Near-Field Communication Sensors and Cloud-Based Smart Restaurant Management System

Hassain Saeed, Ali Shouman, Mais Elfar, Mostafa Shabka, Shikharesh Majumdar, Chung Horng-Lung
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada
{hussainsaeed, alishouman, maiselfar, mostafashabka}@cmail.carleton.ca, {majumdar, chlung}@sce.carleton.ca

***Abstract*— In this paper, we introduce an efficient and user-friendly Smart Restaurant Management System. This system will solve key problems faced by restaurants today through the use of technologies such as Mobile and Web applications, Internet of Things (IoT), Near-Field Communications (NFC) sensors, and cloud computing. Restaurants have many inefficiencies due to human limitations that can be resolved through automation and device-to-device communication. This Smart Restaurant Management System accomplishes this by providing two interfaces for the two types of users in restaurants; an Android mobile application for customers and a web application for restaurant staff members. The Android mobile application allows customers to have a seamless dining experience with features such as finding available parking spaces easier through internet-connected infrared proximity sensors in the parking lot, finding available tables at the restaurant easier through NFC sensors, ordering dishes through an interactive menu, and being able to pay the bill from their NFC equipped phones. The web application provides staff members benefits such as collecting data and statistics on the restaurant's performance in real time and automating the order placement system for waiters and cooks via IoT technology.**

***Index Terms*—Smart Restaurant, Restaurant Automation, Device-to-Device Communication, Near Field Communications-based Smart Cities, Cloud Computing, mobile application.**

I. INTRODUCTION

The increased availability of inexpensive sensors, mobile communication, and cloud technology has led to an increase in the popularity of Internet of Things (IoT), which concerns the interworking of a network of devices using sensors, electronics, software, and connectivity provided by a communication network. Examples of applications that use IoT technology include smart grid, smart homes, as well as sensor embedded bridges and sensor based industrial machinery [1]. Interest in IoT technology is also increasing rapidly: the number of IoT devices is predicted to increase to 50 billion by 2020 [2]. This paper contributes to the world of IoT through introducing the Smart Restaurant Management System (SRMS) that utilizes mobile smart devices, Near Field Communication (NFC) sensors, proximity sensors, a microcontroller, and cloud computing. This Smart Restaurant Management System was developed as a final year Capstone Engineering Project at Carleton University, and it is currently being enhanced in the university laboratory with the aim of building a marketable product.

In almost every single nation and culture in the world today, restaurants hold an important role as a hub for social interactions. Not only do they provide food, drinks, and sustenance to individuals, but they also function as a place where groups of people come together to socialize, connect, and share great experiences together. Restaurants are also extremely lucrative businesses, as the global restaurant industry grew by 6.2% in 2014 to reach a value of $2,737.1 billion, and it is forecasted to grow to a value of $3,805.8 billion by 2019 [3]. However, despite restaurants holding such an important role in our society and despite them earning high revenues, the restaurant industry has had few changes to how it has been operating in the past century. In spite of all of the great technical advancements that have been made in recent decades with the rise of the internet, mobile smart devices, and cloud computing, the restaurant industry still has not fully exploited the full capabilities of these technologies and still has a great deal of inefficiencies that are unaddressed.

Examples of inefficiencies in a regular restaurant management systems include: customers waiting for a table to open up and for a waiter to seat them, receiving a non-interactive paper menu displaying limited information about the food available (typically containing only the dishes' names, prices, and brief descriptions), waiting for a waiter to arrive to them before customers can order, receiving no information about the progress of their meal while waiting for it to arrive, waiting once more for the waiter to provide their bill to pay, and finally no automated way for customers to directly provide feedback to the restaurant's management immediately after finishing their meal. This project aimed at improving the efficiency of the regular management systems used in most restaurants today by utilizing current Information and Communications Technology (ICT), such as mobile applications, web applications, IoT, NFC, and cloud computing.

The SRMS has a number of novel features which include:

- Versatility of an End-to-End Solution: The SRMS supports a variety of smartphones, and it provides assistance at various stages of the entire dining experience from finding parking spots and available tables to ordering food and paying for the bill
- Flexibility: Clients can selectively use the SRMS for specific features and then perform other features (e.g. bill payment) through conventional means.

