Gizmo Point of Sale

Using BLE and Raspberry Pi

Muhammad Awais Akhtar
Faculty of Computer Science and Electrical Engineering,
Kiel University of Applied Sciences
Germany
muhammad.a.aktar@student.fh-kiel.de

Mubashar Khokhar
Faculty of Computer Science and Electrical Engineering,
Kiel University of Applied Sciences
Germany
mubashar.khokhar@student.fh-kiel.de

Dr. Robert Manzke
Faculty of Computer Science and Electrical Engineering,
Kiel University of Applied Sciences
Germany
robert.manzke@fh-kiel.de

***Abstract*—**There exists several open source projects in the Point-of-Sale (PoS) domain, most of which are resource intensive and exceed requirements. In the present paper, we have focused on developing a system that has minimal deployment cost and uses low-energy technologies, thereby prolonging operational time. Our solution, Gizmo PoS, is a web based system that has many added functionalities which can fulfill needs of large as well as small businesses. This system is successfully deployed and tested as restaurant PoS using Raspberry Pi and trivial Android devices for collecting orders. Moreover, incorporating Bluetooth Low Energy (BLE) devices opens up new applied research opportunities.

Keywords*—Point-of-Sale, intelligent systems, business solution, Bluetooth Low Energy, Raspberry Pi*

I. INTRODUCTION

There are many freely available open PoS systems e.g. Floreant PoS, Unicenta and Lemon PoS etc. All these existing PoS are resource hungry when it comes to deployment. Gizmo PoS is a management system designed to provide a way to manage cash register, menu manipulation, order management, data processing, reporting and analysis which aids in helping the management take right decisions for business. It eliminates the cumbersome manual working of billing, maintaining menus and day to day promotional offers etc.

The aim of this project is to provide low cost deployment ready PoS, Which not only contains trivial functionalities, but also takes the advantage of low energy electronics like BLE chips to enhance the business and revenue by collecting different information about context e.g. weight context (of customer), temperature and time context.

In this paper, we have discussed initial implementation and results of Gizmo PoS, GizmoTix, along with of the proof of concept application using BLE (BlueGiga BLE 112 module). Moreover, a brief overview on potential research areas highlighted by this project can be further explored in applied context.

II. REALIZATION

Gizmo PoS provides trivial functionalities offered by any PoS. Users that interact with the system are waiters, customers, chefs and cashiers. Fig.1 shows users and their characteristics. A waiter with his/her Android devices takes order from customer. This order is referred to with table number that is detected with BLE chip attached to the table. Such BLE chips consist of unique identification number that is mapped to the table. The order is then transferred to local server via wireless connection. This local web server is deployed on a Raspberry Pi embedded system which uses Raspbian, Apache and MySQL. Once a waiter selects from an available menu and promotional offer, received from the local server and places the order and it will be printed at the kitchen to be prepared by the chefs.

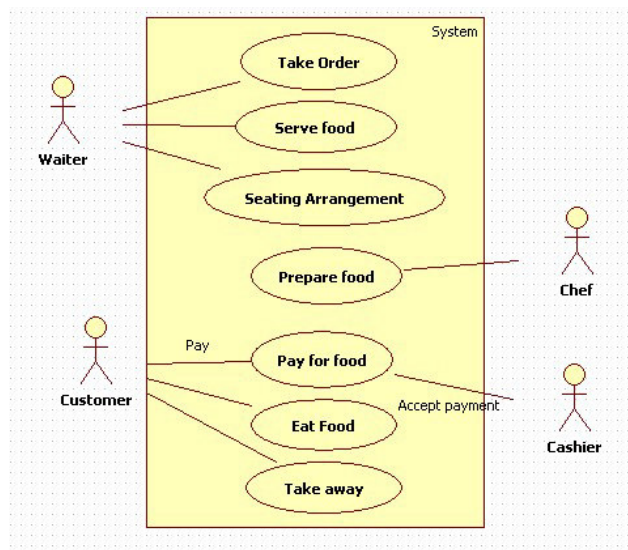

Fig. 1: User characteristics within the system to define the flow of the system.

An order can be modified if anything is requested at a later point. A payment receipt is printed for the consumed food only when requested through the system. The payment receipt includes cumulative cost of the food in case of multiple orders from the same customer. Moreover the order data is synced with the main server to produces reports for business insights. Fig. 2 shows complete Gizmo flow.

