International Journal of Engineering Inventions
ISSN: 2278-7461, www.ijeijournal.com
Volume 1, Issue 8 (October2012) PP: 20-25

PORTABLE AND ECONOMIC SMART GENERIC DATA SERVER FOR SMALL SCALE BUSINESS

Manoj N Revanker, Manoj Kumar M V, Puneetha B H, Prashanth B S
[1,2,3,4]*Dept. of CE&E and IS&E., BIET, Davanagere, INDIA*

Abstract:—Now a day internet plays a very major role in business processes. The internet acts as the kind of the bonding between the customers and an organization (probably business), the customer can obtain the information of his desire by making the single click on organization's website. But, the major infrastructure to carry out this process demands the design, maintenance, hosting of website from the organization side, and it also demands the access to the internet from the customer side (in order to access the internet customer must have to have either personnel computer or any smart phone), but unfortunately we cannot find all the customers with above all mentioned requirements. Therefore, this paper focuses on the issue of building the portable and economic smart data server for small scale business applications. This proposed concept in this paper eliminates the need of internet, hosting, website maintenance, and costly hardware from organization side. And it also eliminates the need of personnel computer, or internet enabled smart mobile phone from customer side. Thus the concept proposed in this paper provides the economic, efficient, portable smart data server for small scale business. To implement this proposed concept the cost involved will be too less, the only requirement is one smart mobile phone (here we used android mobile phone). The prototype has been demonstrated using the android smart mobile phone; this can further be extended to any available smart mobile phones in market.

Keywords:—Broadcast receiver, manifest, SQLite, Context.

I. INTRODUCTION

Accessing to spatial data in mobile environment [1], uses an Internet-based connection to download the data located on a distant data server and a web browser interface to process and to visualize them locally. All data are transferred as the result of queries defined on the client side and executed on the data server. The data management on the Client side uses a data cache and takes into account the spatial aspect of data.

Literature survey conducted by us concludes the fact that no one has initiated the process to implement the system like mobile data server for small business applications, without using the basic infrastructure (like internet, hosting, website etc...). Though there are some of the works had been carried out to use the mobile devices to process the data. This paper aims at proposing the model of data server for small scale business organizations and users with easy to use and minimum implementation cost.

This proposed method uses the concepts like broadcast receiver for polling the incoming data requests. SQLite mobile database for storing and querying for data. The only involved in implementing this proposed concept is cost required to purchase smart mobile phone and cost of sending messages, even cost of sending messages now a days is so minimum and that can be negligible. This paper totally focuses on the small scale business and the overhead involved in the communication between small scale business man and his customers. Further this proposed concept can be extended to large scale business also with the emerging technology support

Further this data server can be only implemented once and can be used with different types of purposes hence the name given is generic. The data server will works fine with grocery shop to hardware and super market to milk shop. This concept which we will going to propose further in this paper is flexible enough to cope with any type of business application to serve date to customers at dynamic time with minimum cost and time.

Following subsequent sections explanations on technology used, System design, Architecture, Query and results, implementation, comparison with the existing system, sequence flow of control between different entity involved in processing the data, results of prototype implementation and finally conclusion.

II. SYSTEM DESIGN

The proposed concept requires building the mobile database at the organization's side to serve the requests of the clients. The android mobile phone provides the SQLite (mobile database SQLlite) database. All the data will be stored in it. In any smart mobile phone system there exist some methodology for notifying the system events like incoming SMS, call etc… (For example- Broadcast receivers [3] in android smart mobile phones). A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Applications can also initiate broadcasts—for example, to let other applications know that some data has been downloaded to the device and is available for them to use. Although broadcast receivers don't display a user interface, they may create a status bar notification to alert the user when a broadcast event occurs. More commonly, though, a broadcast receiver is just a "gateway" to other components and is intended to do a very minimal amount of work. For instance, it might initiate a service to perform some work based on the event.

Before using the service provided by data server customer has to register for service provided by the server. While implementing mobile data server we will be using broadcast receiver to invoke the computation module upon receiving the message from customer, the computational module is invoked only if the incoming message is intended for information retrieval from registered customer. After receiving the incoming text message, the content of the message is read by application and it prepares the data for reply, reply can vary based on the text present in incoming message.