978-1-5090-4130-5/16/$31.00 ©2016 IEEE

- Cost Reduction for Restaurant Owners: Client's smartphones are leveraged to access the SRMS, and not restaurant provided iPads or tablets
- Ease of Integration: Due to the SRMS's architecture being managed by a Cloud Web Server and Cloud Database in the back-end, SRMS can be easily integrated into any establishment with little effort or strain on their behalf and on our back-end systems.
- Support for Data Analytics: the SRMS can track user data such as the demographics and ordering habits of customers, and key performance indicating statistics such as mean food preparation and serving times

The rest of the paper is structured as follows. Section II discusses a representative of the previous works related to this Smart Restaurant Management System, and how this system improves on the existing state of the art. Section III describes the technologies, the system architecture, and the design approaches used to develop SRMS. Section IV focuses on the functionality and the interfaces of the system. Finally, Section V presents the conclusions and plans for future work.

II. Related Work

Efforts to automate traditional restaurant management systems have been conducted in the past few years, and a representative set of such works is presented in this section. The usual process that takes place when ordering food is almost entirely manual and involves several staff members – including waiters and cooks – to coordinate their activities with each other in order to process customer orders. The waiter has to provide customers with paper menus, check whether or not they are ready to order, and write down their orders. The waiter has to then communicate the details of the order to the kitchen staff in order for the dishes to be prepared, and check on the status of the order whenever they go back to the kitchen. Once the customers finish their meals they would have to wait for the waiter to pass by and give them the bill. The waiter would then have to process the payment of every customer on the table. This process is lengthy and it can be prone to human errors and delays due to waiters getting the wrong order or delivering the wrong bill.

The main driver for efforts to automate traditional procedures has been the swift rise of wireless technologies used to ease conventional food ordering processes. Systems that use wireless food ordering include Wireless Ordering System (WOS) [4], Microworks [5] and Nextepsystems [6]. All of these systems however, require either Personal Digital Assistants (PDAs) or iPads, and they are mainly used to put in orders. This eliminates errors relating to receiving orders, but they do not eliminate the time delay for the waiter to show up and collect the iPads/PDAs or the time delay for waiters to collect bill payments. These systems are prone to limitations such as:

- Restaurants have to purchase and maintain enough iPads/PDAs to be able to serve customers at peak hours. This places a significant financial and maintenance responsibility on the restaurant owners, especially small restaurants. In addition some devices might only be used during peak hours and put on the shelf during normal hours.
- These systems offer individual services to automate a specific part of the ordering procedure and not the process in its entirety. Hence, each of these systems eliminate certain inconveniences but not all of them.
- Real time interaction with the kitchen and waiters, including facilities for proving real time feedback, do not seem to be supported in most systems.

To overcome these limitations, we have devised an IoT based Smart Restaurant Management System that uses mobile smart devices, NFC sensors, and cloud technology to offer a more complete and efficient automated dining experience.

III. Technology, Architecture, and Design

This section presents the technologies used and the architectural and design decisions that were taken to develop the Smart Restaurant Management System.

A. Technologies Used

The SRMS described in this paper uses Android smartphones and tablets to pair with the NFC sensors and cloud technology. NFC is becoming widely more popular due to its ease of use, affordability, and extremely power efficient data communication. NFC provides a short span of wireless communication between two NFC enabled devices or an NFC enabled device and an NFC chip (also known as an NFC tag or sensor), without the need for an internet connection. [7]

Most Android smartphones and tablets are NFC enabled, and can be programmed to read and react to NFC tags. Hence, applications written for exploiting Android's NFC capabilities can lead to a product that is expected to be attractive to a large number of customers.

The Android application developed for SRMS was implemented using Java and Extensible Markup Language (XML). Android Studio was used as the IDE for the Android application, with its Android SDK including the Android debugger and libraries. The web-application developed for SRMS was programmed with JavaScript, Hyper-Text Markup Language (HTML) and PHP.

Cloud technology involves both computing and storage. The cloud computing and storage services used in this project were provided by Amazon Web Services (AWS). Cloud storage is used for storing and accessing information from a remote data center via the Internet. The type of database used in SRMS for storing data on the Cloud was MySQL. Cloud computing is used for executing programs on virtual machines provided by a remote data center. It is often used by businesses with changing requirements and agile environments, as well as scientific and engineering research centers. Scaling up and down the cloud capacity is easier than upgrading physical servers or other storage and computing units [8]. The web applications of SRMS were deployed on an Apache HTTP Server installed on a virtual machine in the Cloud.

