A Location-Based Mobile Advertising System for Small-to-Medium Businesses

Ahmed Abdelmoamen Ahmed and Anitha Palusa

Abstract Business owners need more affordable venues to proclaim their services and products. Mobile technologies offer a convenient path for implementing smart location-based advertising (LBA) solutions using intelligent handheld devices. This chapter presents a case study for a mobile-based LBA system that has components running on smartphones as a mobile app and cloud servers as a web application. Google Maps is used in the mobile app as an underlying service for enhancing the experience of mobile users in using the system. Mobile users are able to utilize the mobile app to travel to and from their destinations meanwhile seeing and engaging relevant advertisements, including new job openings located in their local neighborhoods and discounts on their favorite meals. This chapter proposes a novel sensing approach for representing the switching of sensing contexts in the proposed LBA system. The main goal of multi-modal sensing is to decrease the energy consumed from the mobile app by identifying the changes in the current sensing mode automatically, thus avoiding needless sensing overheads. We organized the proposed LBA system into these two components: mobile user and business owner sides. First, the business owners are allowed to initiate new advertisements by entering simple metadata about their advertisements on the business side. Second, mobile users, on the user side, can search the Google Map to see attractive advertisements while doing their daily activities such as driving, bicycling, jogging, walking, or relaxing in their homes. We carried out a set of experimental evaluations to show the performance and scalability of the proposed LBA approach.

Keywords LBA · Multi-modal sensing · Mobile app · Energy-efficient

A. A. Ahmed (✉) · A. Palusa
Computer Science Department, Prairie View A&M University, Prairie View, TX, USA
e-mail: amahmed@pvamu.edu

F. Torres-Guerrero et al. (eds.), *2nd EAI International Conference on Smart Technology*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-031-07670-1_1

1 Introduction

Mobile and intelligent technologies have created an opportunity for small businesses to advertise their products affordably using LBA technologies.Using LBA, the owners of small businesses can customize their ads to potential customers in real time according to their contemporary geolocation. This has taken away the constraints between customers and businesses when they are in close proximity to the target business locations [14].

In this chapter, we present a prototype implementation of a mobile-based LBA solution that various types of businesses can use to promote their services and products in an affordable way. The developed system is organized into components running on mobile devices and a cloud server as a web-based application. We developed a mobile app on the user side to increase the LBA system's user experience. We developed a web application on the business side, allowing business owners to create and manage various types of advertisements via an interactive GUI.

The power efficiency of smartphones becomes a crucial factor in developing mobile apps. In this chapter, we found an opportunity for conserving the energy of the developed mobile app by optimizing the sensing process of mobile devices by modeling its evolving sensing context. We call this approach *context-aware multi-modal sensing* [1]. Utilizing multi-modal sensing, we could represent the evolving sensing needs of mobile apps that can change their behavior based on the changes in context that a mobile device is in real time.

We conducted experimental evaluations on the scalability and power demand when using our LBA solution. The experimental results have demonstrated the energy efficiency and scalability of the proposed multi-modal system.

There are many interesting projects—in academia (e.g., [17, 38]) and industry (e.g., [21, 37])—that involve LBA applications and services in different domains such as monitoring traffic data [20, 37], rescue management [15], preventive health [34, 39], games and entertainment [35], crowdsensing [19], and mobile-based advertising [4, 33].

The proposed approach is more related to research focused on providing support for mobile-based LBA frameworks. The existing work has taken different ways to support LBA applications and services, which focus on the programmability [10] and participatory crowdsensing [22].

Compared to the existing work, many existing platforms for LBA systems are either expensive products or periodically paid services that small businesses cannot afford. Moreover, the existing solutions are designed to study the attitudes of mobile users [14] or analyze the existing business models [18]. Furthermore, they cannot support the simultaneous execution of many LBA services on a single platform, eliminating the opportunities to optimize the overall power consumption by sharing similar sensing requirements among these LBA services [1].

2 Design

As shown in Fig. 1, the proposed LBA architecture is organized into components running on cloud servers (i.e., application and database) and mobile devices such as smartphones and tablets. We used the REST service to coordinate the communication between the two sides.