III. TECHNOLOGY OVERVIEW

- *Android manifest*

The manifest file glues everything together. It is this file that explains what the application consists of, what all its main building blocks are, what permissions it requires,and so on

The foundation for any kind of mobile application is the manifest file: AndroidManifest.xml in the root of project For a simple application, offering a single activity and nothing else, the auto-generated manifest will probably work out fine, or perhaps require a few minor modifications

<manifest xmlns:android=http://schemas.android.com/apk/res/android package="com.commonsware.android.search">
</manifest>

- *Broadcast Receiver*

A broadcast receiver is an Android component which allows registering for system or application events. All registered receivers for an event will be notified by Android once this event happens Broadcast Receiver extends the BroadcastReceiver class and which is registered as a receiver in an Android Application via the AndroidManifest.xml file(or via code). Alternatively to this static registration, one can also register a Broadcast Receiver dynamically via the Context.registerReceiver() method [3]. If the event for which the Broadcast Receiver has registered it receives a Broadcast Intents from the Android system in its onReceive() method.

IV. ARCHITECTURE

Following figure 1 shows the proposed system architecture for portable mobile data server. It consists of two sides, one is client or customer side and another is organization or server side. The client side consists of only the basic mobile phone with the text message sending facility, and server side consists of components like broadcast receiver, SQLite database and business logic.

In order to use the service provided by broadcast receiver, the business logic has to register with the broadcast receiver to the event in which business logic interested in (for this application business logic has to register with the incoming SMS event. After registering the business logic will be invoked upon receiving the incoming message). Various processing steps are explained as follows (in figure 1).

1) Registered Client sends the data request to smart phone data server (using pre defined keywords)
2) Upon receiving the incoming message, the broadcast receiver will check the registered broadcast events list
3) If broadcast receiver finds any registrations for incoming SMS it will notify the registered application.
4) After being notified by the broadcast receiver, the business logic checks the content of incoming message
5) Arrows 5, 6 and 7, Business logic checks registered users list in mobile database for registration details, against the incoming user SMS. If user is registered with the server earlier, further computations will be carried out if not the incoming SMS will be ignored.
6) 8, 9 and 10, the business logic will compute the text to reply using the business data (data stored in database).
7) 11 and 12, the prepared text data using user query is sent to client.

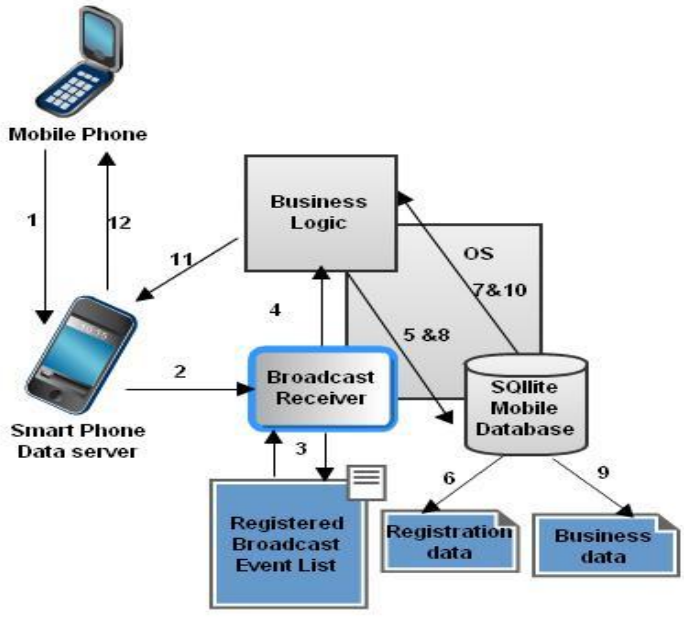

Figure1. Architectural Diagram

V. QUERY AND RESULTS

Queries are sent from the client and the results are sent from the server as results to corresponding queries from the clients. Following will gives the sample list of the query and the response message formats which are used in implementing this concept.

For prototype purpose the request and response query types are limited to following 4 categories, and category of the query can be extended to any numbers.

1. Registration query
Client: REG <space> <mobile _ number>
Server: 1. if user is already registered- user already exist.
2. If user is new – Registration successful

2. Cost query
Client: COST <Product _ code>
Server: cost of particular product if any product with requested product code exists.