The parking feature of SRMS requires the use of an Arduino Uno R3 Microcontroller and proximity sensors (Infrared distance sensors) located at every parking spot. The proximity sensors are installed above every parking spot so that they can detect if there is a car parked 5 metres under it. If it detects a parked car it sends a unique voltage signal to the Arduino microcontroller. The microcontroller reads the signal, and then uses its internet connection to update the MySQL database in the cloud to indicate that the parking spot is occupied. Both the Arduino and the proximity sensors were chosen because they provided the desired functionality at a low cost. However, while the Arduino microcontroller was used for implementing a functioning prototype, for future work a more scalable and powerful controller would be used.

B. System Architecture

The system architecture is presented in Fig. 1 that shows the interconnection among the various system components. The two main software components of SRMS are the Android application and the web application. The Android application is the means by which the customers interact with the system, while the web application is the means by which the restaurant staff interact with the system. The Android and web applications in the system do not interact directly, instead they both have access to the Apache HTTP web server hosted on the EC2 service of the AWS suite. Both applications access, query, and update the database via this web server.

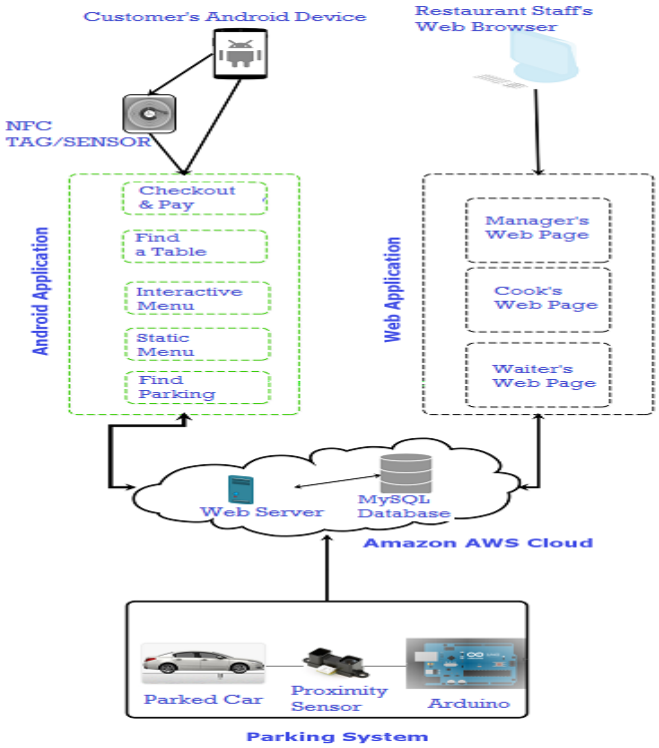

Fig. 1.System architecture of the Smart Restaurant Management System

Fig. 1 presents a high level overview of the various components including the AWS cloud, Android application, and web application, NFC sensors, the proximity sensor, and Arduino Uno. These components can be grouped together to create 5 subsystems in this project. All of the subsystems work together to create the fully functioning system. The NFC subsystem, AWS cloud subsystem, and the parking subsystem, are the hardware subsystems while the web application and the Android application are the software subsystems. A short description of the functionality provided by each hardware subsystem is presented.

- The NFC subsystem includes all of the NFC sensors that are used to trigger the launch of specific features in the Android application.
- The parking subsystem is used to collect and disseminate information about the availability of parking spots. The parking subsystem consists of infrared proximity sensors connected to an Arduino Uno R3 Microcontroller, which in turn is connected to the Internet so it can update the database on the cloud.
- The AWS cloud subsystem consists of an Apache HTTP Server as a web server and a MySQL relational database running on a virtual machine in Amazon's cloud service.

C. System Design

The design of the Smart Restaurant System is split into three main phases. These phases follow a chronological order of actions that are performed by the customers when visiting a restaurant. Phase 1 deals with all the scenarios that occur before the customer enters the restaurant, such as finding parking spots and booking a table inside the restaurant. Phase 2 deals with what happens from the customer's end once they are seated at the restaurant and begin to order their meals from the Android application on their phone via NFC sensors. Finally Phase 3 deals with all of the events that occur after the customer finishes eating their meal. This includes providing the customer with flexible payment options such as paying for their own bill, paying for a friend's bill, or splitting the cost of an item with a friend. Due to space limitations, this section will only go through the design of some of the main features of each phase of the application. The complete discussion of the system design is described in our Capstone Project report [9].

