available at https://www.sharjah.ac.ae/en/Pages/default.aspx/

24. M. Sharples, J. Taylor, and G. Vavoula, *Towards a theory of mobile learning*, Proc. MLearn 1 (2005), 1–9.

25. SJSU, 2020, available at http://www.sjsu.edu/

26. N. Smyth, *Android Studio 2.2 Development Essentials-Android*, 7th ed., Payload Media, 2016.

27. N. Smyth, *Firebase Essentials-Android Edition*, Payload Media, Inc., 2017.

28. G. Stockwell, *Investigating learner preparedness for and usage patterns of mobile learning*, ReCALL. 20 (2008), no. 3, 253–270.

29. B. Stonehem, *Google Android Firebase: Learning the Basics*, vol. 1, First Rank Publishing, 2016.

30. S. Z. Sweidan and K. A. Darabkh, *A new efficient assembly language teaching aid for intel processors*, Comput. Appl. Eng. Educ. 23 (2015), no. 2, 217–238.

31. S. Z. Sweidan and K. A. Darabkh, *VREG: A virtual reality educational game with arabic content using android smartphones*, J. Software Eng. Appl. 11 (2018), no. 10, 500–520.

32. S. Z. Sweidan, R. Saifan, K. A. Darabkh, S. Abu-Kaff, and S. Al-Ali, *Kids’ tracker: An android application for tracking children*, J. Software Eng. Appl. 10 (2017), no. 13, 907–924.

33. S. Z. Sweidan, H. Salameh, R. Zakarneh, and K. A. Darabkh, *Autistic Innovative Assistant (AIA): An Android application for Arabic autism children*, Interact. Learn. Environ. (2019), 1–24.

34. K. W. Tracy, *Mobile application development experiences on Apple’s iOS and Android OS*, IEEE Potentials. 31 (2012), no. 4, 30–34.

35. University of Jordan, 2020, available at http://www.ju.edu.jo/Home.aspx

36. H. Van der Merwe, B. Van der Merwe, and W. Visser, *Verifying android applications using Java Pathfinder*, ACM SIGSOFT Softw. Eng. Notes. 37 (2012), no. 6, 1–5.

37. M. Van Drongelen, *Android Studio Cookbook*, Packt Publishing, Birmingham, 2015.

38. WHO, 2020, available at https://covid19.who.int/

39. WHO, 2020, available at https://covid19.who.int/?gclid=CjwKCAjw97P5BRBQEiwAGfV6w9CtWMfjo6p_cLrklXZ2ZTvz8FA_W2nqfRuFTEyxpXE09N7mZCwddRoCnb8QAyVd_BwE

40. A. R. Wiratno and K. Hastuti, *Implementation of firebase real-time database to track BRT trans semarang*, Scientific J. Inform. 4 (2017), no. 2, 95–103.

41. M. Wolfson, and D. Felker, *Android developer tools essentials: Android Studio to Zipalign*, O’Reilly, Sebastopol, 2013.

42. Ying-Ying Zheng, Yi-Tong Ma, Jin-Ying Zhang, and Xiang Xie, *COVID-19 and the cardiovascular system*, Nat. Rev. Cardiol. 17 (2020), no. 5, 259–260.

**AUTHOR BIOGRAPHIES**

**Saadeh Z. Sweidan** received his Master degree of science in Computer Engineering from Jordan University for Science and Technology, Irbid, Jordan, in 2007. He is currently a full-time instructor in Computer Engineering Department at the University of Jordan, Amman, Jordan. His research interests include hardware design and implementation, digital arithmetic, cryptography algorithms, and assembly simulators.

**Sarah S. Abu Laban** received her BSc degree in Computer Engineering from University of Jordan Amman, Jordan, in 2020. Her research interests include interactive leaning environments, software design and implementation, and E-learning.

**Njood A. Alnaimat** received her BSc degree in Computer Engineering from University of Jordan Amman, Jordan, in 2020. Her research interests include interactive learning environments, software design and implementation, and computer networks.

**Khalid A. Darabkh** received the PhD degree in Computer Engineering from the University of Alabama in Huntsville (UAH), USA, in 2007 with honors. He was selected for inclusion in the Who’s Who Among Students in American Universities and Colleges. He was honored by UAH for being not only so outstanding, but also finishing the PhD requirements along with publishing a massive
number of reputable scientific articles in record time. He has joined the Computer Engineering Department at the University of Jordan as an Assistant Professor since 2007 and obtained a distinguished professorship in 2016. He authored and coauthored of at least a 170 highly esteemed research articles. He is among World’s Top 2% Scientists List, for various disciplines, compiled by Stanford University in 2020. Prof. Darabkh is the recipient of 2020 Federation of Arab Scientific Research Councils Reward—Theme of Innovation. Prof. Darabkh is further the recipient of Ali Mango Distinguished Researcher Reward for Scientific Colleges and Research Centers in Jordan. He is among Top 5 JU Researcher’s List who published the highest number of quality manuscripts in Scopus. Not only to this extent, but rather he is Sun Certified Programmer for Java Platform (SCJP), Oracle Database Administrator Certified Associate (OCA), as well as Oracle Java Developer Certified Trainer (OJDC-T). In fact, he serves on the Editorial Board of Telecommunication Systems, published by Springer, Computer Applications in Engineering Education, published by John Wiley & Sons, and Journal of High Speed Networks, published by IOS Press. In addition, he serves as a TPC member of highly reputable IEEE conferences such as GLOBECOM, LCN, VTC-Fall, PIMRC, ISWCS, ATC, ICT, and IAEAC. Moreover, he is a member of many professional and honorary societies, including Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma XI. He was listed among those of IBC Leading Scientists of the World, Cambridge, England, as well as IBC 2000 Outstanding Scientists, Cambridge, England. He was selected for inclusion in the Marquis Who’s Who in the World. He supervised and examined at least 100 masters’ theses in most of Jordanian universities. With administrative experience at the University of Jordan, he served as Assistant Dean for Computer Affairs in the College of Engineering from Sept 2008 to Sept 2010. In addition, he served as Acting Head of the Computer Engineering Department from June 2010 to Sept 2012. Also, he is engaged in research mainly on Internet of things, Software-defined networks, vehicular networks, flying ad-hoc networks, Fog networking, Full duplex cognitive radio networks, queuing systems and networks, multimedia transmission, channel coding, steganography and watermarking, as well as innovative and interactive learning environments.

How to cite this article: S. Z. Sweidan, S. S. Abu Laban, N. A. Alnaimat, and K. A. Darabkh, SIAAA-C: A student interactive assistant android application with chatbot during COVID-19 pandemic. Comput Appl Eng Educ. (2021), 1–25. https://doi.org/10.1002/cae.22419