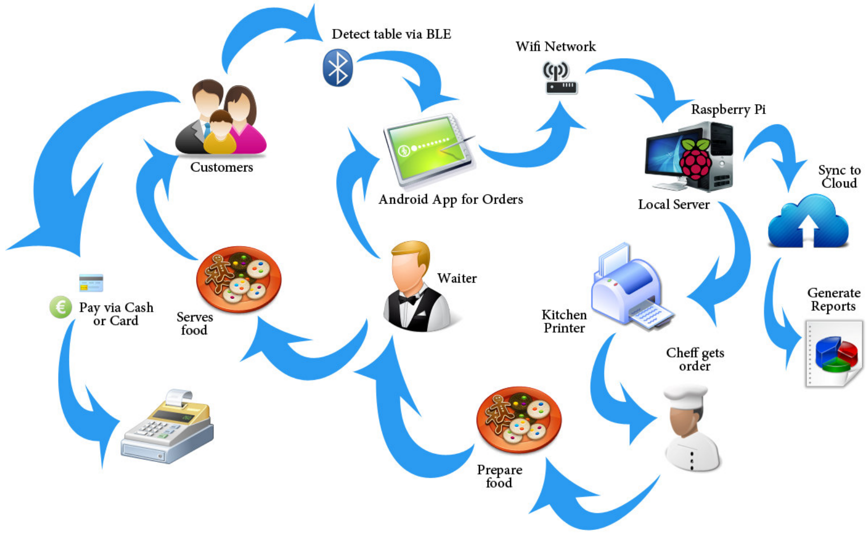

Fig. 2: Gizmo POS complete process flow

A. *Local Server*

The local server is a Raspberry Pi web server, located on the site. This server is attached with 2 tag printers. One printer prints the order into the kitchen and other is on the cashier's desk to print the receipt. A web interface is provided to the cashier/manager to view the status of the running orders and to proceed with the payments. The local server is fully manageable from the web interface. The local server has a built-in alpha version of printing services that can print the required data on any selected printer attached. Fig. 3 shows the Dashboard, hosted on Raspberry Pi, displaying a list of orders with order number, table number, amount, item count, order status (printed, served, cash out) and order time. Left control bar is for general PoS management functions like user, menu and items management. Images of items can also be added to the menus and items. Image files are converted to thumbnails to enhance the performance of server.

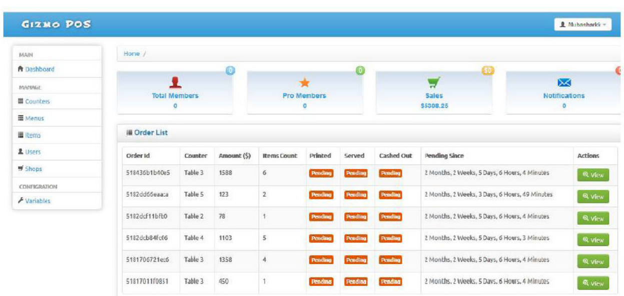

Fig. 3: View of the orders placed

B. *Android Application*

Gizmo PoS system includes an Android based application that is used by the waiters to take orders from the customers. This application can be easily built on different platforms as it is built by using PhoneGap [6], a hybrid mobile application development platform. This app sends a HTTP request to the local server to get the data required by the app and as a result get a list of items and menus with the required hierarchy of menus. On successful response from the server the app will display the list of menus for selection and an order can be prepared. On the completion of the order, the data is then sent back to the local server. Fig. 4 shows current order screen on Android device.