Phase 1: The 'Find Available Parking' feature provides the customers with an up-to-date parking map on their Android device that displays the availability information of each parking spot. This feature requires the customer to tap their Android device on the NFC tag placed at the entrance of the parking lot. The NFC tag is programmed to redirect the Android application to the parking lot page, which will send a query request for the table "PARKING_SPOTS" stored on the MySQL database located on the Amazon cloud. The "PARKING_SPOTS" table maintains information about the availability of the parking spots in real time. The Android application initiates this database query request by executing a PHP script, which connects to the database through the web server. This PHP script then generates a query response that states which parking spots are available and which ones are not. This query response is sent back to the Android application, which reads it and eventually displays the information on the screen of the Android device in a user-

friendly way. As mentioned previously, the database is kept up-to-date in real time through proximity sensors placed above each parking spot that are connected to an Arduino Uno microcontroller. The Arduino Uno has Internet connectivity and updates the "PARKING_SPOTS" table on the database if the proximity sensors detect a parked car. A sequence diagram for the Find Available Parking feature is displayed in Fig. 2.

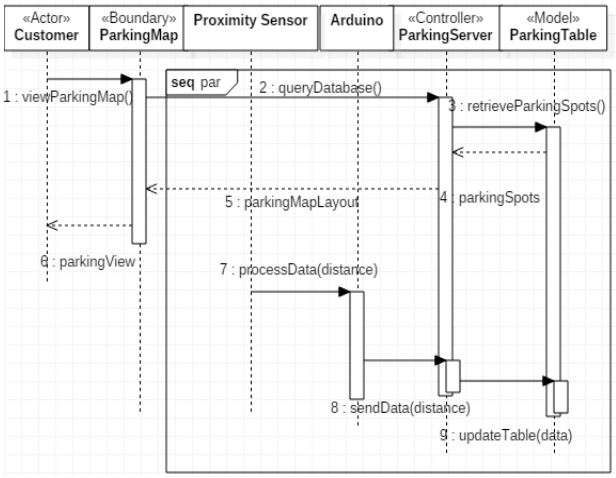

Fig. 2. Sequence diagram for the Find Available Parking feature

Phase 2: The "Interactive Menu" feature allows customers the flexibility of ordering food directly from their personal smartphone or tablet. To access this feature, customers need to be logged in to their Android application and then tap the NFC tag on their table. Thereafter, the user can either view the nutritional information of the dishes or directly choose the dishes they want to eat and checkout. They are then taken to an 'Order Summary Page' containing all of their ordered dishes as well as their total bill. From there they can either go back to the menu page if they wish to modify their order or, if they are satisfied they can submit their order by pressing a 'Send Order' button. The order is then inserted in the database on the cloud which automatically updates the 'Cook's Web Page' by adding the newly placed order. In addition, customers can tap a different NFC tag in order to check the status of their order. This is done by querying the database and displaying the results back to customer whenever the "Check Dish Status" NFC tag is tapped. The sequence of events that describes the 'Interactive Menu' feature is shown by the sequence diagram in Fig. 3.

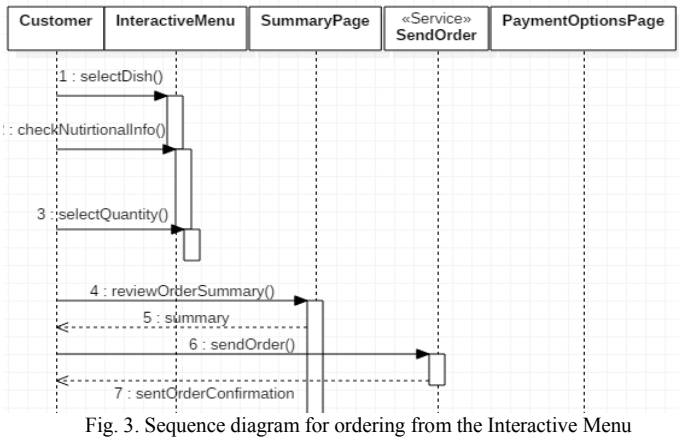

Fig. 3. Sequence diagram for ordering from the Interactive Menu

Phase 3: The "Split Items with Tablemates" feature provides the customer with the capability to split the cost for one or more items on their bill equally with their tablemates. When the "Split Item with Tablemates" page is opened, the Android application will query the "ORDERS" database table stored on the cloud database. This query then returns the list of all items ordered by the customer and also a list of all of the other customers sitting on the same restaurant table (i.e. the tablemates). The customer can then select the specific items they want to split the cost of and the tablemates they want to split them with. After selecting the specific items and tablemates, the customer needs to press the "Send Split Request Button". This will then send a notification to the Android devices of the requested tablemates. After this notification has been sent, the cloud database updates to reflect that items have been split by multiple customers. Both the customer and their tablemates get redirected back to the "View Final Bill" page where their bills are updated to reflect the split food item costs. The sequence diagram for this feature is shown in Fig. 4.