The proposed multi-modal approach—for modeling the sensing context switching—is implemented at the user side as a mobile app. The multi-modal sensing process could be represented by using a finite-state machine (FSM), which we model as follows:

$$\langle M, \Sigma, \delta, M_0 \rangle, \tag{1}$$

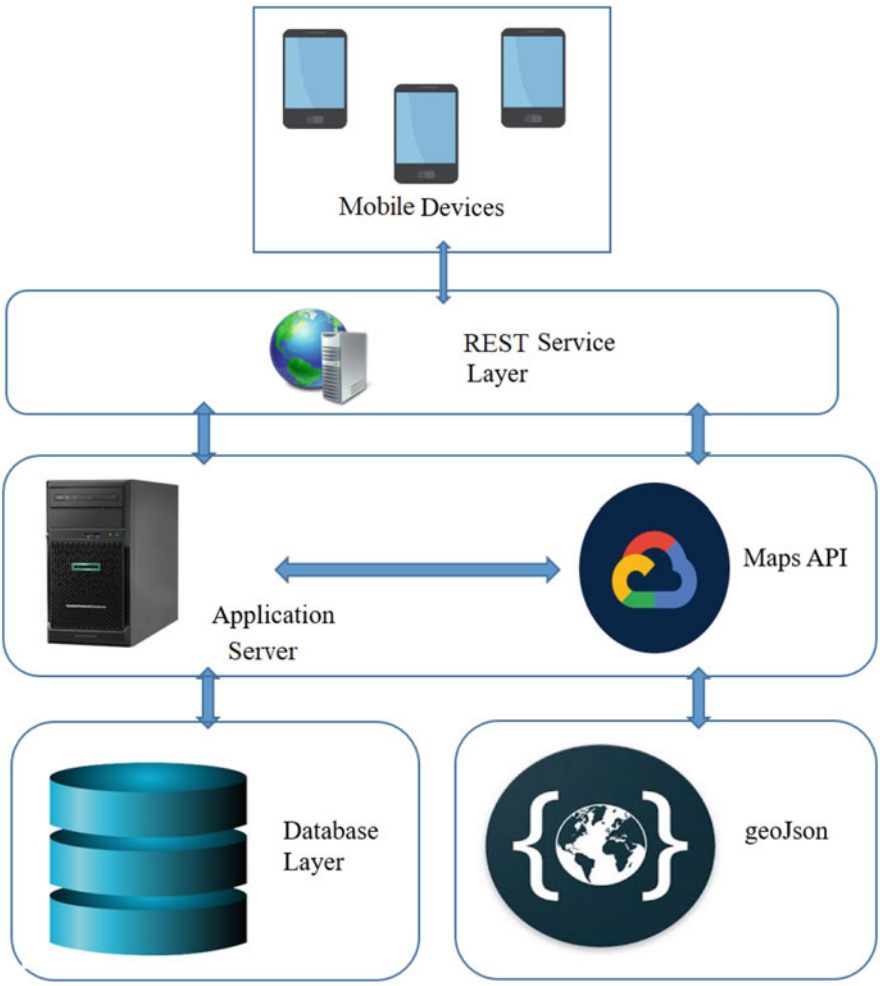

Fig. 1 System architecture

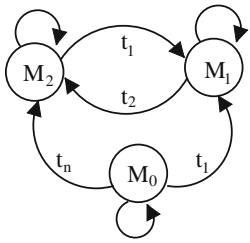

Fig. 2 Multi-modal sensing

where M is a non-empty finite set of sensing modes, Σ is a non-empty finite set of the input sensing data, and $M_0 \in M$ is the initial sensing state.

Every mode $M_j \in M$ models a sensing state (i.e., mode) in which mobile devices behave at a particular point in time. Σ set represents some sensor input readings, a notice of a context switching to another state, or an occurrence of an anticipated event such as getting closer to one location of a business. $\delta : M \times \Sigma \rightarrow M$ is a function that represents a transition from the current state to another one. Σ set represents the newly sensed data that fires the triggers for a state change. Figure 2 illustrates an example of an FSM that has three modes. As shown in the figure, the sensing process of a particular data could fire a trigger $t_i \in \Sigma$ that causes a transition from mode M_i to mode M_j.

The business owner side is implemented as a web application, which uses GeoJSON for modeling the businesses' geolocations. Each GeoJSON object represents some features and attributes of a business, which includes the location coordinates, location size, and location boundaries. Each business is defined by coordinates that shape polygon borders on top of the Google Map.

Business owners can precisely pick out the location coordinates of each business on the map using the developed GUI at the business side. A cloud-based database is used to store these coordinates. When a new business is initiated, the selected coordinates of the drawn polygon are sent to the user side (i.e., mobile app) as a GeoJSON file communicated through the developed REST service.

We developed a python script at the mobile app to parse the incoming GeoJSON files to generate the place polygon coordinates. Then, these coordinates are sent to the mobile app, which uses Google Maps API for displaying the business location on Google Map. An example of one GeoJSON file that is used in our system is shown in Fig. 3. As shown in the figure, each GeoJSON object represents different spatial attributes of the bounded entity of a business location.