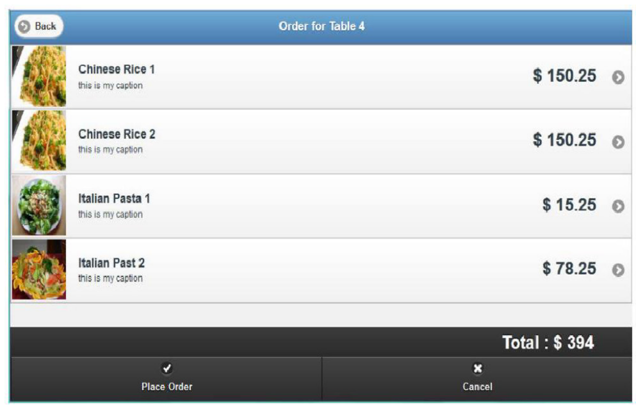

Fig. 4: Android app menu placement screen

C. *GizmoTix*

To enhance the business intelligence part of the system we have added web server that collects data on daily basis from all the local servers at specific time of each day. This is a centralized server for all our clients. This system not only gives business analytics in different visualization but also offers intelligent suggestions to clients, offering them different deals on basis of context. This web system will have a recommender system that will help in aiding business by making intelligent recommendation about inventory usage and customer preferences for a particular time period. By including different business clients into our system from different parts of the country, we can create a smart business network that can benefit the end user by giving him/her a best recommendation about a nearby restaurant. Fig 5 shows web analytics view which uses D3js and bootstrap.

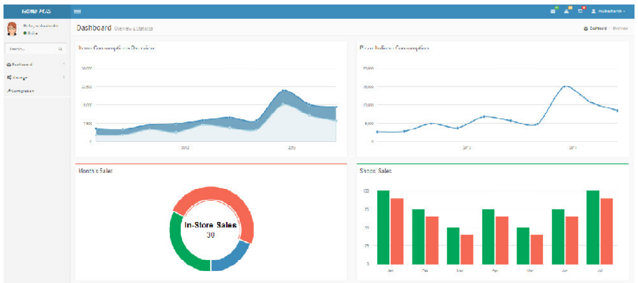

Fig. 5: GizmoTix Web Analytics View

D. *PoC BLE Application*

Bluetooth smart ready devices are the devices that can connect with Bluetooth smart (Bluetooth 4.0). Currently we only have developed a BLE device to uniquely identify the table identification number. We are working on adding more functionality to the device. For example Bluetooth smart ready devices can be used to communicate with the Android devices i.e., to notify the waiter that either customer is calling for bill/order and also to map the tables in the restaurant. Android 4.3+ supports BLE [2].

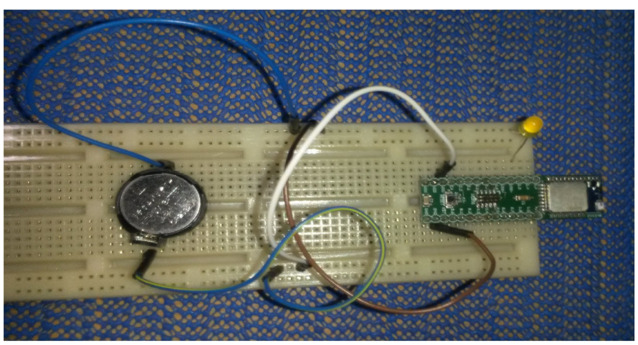

Fig.6a: BLE hardware Setup

BLE uses very low energy [1] and can run for around a year when powered by coin cell. We deployed BG script which uses GATT profile with different services including weight, battery and device information services. Some of these services are highlighted below:

1) Information Service, This service carries different characteristics e.g. Device Name, Hardware Serial number and Model.
2) Weight Service, Weight Service will contain weight characteristic which can retrieve the information about the weight of customer using load cell but for now it sends back incremented number from BlueGiga 112 module.
3) Battery Service, It gives us the information about the power left in the coin cell calculated as percentage on basis of voltage on the port VDD. This is provided as sample [2].

Fig. 6a shows BLE setup consists of a coin cell, a BlueGiga BLE112 module and a LED light which blinks when a device is connected or disconnected. Fig. 6b is proof of concept application that scans for the BLE devices and after connecting to BLE node, retrieves above mentioned services and their characteristics.

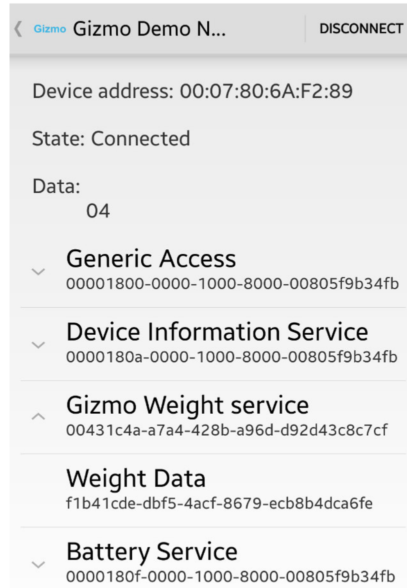

Fig 6b: Android App connected with Gizmo BLE Node.