3. Cost query
Client: COST <Product name><Space><Quantity>
Server: Returns value corresponding to (cost of item corresponding to requested product code* quantity).

4. Branch list
Client: Branch list
Server: List of available branches

VI. IMPLEMENTATION

A broadcast receiver is a class which extends Broadcast Receiver [4] and which is registered as a receiver in an Android Application via the *AndroidManifest.xml* file (or via code). Alternatively to this static registration, user can also register a Broadcast Receiver dynamically via the *Context.registerReceiver ()* method. This class will be able to receive *Broadcast Intents*. *Broadcast Intents* can be generated via the *Context.sendBroadcast()* method.
The class *BroadcastReceiver* defines the *onReceive()* method. Only during this method your BroadcastReceiver object will be valid, afterwards the Android system can recycle the BroadcastReceiver. Therefore you cannot perform any asynchronous operation in the *onReceive()* method.

VII. COMPARISON

This section makes the comparison with the existing data server and proposed data server

Table 1: comparison with existing system

	Existing	**Proposed**
Requires computer server	Yes	No
Cost of implementation	Too high	Too low
Portable	No	Yes
Physical connection	Requires	Does not require
Internet	Requires	Does not require
Multimedia data	Yes	No
Response Time	High	Low
Power Requirement	High	Too Low
Space Required	Large	Negligible

VIII. RESULTS AND OUTCOMES

To check the load on the smart mobile phone which implementing the server prototype model, the request for data was made to serve the more than 1000 data requests. A Data request array with different request query string is created to consist more than 1000 request. Prototype model serves more than 1000 data requests with no wrong results. The study of raw output during model execution, we observed that, configuration and registration of broadcast receiver plays a very major role, when different user has same query string to server then the server serves the data request with only one computation by caching or storing at least previous 10 data requests. Further, this evaluation addresses the comparative result Data server and mobile phones. This comparison made through the evaluation of proposed model and mechanism and analysis and study data server and mobile phones.

As we know that most of the small shop vendors have normally have mobile phone, work proposed to implement the data server with mobile phones and to introduce the dynamic interaction on demand with the customers .Proposed mechanism gives the standard to implement mobile phone as the data server. This includes GSM gateway to query and result via SMS. No doubt, it also includes the small cost which is affordable.
The following figure 2 reveals the main menu of the test bed implemented at the server side. It consists of the register users option, adding a new item into a database built for the vendor where the server has been implemented, updating database & help contents. Each & every icon specified performs a specific function & the value changes for every update.

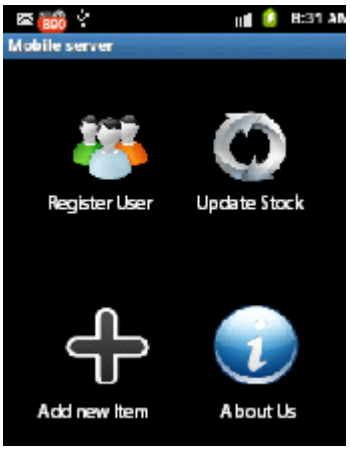

Figure 2: Mobile server with different options.

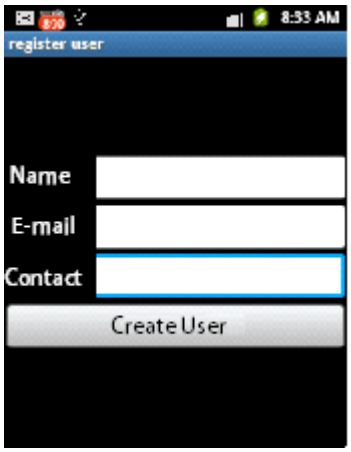

Figure 3: Register user.

The above figure3 reveals the user registration menu, where the regular customer s details can be added into a database which helps the vendor to identify the user while performing the transaction .Every user must register themselves to the data server before performing any transaction.