3 Implementation

We implemented the business side as a web-based app, which provides an interactive dashboard for businesses to initiate a fresh advertisement, edit an advertisement, display responses and inquiries to their advertisements, and response to inquiries

```
geo_json = {"type": "Feature", "geometry": {"type":
"Polygon", "coordinates":[(30.095319, -95.993581),
(30.096340, -95.993399), (30.096015, -95.991854),
(30.095040, -95.992219), (30.095319, -95.993581)]},
        "properties": {"name": " Pizza Restaurant No 1",
           "styleUrl": "#poly-4F2682-3000-128",
           "styleHash": "-50cd947a",
           "styleMapHash": {
              "normal": "#poly-4F2682-3000-128-normal",
              "highlight": "#poly-4F2682-3000-128-highlight"
           },
           "description": "Pizza Restaurant location",
           "stroke": "#4f2682",
           "stroke-opacity": 1,
           "stroke-width": 3,
           "fill": "#4f2682",
           "fill-opacity": 0.5019607843137255
    }
}
```

Fig. 3 An Example of a GeoJSON file

and requests. We used HTML5, PHP 7.4, CSS3 and JS, MySQL 8.0, in addition to Bootstrap [16] and JSON to develop the business side.

Figure 4a illustrates the registration form for business owners. The forms ask business owners their name, email address, telephone #, and username and password.

Figure 4b illustrates the form that is used to initiate an advertisement on the business side. This form allows businesses to draw the place's polygon using the interactive map for specifying the boundary for their business locations. Business owners need to enter: (1) the name of their businesses that would appear on the mobile side; (2) the type of the created ad such as garage sale, restaurant opening, and job; (3) other attributes, such as description, business photos, time frame, etc.

Figure 4c illustrates the login page at the cloud side for business owners. Similarly, we implemented a login page that utilized the `$_POST` technique to authenticate users. Figure 4d illustrates the existing advertisements list created by a business owner. Owners are also allowed to see the current status of their advertisements and edit and delete the existing advertisements. This page illustrates the title, description, type, time frame, price, address, and geographical business location.

We used the Android ADT bundle Development Environment (64 bit) to implement the mobile app on the user side. The mobile application utilizes various technological tools and programming languages: (1) Android SDK to develop the front-end mobile user activities; (2) PHP 7.4 for implementing the middle-ware between the database server and the mobile app; and (3) MYSQL 8.0 to implement the system database.

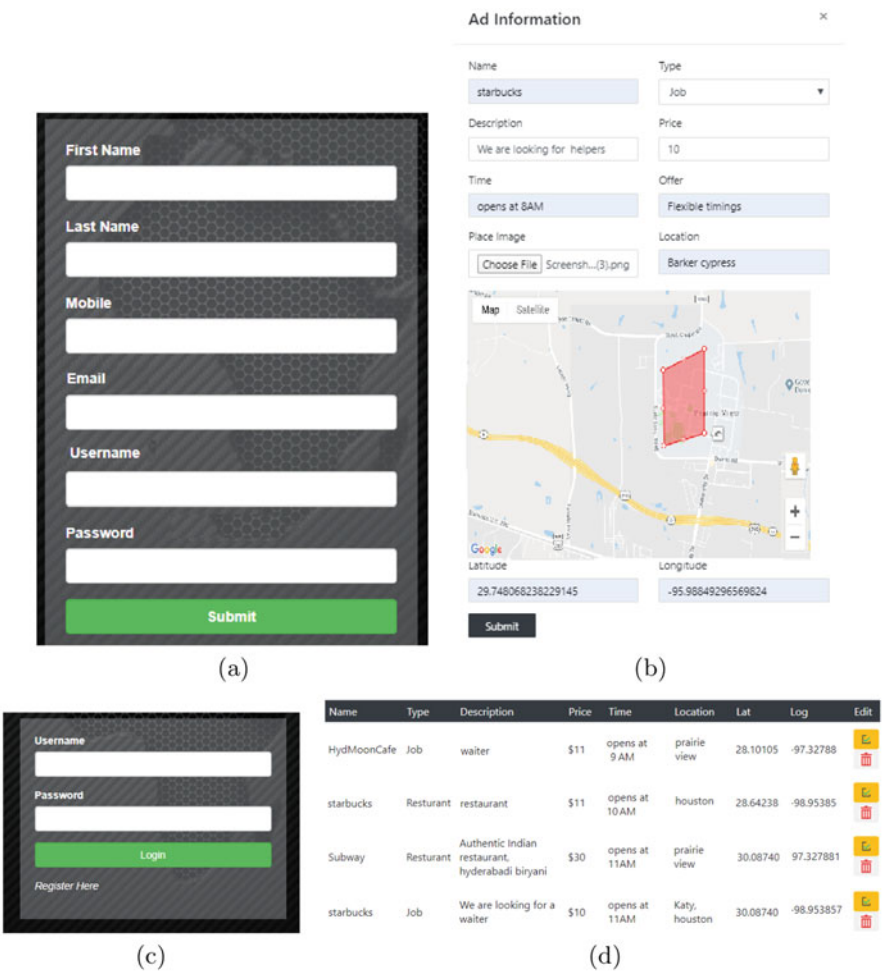

Fig. 4 The implementation of the cloud-based side. (**a**) Registration. (**b**) Creating a new Ad. (**c**) Login. (**d**) List of Ads