III. Dependencies

This project has following hardware dependencies.

- Raspberry Pi
- Wi-Fi Router
- Android Tablet PC / Apple iPad
- Tag Printers
- BLE Chips

Beside those below are software dependencies

- Raspbian (Jan 14)
- Apache Server 2.4.1+
- PHP 5+
- MySQL 5+
- Android 4.3+

IV. Future Prospects

In future, our first main focus is to add more functionality to the GizmoTix server. After incorporating BLE and Android devices this project opens up great opportunities to further extend its applications and capabilities in future. Following are the potential research areas that can be explored further:

A. *Bluetooth Low Energy*

Exploring BLE more in details and looking for more use-cases of BLE e.g. overcoming the range hurdles by making ad-hoc connections or master-slave hierarchy where BLE devices are integrated in to the table can collect data from BLE integrated slave-chairs. Fig. 7 depicts master-slave hierarchy where table collects data from nearby chairs and then send this data along with the order data to the local server. This data is very useful in terms of users' context. This BLE network setup can also be used to call waiter without any delay or waving/raising hand.

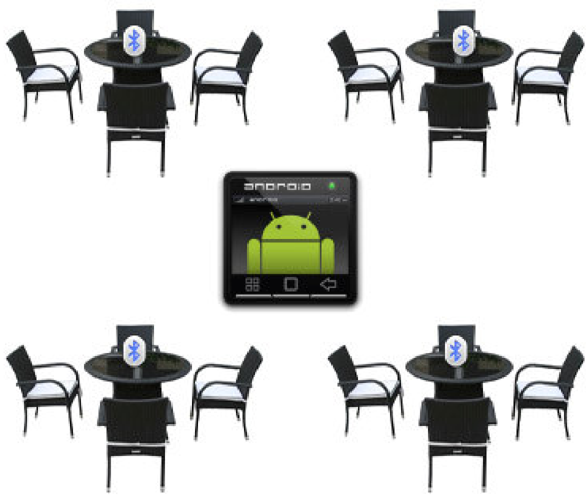

Fig 7: Master Slave Organization of the BLE devices within the system.

B. Business Intelligence

To make Gizmo PoS more intelligent, we still need to look into neural network or machine learning algorithms in accordance to collect data for helping or automating the business intelligence. For example, predicting sales of items on basis of temperature and weather conditions or offering food deals on basis of weight of customer or on weather.

C. Load Cells

Load cell is not a new technology; it is used for measuring the load or strain. We can also use these load cells to measure the weight. These load cells vary in shape and size [5]. The idea behind load cell is to convert force into electrical signal. The strain changes the effective electrical resistance of wire that can be then amplified [3] and hence measured [4]. A load cell can be attached to a chair and a node that is attached to the table can collect the weight information from these devices. From here user can collect it via Bluetooth devices or with nearby installed BLE dongle. For this purpose master-slave approach or broadcasting protocol needs to be researched.

Fig 8: A load cell mounted chip, SMD Sensors [4]

D. Security and Privacy

A customer's weight information is his intellectual property and he must be notified. On one hand, where broadcasting weight information broadens the interactive apps domain, also raises privacy concerns that must be pondered on.

V. Conclusion

We have successfully developed a deployment ready Gizmo PoS system using Raspberry Pi as local server along with an Android application for ordering the food. The live web server which collects data is still under development, where information will be visualized in different forms for analysis. Moreover, the logic to make it more intelligent by including machine learning and pattern recognition training on context information to guess what would be the food items liked by most customers, is still need to be implemented.

VI. Acknowledgement

The authors would like to thank everybody who has contributed to this project in one way or another specially to our colleagues Lukas and Simon for helping us in setting up BLE and Haseeb Ghaffar for helping us with the documentation of this project.

References

[1] Bluetooth Low Energy technology, BlueGiga Technologies website. (Retrieved: July 10, 2014)

[2] Bluetooth Low Energy, Android developers guide website. (Retrieved: 2nd October 2014)

[3] "Working with a load cell and an Arduino", Electronic Design Group, University of Chicago, USA. (April 2012)

[4] ACAM, "How to build Digital Load Cells with PicoStrain conveniently". (March 2010)

[5] "Load Cells and Force Sensors", Strain Measurement Devices website. (Retrieved: 2nd October 2014)

[6] "PhoneGap" by Adobe. (Retrieved: 2nd October 2014)