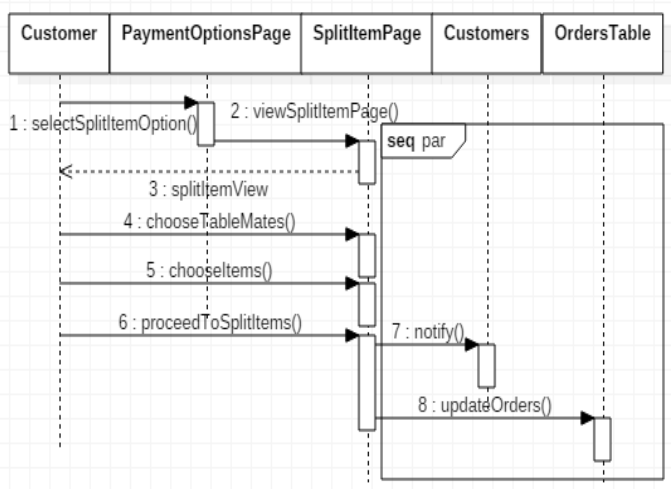

Fig. 4. Sequence Diagram of the Split Items with Tablemates feature

IV. Functionality and Interfaces

This section presents the various functionalities of the interfaces of the Smart Restaurant Management System utilized by its two main users, the customers and the staff members of restaurants. For each user a different interface is provided; restaurant customers use an Android mobile application and restaurant staff members use a web-based application accessible through any Internet browser. Due to space limitations, this section will only cover the functionality of these two interfaces and not the functionality of the Parking subsystem, the AWS cloud subsystem, or the NFC subsystem. A complete discussion of the functionalities of every subsystem is presented in our Capstone Project report [9]

D. Interface for Customer Interaction

Customers interact with SRMS through the interface of an Android mobile application. Upon logging in to the Android application, customers have three available features, each of which can be invoked upon tapping a specific NFC tag located at the restaurant: 'Find Available Parking', 'Find an Available Table', and the 'Interactive Menu'. Fig. 5 shows a screenshot

of the 'Find Available Parking' feature, which displays 'Available' for parking spots in which the infrared proximity sensors do not detect cars over and displays 'Unavailable' for the parking spots for which the infrared proximity sensors do detect a car over. Fig. 6 demonstrates the 'Find an Available Table' feature, which shows the different types of tables present at the restaurant (i.e. 2 seat table, 4 seat table, 5 seat table, etc.). Upon clicking any of these types of tables the Android application will display if there is a table of that type available based on the data stored in the database. A table is designated 'Unavailable' in the database if the Interactive Menu NFC tag on the table has been accessed and read, and then designated 'Available' once the customers at that table have paid their bills. Fig. 7 demonstrates the 'Interactive Menu' feature, in which users can swipe through the tabs to select food items from the various Breakfast, Lunch, Dinner, Dessert, or Drinks menus. Pressing the name of the food item will also display a picture and the nutritional information of the dish.