Figure 5a illustrates the mobile app landing form. The form has a search `textarea`, which allows mobile users to search for and navigate to any destination provided by the Google Maps API. As shown in Fig. 5b, the autocomplete option is used in the search functionality to increase the mobile user experience using our LBA system.

Users can commute to their target destination by tapping a button titled `START JOURNEY`. The mobile application then creates the path between the mobile's current geolocation and the destination. This process is carried out; meanwhile, the path linking between the start and destination points is being rendered. An example

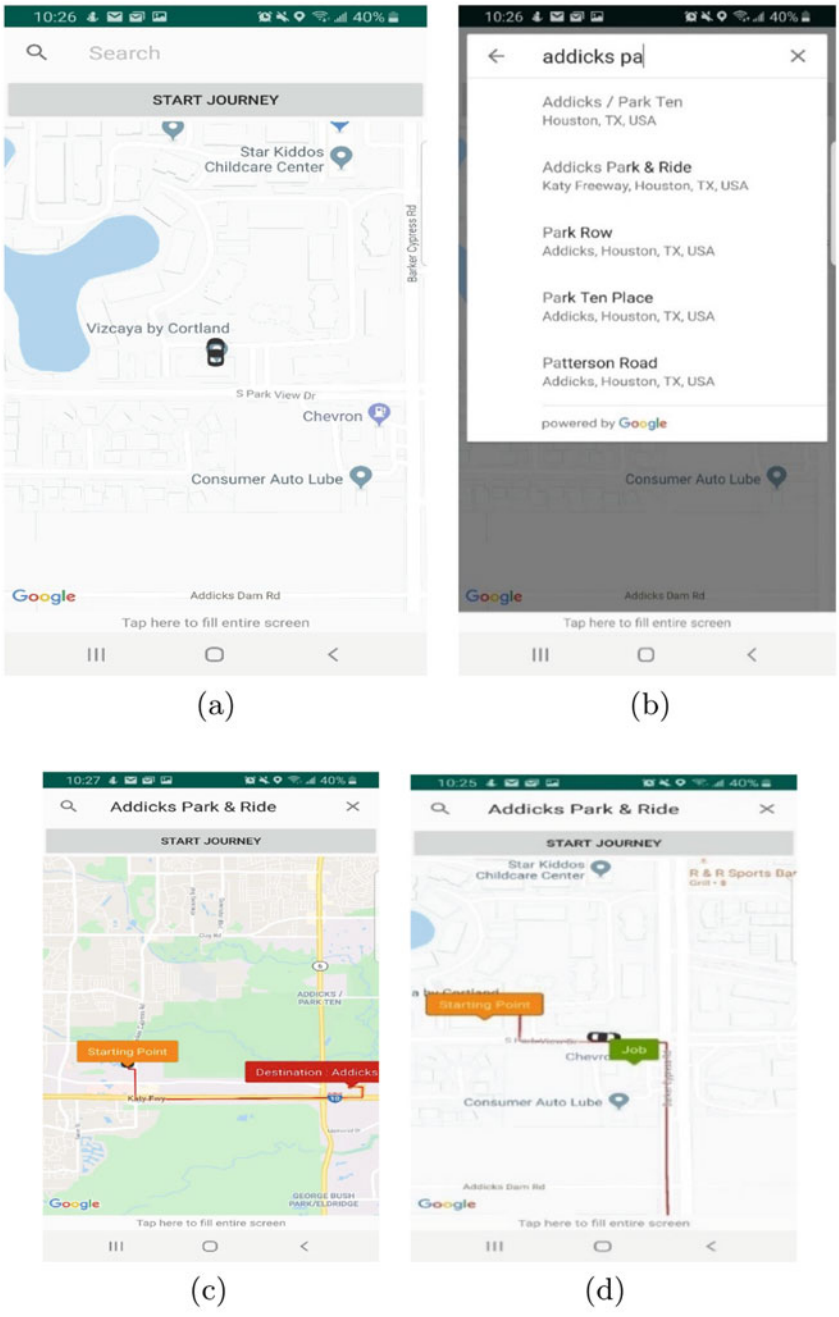

Fig. 5 Screenshots of the mobile user application. (**a**) Landing page. (**b**) Selecting a destination place. (**c**) Displaying the route map. (**d**) Showing an advertisement during the driving state