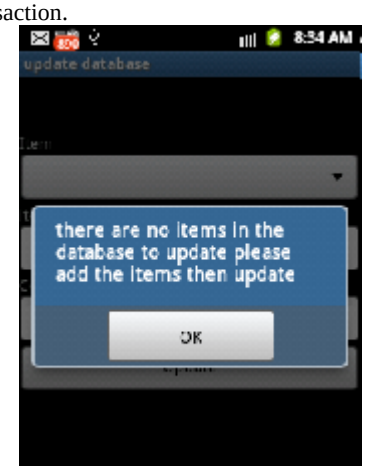

Figure 4: Updating the database with no items.

The above fig. 4 and following figure 5 reveals information about a vendor performing an update operation on the existing item consisting of new cost and the item code respectively. The vendor can perform this update asynchronously depending upon the change in the market price.

This figures also infers that there are no items in the vendors database yet and asking for vendor to update the database by pushing new items into it.

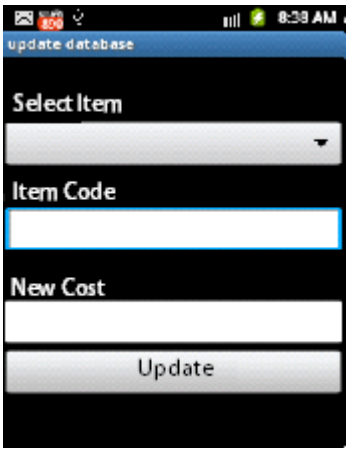

Figure 5: Update existing Item Database.

The following fig. 5 provides a view of vendor adding a new item into the database consisting of item name, item code along with price. This adds information of the new item into database.

Figure 6: Adding new item.

The following fig. 7 shows the user registered already with the data server of a store, requesting an item by sending a message in the format *Item:Rice* to the mobile data server. The user side device can be of any type (android /non-android).

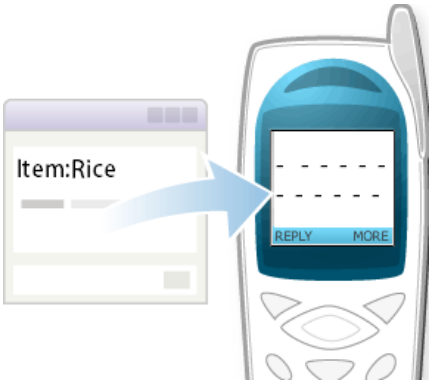

Figure 7: User Request

The above fig. 8 reveals the data sent by the mobile data server in the form of text message after receiving the query from the current registered mobile user ,consisting of the item ,its cost on the current date .

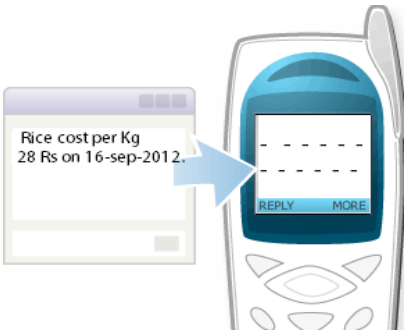

Figure 8: Response from data server

IX. CONCLUSION

This paper aims at providing the data server with low cost, portable and least maintenance. The proposed architecture does not require internet, and this works good with any type of mobile phone at customer's side. The proposed concept does not require any huge investment and hence it is suitable for small business organizations (this proposed method can be easily implemented with amount of 5,000 – 10,000 the major requirement is a smart mobile phone).

The further enhancement of this proposed concept can be done by extending the data transfer to multimedia data transfer using the same proposed concept. This concept can further be extended with the fast data portable servers with the additional hardware support.

REFERENCES

1. Stockus, A. Lab. d'Informatique etd'Imagerie Industrielle, Univ. of La Rochelle **Accessing to spatial data in mobile environment,** Web Information Systems Engineering, 2001. Proceedings of the Second International Conference.
2. Li Ye-bai Coll. of Inf. Eng., North China Univ. of Technol., Beijing, China Zhang Bin ; Wang Hai-bin , Study and Design on Data Management Model of SQL Server CE for Mobile Application, in e-Education, e-Business, e-Management, and e-Learning, 2010. IC4E '10. International Conference on
3. http://developer.android.com/reference/android/content/BroadcastReceiver.html
4. http://www.serverwatch.com/sreviews/article.php/3937296/11-Server-Apps-for-Android-OS-Devices.htm
5. http://en.wikipedia.org/wiki/Server_(computing)
6. en.wikipedia.org/wiki/Serve