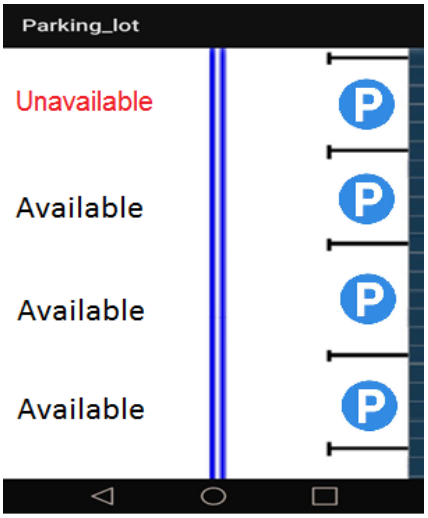

Fig. 5: 'Find Available Parking' feature on the Android device

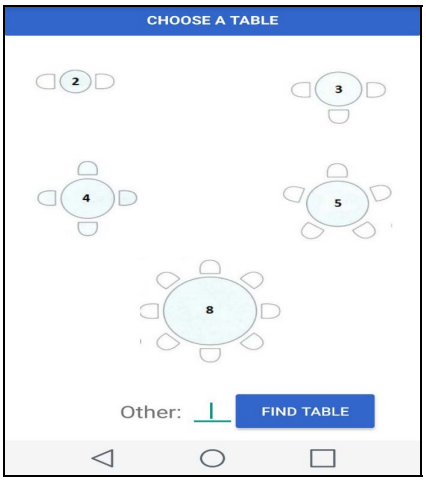

Fig. 6. 'Find an Available Table' feature on the Android device

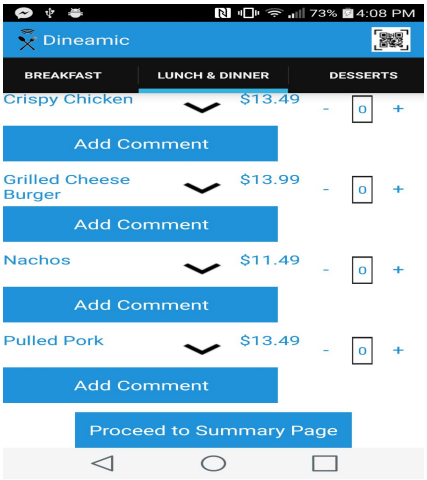

Fig. 7: 'Interactive Menu' feature on the Android device

After selecting all of the menu items they desire, customers can press the 'Proceed To Summary Page' button, at which point they will be shown a summary of their order. If satisfied with the summary, customers can press the 'Send Order' button to officially send their complete order to the kitchen staff. As users wait for the kitchen staff to prepare the dish, they can tap another NFC tag to view the status of their order, as can be seen in Fig. 8. Finally, after customers have finished their meal and wish to leave, they can directly pay their bill though the Android application by entering their credit card information (either manually or through a camera picture) or using a PayPal account, as can be seen by Fig. 9. Before paying their bill, customers have the option to equally split the cost of specific food items with other tablemates, or to pay for the entire bill of one of their tablemates. After paying the bill, customers are prompted to leave feedback of their restaurant experience, which restaurant managers can then read and learn from.

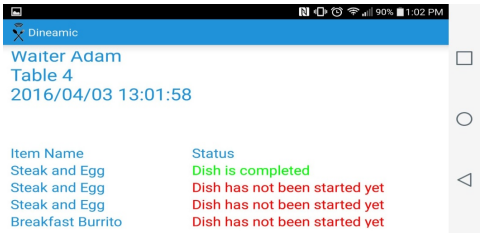

Fig. 8. 'View the Status of your Order' feature on the Android device

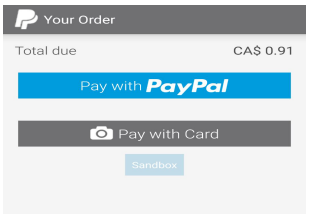

Fig. 9. 'Pay Your Bill' feature on the Android device

E. Interface for Restaurant Staff Interaction

Restaurant staff members interact with SRMS through the interface of a web application, which is accessible through any internet browser on a desktop, tablet, or a mobile device. There are three specific pages in the web-application that staff

members can access based on their job position at the restaurant: the Waiter page, the Cook page, and the Manager page. Staff members can only log into their respective pages using their employee username and password. The Waiter page, as can been seen in Fig. 10, notifies waiters immediately as soon as customers request to see them through their Android application or when orders are finished in the kitchen and ready to serve. The Cook page, which is displayed in Fig. 11, shows every food item ordered by customers through their Android applications. As cooks prepare the food items, they can update the status of each dish to indicate if the dish is 'Not Started', 'Being Prepared' or 'Completed'. The Manager page, which can be seen in Fig. 12, allows the manager the option to 'View Analytics'. The View Analytics feature displays real-time analytics and data about the performance of the restaurant for the current day, including information such as how much revenue has been earned, number of orders placed in the day, most popular dish, as well as the busiest and slowest hour(s).