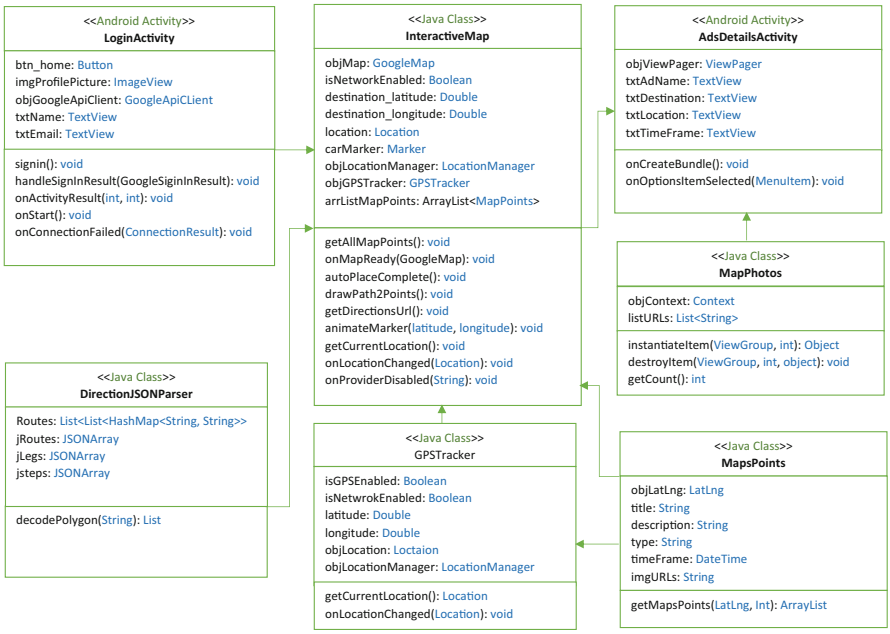

Fig. 6 The class diagram of the mobile side

of this process is illustrated in Fig. 5c where `Addicks / Park Ten` is selected as a target destination.

A screenshot of the mobile app displaying an ad in the driving mode is shown in Fig. 5d. Figure 5d shows different advertisement categories rendered on the app's map. When a mobile user taps on a particular advertisement, the app displays another form that contains more details of this selected advertisement such as photos, time frame, and detailed description of the ad.

Figure 6 shows the class diagram of the system on the mobile side. Two types of class entities are defined in our LBA system, namely Android Activities and Java Classes.

As illustrated in Fig. 6, the `LoginActivity` enables mobile users to login into the system via their Google accounts. Then, they are redirected to the `InteractiveMap` form, using which users will be able to search for a particular location or destination. The `InteractiveMap` class is used to render the path between the current user location and the target location. The `Advertisements- DetailsActivity` represents the mobile form that displays the advertisement information (e.g., photos and detailed description). The `DirectionJSONParser` class is developed for parsing the geolocation of the landmarks and the target location, which Google Maps support. We created the `Routes` list to load the resulting places that were generated by the map. Each item

in `Routes` contains an array of polygon coordinates that states the boundaries of that place location on Google Maps.

We developed the `GPSTracker` class to get the current geolocation of mobile users (i.e., longitude and latitude). This information is stored in the `LatLng` object temporarily before sending it to the database server on the cloud. We developed the `MapPhotos` class to render the advertisement photos on Google Maps. First, the photo URLs are fetched from the database and then sent to the mobile app to display them when the user clicks on an ad. We implemented both `MapsPoints` and `InteractiveMap` classes to be used in rendering the different advertisements on the mobile map. We used JSON to serialize all the details of the selected advertisement, which are communicated between the server and the mobile app. Once the JSON file is received at the mobile app, it will be converted into Java objects using the Retrofit API [36].

4 Evaluation

We conducted a set of experiments to evaluate the performance and scalability of our LBA system. We installed some instrumentations inside the mobile app in order to calculate the CPU processing time that is taken to execute the different processing jobs, including the sensing data collection, context-switching process for the multi-modal approach, acquiring the geolocation coordinates, and making the mobile user trajectories anonymous. Also, a set of instrumentations were developed in the app to calculate the power consumption used by these sensors: gyroscope, GPS, and accelerometer. We carried out every experiment presented in this section for 10 trials. Then, we took and showed the average of the trials' output.

We used a case study for the experimental evaluation. It involves several human activities to mimic various states for the mobile user when viewing advertisements on the map. The mobile application is executed on top of a smartphone running Android 10, which has the following specs: Samsung S8, 4 GB RAM, LTE Category 16, Adreno 540 GPU, and 2.3 GHz Octa-core CPU. We defined 4 activity modes, namely driving, bicycling, walking, and stilling.

Every state (mode) has different requirements in terms of sensors (e.g., accelerometers, gyroscopes, and GPS sensors). The stilling mode requires only the accelerometer data. The walking mode requires both the gyroscope and accelerometer and data. Both the driving and bicycling states require both GPS and accelerometer data.

Additionally, the logic of state transition is mainly based on sensor readings from the three sensors. It is assumed that the three sensors are sampled every one second at a sampling rate of 1 Hz. In other words, the sensors that are needed for the current state (e.g., driving) functionality are sampled at a sampling rate of 1 Hz to detect any triggers for mode transition. For the other sensors, it is required to collect fresh sensor data as described in [29].

4.1 Performance

In order to estimate the scalability of our LBA system on the mobile side, we calculated different processing resources required to create a single ad. Separately, we calculated the ongoing monitoring cost to detect any triggers for state transition. In addition, we measured the processing CPU time and the sensing overheads required to execute the instructions for triggering the state transitions.

We measured processing time for the ongoing process of monitoring the state transition triggers. These processing times are shown in Table 1.

The CPU time (measured in milliseconds) is taken to acquire, process, and examine the three sensor readings to detect any triggers for mode transition. The tag *Old* happens in the case of utilizing the already collected sensor readings for the current state functionality. The tag *New* occurs in the case of collecting new sensor data to detect mode transition triggers only. Both the sampling rate of the mode functionality and evolution was set to be 1 Hz. The time taken for transitioning each sensor from the existing sampling rate to a different rate was calculated to be 6.21 ms (with a standard deviation of 1.21) for the accelerometer sensor, 10.71 ms (with a standard deviation of 1.76) for the gyroscope sensor, and 17.39 ms (with a standard deviation of 2.36) for the GPS sensor.

The overhead costs of making a transition from one mode to another one are displayed in Table 2. These overhead costs include the changes in the sampling rate of the involved sensors and the processing time needed to trigger that change. We also measured the processing time for the change in the sampling rate of all involved sensors. This happens only if the sensors used in the mode functionality cannot be reused for the mode transition to a new mode. Note that the processing time between any pair of states is symmetric.

To put these overhead costs in some perspective, around 28 states (modes) per every second, on average, can be hosted on a smartphone of our modest configuration. For instance, when a mobile app needs around 10 sensor samples every second, which could be collected from different sensors, our LBA system could support about 2.8 applications simultaneously. In the case of 1 sensor sample

Table 1 The processing time for the ongoing process of monitoring the state transition triggers (in *milliseconds*)

Mode	Accelerometer	Gyroscope	GPS	Total
Stilling	Old: 0.39	New: 4.50	New: 7.31	12.2
Walking	Old: 0.39	Old: 0.39	New: 7.31	8.09
Bicycling	Old: 0.39	New: 4.50	Old: 0.39	5.28
Driving	Old: 0.39	New: 4.50	Old: 0.39	5.28

Table 2 The overhead costs of making a transition from one mode to another one (measured in *milliseconds*)

	Stilling	Walking	Bicycling	Driving
Stilling	X	12.32	19	19
Walking	12.32	X	31.32	31.32
Bicycling	19	31.32	X	1.61
Driving	19	31.32	1.61	X

Table 3 The overhead energy consumption of our mobile app (measured in *milli-joule*)

	Accelerometer	Gyroscope	GPS
Cost per transition	2.42	3.18	4.05
On-going cost per feed set	0.63	1.24	1.93

(i.e., 1 hz), our LBA system could support around 115 applications simultaneously on one smartphone.

4.2 Power Consumption Overhead

The overhead energy consumption of our mobile app is displayed in Table 3. It were calculated to be 0.63 mJ for the accelerometer sensor, 1.24 mJ for the gyroscope sensor, and 1.93 mJ for the GPS sensor. We found that the overhead costs of switching the sampling rate of the accelerometer sensor to be 2.42 mJ, the gyroscope sensor to be 3.18 mJ, and the GPS sensor to be 4.05 mJ.

4.3 Overhead Analysis

Without the actual sensing, we calculated the power overheads caused by our mobile app for trigging state transitions. We used this overhead analysis to measure the energy non-sensing overheads of the proposed multi-modal context-switching approach.

The average power used by the multi-modal approach were measured to 70.4 mJ and 79.6 mJ for the accelerometer and gyroscope sensors, respectively. This was approximately 4% of the total power demand of the accelerometer sensor experiments and 0.8% for the power demand of the gyroscope sensor experiments. The significant power difference is justified by the order-of-magnitude more massive power overheads of the gyroscope sensor.