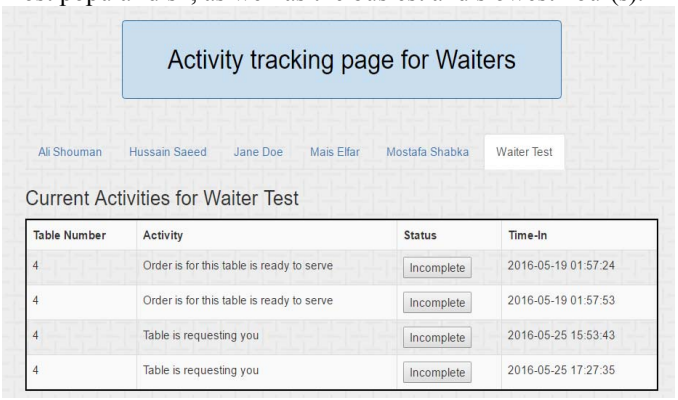

Fig. 10. The Waiter page on the web application

Fig. 11. The Cook page on the web application

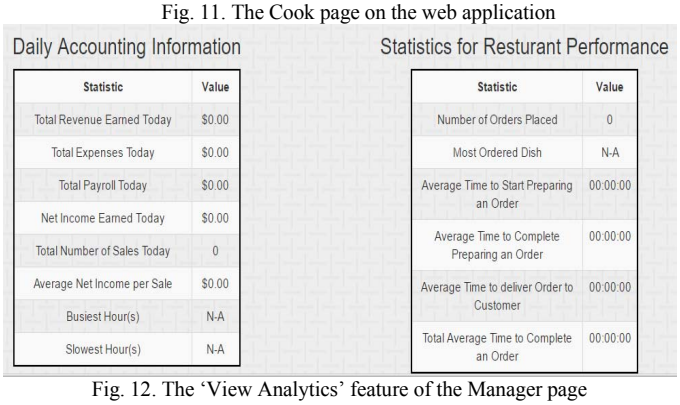

Fig. 12. The 'View Analytics' feature of the Manager page

V. Summary and Conclusions

In this paper, we have presented an efficient and user-friendly solution that will solve many of the problems faced by restaurants, by effectively using technologies of mobile and web applications, Internet of Things, Near-Field Communication sensors, and cloud computing. Different from previous approaches, our solution of a Smart Restaurant Management System does not require restaurants to purchase multiple iPads/PDAs to give customers for ordering. Instead it allows customers to bring their own device and wirelessly order food from an Android application on their mobile smartphone or tablet. As well, through the use of proximity sensors and NFC sensors, customers can also use the Android application on their personal smart devices to find available parking spaces, find an available table, and pay for their own bill. SRMS also provides a web application that allows restaurant staff members to manage their respective work and view real time analytics about the restaurant. Rigorous functional testing and live demonstrations of SRMS to multiple audiences have indicated its effectiveness

The current status of the project is that a fully functioning prototype of a SRMS has been built, and the project has been selected to receive additional funding by Carleton University in order to enhance it into a more marketable product. Future work includes the testing of SRMS in real restaurants with actual customers in order to receive feedback about its usability, reliability, and practicality. Making SRMS compatible with iOS devices and QR code based tags also forms an important direction for future work.

References

[1] A. McGregor *et al.*, "A Cloud-Based Platform for Supporting Research Collaboration," *2015 IEEE 8th International Conference on Cloud Computing*, New York City, NY, 2015, pp. 1107-1110, 2015.

[2] L. Trappeniers, M. A. Feki, F. Kawsar and M. Boussard. "The internet of things: the next technological revolution." *Computer*, vol. 46, no. 2, pp. 24-25, 2013

[3] "Global Restaurants Industry Profile." *Restaurants Industry Profile: Global* (2015): 1-33. *Business Source Complete*. Web. 14 June 2016.

[4] K. Kamarudin, et al., "The Application of Wireless Food Ordering System," MASAUM Journal of Computing, vol. 1, pp. 178-184, 2009.

[5] Wireless Tablet Ordering application, online at http://www.microworks.com/products/Handheld-Ordering.htm

[6] Dynamic Menu Displays application, online at http://www.nextepsystems.com/dynamic-menu-displays.

[7] NFC Forum, online at http://nfc-forum.org

[8] E. Gorelik, "Cloud Computing Models," Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA., January 2013.

[9] H. Saeed, A. Shouman, M. Elfar, M. Shabka. "Cloud Assisted NFC-Based Smart Restaurant Solution," Project Report, Systems & Computer Engineering, Carleton University, Ottawa, Canada, April 2016.