5 Conclusions

In this chapter, we presented an LBA-based system that would help businesses advertise their projects and services affordably. We implemented two types of applications: a web-based application running on a server hosted in the cloud and a mobile-based app running on smartphones. Also, we presented the multi-modal sensing approach to represent and program mobile apps based on sensor context switching. The main objective of the proposed multi-modal sensing approach is to

separate the sensing and functionality concerns, which makes developing a context-switching app more modular.

Several sets of experiments were conducted to assess the scalability and performance of our LBA system at the mobile user side. The experiments' results illustrated that the developed LBA system is scalable and highly responsive in hosting numerous advertisements in the system.

For future work, we are looking at ways of how we can compose ModeSens [30], a multi-modal sensing approach, with ShareSens [27] in order to provide the ability to share sensor data between a large number of apps running on the same mobile device. The combination of ShareSens and ModeSens would be beneficial for backing the sensing requirements of a broad domain of research work [3, 6, 8, 23–25, 28, 32] and applications [1, 2, 5, 7, 9, 11–13, 26, 31]. Finally, extensive experiments will be conducted using big datasets to further study the robustness of our LBA system.

Data Availability

The data and the source code are available online for the public use at: https://github.com/ahmed-pvamu/Location-based-Mobile-Advertising-System

Acknowledgments National Science Foundation (NSF) partially supports this research work under award number 2011330.

References

1. A. Abdelmoamen, A modular approach to programming multi-modal sensing applications, in *Proceedings of the IEEE International Conference on Cognitive Computing San Francisco* (2018), pp. 91–98
2. A. Abdelmoamen, N. Jamali, A model for representing mobile distributed sensing-based services, in *Proceedings of the IEEE International Conference on Services Computing, San Francisco* (2018), pp. 282–286
3. A. Abdelmoamen, D. Wang, N. Jamali, Approaching actor-level resource control for Akka, in *Proceedings of the IEEE Workshop on Job Scheduling Strategies for Parallel Processing, Vancouver* (2018), pp. 1–15
4. AdWords: grow business with Google ads https://ads.google.com/home/. Accessed 26 Oct 2021
5. A.A. Ahmed, A model and middleware for composable IoT services, in *Proceedings of the International Conference on Internet Computing & IoT, Las Vegas* (2019), pp. 108–114
6. A.A. Ahmed, A privacy-preserving mobile location-based advertising system for small businesses. Eng. Rep. **e12416**, 1–15 (2021). https://doi.org/10.1002/eng2.12416
7. A.A. Ahmed, G. Agunsoye, A real-time network traffic classifier for online applications using machine learning. Algorithms **14**(8) (2021). https://doi.org/10.3390/a14080250
8. A.A. Ahmed, M. Echi, Hawk-eye: an AI-powered threat detector for intelligent surveillance cameras. IEEE Access **9**, 63283–63293 (2021). https://doi.org/10.1109/ACCESS.2021.3074319

9. A.A. Ahmed, T. Eze, An actor-based runtime environment for heterogeneous distributed computing, in *Proceedings of the International Conference on Parallel & Distributed Processing, Las Vegas* (2019), pp. 37–43
10. A.A. Ahmed, N. Jamali, CSSWare: a middleware for scalable mobile crowd-sourced services, in *Proceedings of the 7th EAI International Conference on Mobile Computing, Applications and Services (MobiCASE'15), Berlin* (2015), pp. 181–199
11. A.A. Ahmed, G.H. Reddy, A mobile-based system for detecting plant leaf diseases using deep learning. AgriEngineering **3**(3), 478–493 (2021). https://doi.org/10.3390/agriengineering3030032
12. A.A. Ahmed, A. Olumide, A. Akinwa, M. Chouikha, Constructing 3d maps for dynamic environments using autonomous UAVs, in *Proceedings of the 2019 EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous'19), Houston* (2019), pp. 504–513
13. A.A. Ahmed, S.A. Omari, R. Awal, A. Fares, M. Chouikha, A distributed system for supporting smart irrigation using IoT technology. Eng. Rep. **3**, 1–13 (2020). https://doi.org/10.1002/eng2.12352
14. G. Aydin, B. Karamehmet, A comparative study on attitudes towards SMS advertising and mobile application advertising. Int. J. Mobile Commun. **15**(5), 514–536 (2017)
15. L. Besaleva, A. Weaver, CrowdHelp: a crowdsourcing application for improving disaster management, in *Proceedings of the IEEE Conference on Global Humanitarian Technology* (2013), pp. 185–190
16. Bootstrap: an open-source framework for designing front-end web application https://getbootstrap.com/. Accessed 26 Oct 2021
17. A.D. Carli, M. Franco, A. Gassmann, C. Killer, B. Rodrigues, E. Scheid, D. Schoenbaechler, B. Stiller, WeTrace – a privacy-preserving mobile covid-19 tracing approach and application (2020). arXiv:2004.08812
18. S. Dhar, U. Varshney, Challenges and business models for mobile location-based services and advertising. Commun. ACM Mag. **54**(5), 121–128 (2011)
19. W. Gong, B. Zhang, C. Li, Location-based online task assignment and path planning for mobile crowdsensing. IEEE Trans. Veh. Technol. **68**(2), 1772–1783 (2019)
20. R.Y. Li, S. Liang, D.W. Lee, Y.J. Byon, TrafficPulse: a mobile GISystem for transportation, in *Proceedings of the ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems* (2012), pp. 9–16
21. S. Meng, W. Huang, X. Yin, M.R. Khosravi, Q. Li, S. Wan, L. Qi, Security-aware dynamic scheduling for real-time optimization in cloud-based industrial applications. IEEE Trans. Ind. Inf. **17**(6), 19–28 (2021)
22. M. Min, R. Sasank, S. Katie, Y. Nathan, B. Jeff, E. Deborah, H. Mark, H. Eric, W. Ruth, P. Boda, PEIR: the personal environmental impact report, as a platform for participatory sensing systems research, in *Proceedings of the ACM Conference on Mobile Systems, Applications, and Services MobiSys, Poland* (2009)
23. A.M.A. Moamen, H.S. Hamza, On securing atomic operations in multicast AODV. Ad-Hoc Sensor Wireless Netw. **28**, 137–159 (2015)
24. A.A. Moamen, N. Jamali, An actor-based approach to coordinating crowd-sourced services. Int. J. Serv. Comput. **2**(3), 43–55 (2014)
25. A.A. Moamen, N. Jamali, CSSWare: a middleware for scalable mobile crowd-sourced services, in *Proceedings of MobiCASE, Berlin* (2015), pp. 181–199
26. A.A. Moamen, N. Jamali, CSSWare: an actor-based middleware for mobile crowd-sourced services, in *Proceedings of the 2015 EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous'15), Coimbra* (2015), pp. 287–288
27. A.A. Moamen, N. Jamali, ShareSens: an approach to optimizing energy consumption of continuous mobile sensing workloads, in *Proceedings of the 2015 IEEE International Conference on Mobile Services (MS'15), New York* (2015), pp. 89–96

28. A.A. Moamen, N. Jamali, Supporting resource bounded multitenancy in Akka, in *Proceedings of the ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH Companion 2016)* (2016), pp. 33–34
29. A.A. Moamen, N. Jamali, Opportunistic sharing of continuous mobile sensing data for energy and power conservation. IEEE Trans. Serv. Comput. **13**, 503–514 (2020)
30. A.A. Moamen, J. Nadeem, ModeSens: an approach for multi-modal mobile sensing, in *Companion Proceedings of the 2015 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity* (ACM, Pittsburgh, 2015), pp. 40–41
31. A.A. Moamen, H.S. Hamza, I.A. Saroit, Secure multicast routing protocols in mobile ad-hoc networks. Int. J. Commun. Syst. **27**(11), 2808–2831 (2014)
32. A.A. Moamen, D. Wang, N. Jamali, Supporting resource control for actor systems in Akka, in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS 2017)* (2017), pp. 1–4
33. MobileAds: third-party ad serving platform for rich media and video advertising https://www.mobileads.com/. Accessed 26 Oct 2021
34. D. Paulino, A. Reis, J. Barroso, H. Paredes, Mobile devices to monitor physical activity and health data, in *Proceedings of the 12th Iberian Conference on Information Systems and Technologies (CISTI)* (2017), pp. 1–4
35. PokemonGo: an augmented reality mobile game https://www.pokemongo.com/en-us/. Accessed 26 Oct 2021
36. Retrofit: a type-safe http client for Android and Java https://square.github.io/retrofit/. Accessed 26 Oct 2021
37. Waze: a GPS navigation software app owned by Google https://www.waze.com/. Accessed 26 Oct 2021
38. K. Xu, W. Zhang, Z. Yan, A privacy-preserving mobile application recommender system based on trust evaluation. J. Comput. Sci. **26**, 87–107 (2018)
39. M. Zook, M. Graham, T. Shelton, S. Gorman, Volunteered geographic information and crowdsourcing disaster relief: a case study of the Haitian earthquake. J. World Med. Health Policy **2**(2), 7–33 (2010)