

ASHESI UNIVERSITY COLLEGE

ONLINE RESTAURANT ORDER SYSTEM

By:

SALIH AL-HASSAN

Dissertation submitted to the Department of Computer Science,

Ashesi University College

In partial fulfillment of Bachelor of Science degree in Computer Science

APRIL 2010

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:.....

Candidate's Name:.....

Date:.....

I hereby declare that the preparation and presentation of the dissertation were supervised in accordance with the guidelines on supervision of dissertation laid down by Ashesi University College.

Supervisor's Signature:.....

Supervisor's Name:.....

Date:.....

ACKNOWLEDGEMENT

I am very thankful to my supervisor, Dr. Nathan Amanquah, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I would also like to give thanks to my mother Dr. Ramatu Al-Hassan for all the needed support given.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

ABSTRACT

The use of computers has become one of the most common activities in Ghana today. With the wide variety of possible uses that exists for these complex electronic systems, it is no wonder that most of the commercial industries that exist today implement its uses one way or another. There however exist other alternatives to PC use which if explored could pave the way to greater productivity.

In most restaurants that exist today, various methods are used to receive orders from customers. One of such modern day methods is the use of an online medium through which orders are made by customers through the internet, from any location. With this method, orders are normally received and managed at the restaurant on desktop computers. Using desktop computers to manage these orders however often come along with problems such as the high cost involved with the acquisition of computers, maintenance of these computers and of electricity used to power these computers. Some employees also sometimes use these computers for unproductive activities such as playing games, chatting online and sometimes viewing pornographic sites.

This document reports on a project undertaken to develop an alternative system that could replace the desktop computers used to manage orders in such restaurants and also curb the problems that come with using desktop computers. It provides documentation on how the designed system works,

the architecture behind it, and the procedure undertaken to arrive at the final product.

Contents

ACKNOWLEDGEMENT	iii
ABSTRACT	iv
CHAPTER 1	1
1.1 INTRODUCTION.....	1
1.2 BACKGROUND.....	2
CHAPTER 2	7
2.1 METHODOLOGY	7
2.2 APPROACH.....	8
Functional Requirements	8
Non-functional Requirements	10
Detailed Description Of System Functionality	10
2.3 DETAILED SYSTEM DESCRIPTION	15
System Architecture Design	18
Interface Design.....	19
Database Design	20
Choice of Database Management System (DBMS).....	21
Tables.....	22
CHAPTER 3	24
3.1 IMPLEMENTATION.....	24
Implementation of different modules or System Units	25

CHAPTER 4	29
4.1 RESULTS	29
Integration and Testing	29
4.2 CONCLUSION	30
APPENDIX	32
REFERENCES	36

CHAPTER 1

1.1 INTRODUCTION

This project was inspired by the urge to introduce Ghanaians to alternatives to the desktop computer. The project aims to do this by taking a system that would normally require the use of a desktop computer and creating an alternative system that substitutes a PC component within this system with an alternative device.

The system that will be re-created is a system for managing orders that have been placed online by customers for a restaurant. Traditionally, the orders would be managed using a desktop computer connected to a server with a database for storing the orders.

Obtaining these desktop computers is often expensive, costing from GHC400 to over GHC1000. An even larger cost is incurred when a server is required in addition to the computer for receiving orders. It will mean acquiring more than one computer system and as such paying more than double the amount of purchasing a single desktop computer. Other costs incurred for using desktop computers include maintenance costs and the cost that arrive with electricity usage. All these costs put together creates an expensive system. Using these desktop computers also leads to less productivity from employees placed in charge of managing the orders. Most often, these employees spend time that should be spent on the job, playing games, watching videos, and raising the cost of internet access by browsing unnecessary websites such as social networking sites and sometimes even pornographic sites.

The task ahead is to create the same kind of system or an even better one without the above stated problems and without the use of the traditional desktop computer for the purpose of receiving orders. Such a system would help reduce cost and increase productivity.

Various kinds of technologies available today will be explored to help make the project a success and to prove that alternative methods do exist that could easily serve as substitutes to the desktop computer in Ghana.

1.2 BACKGROUND

No one is certain about exactly when research into creating systems to work as alternatives to the desktop computer (PC) began. It is therefore believed that research into such systems began with the invention of the laptop. Once again, it is unclear about who were the first developers of these portable or mobile computers.

To some, the first design was in 1970 by William Moggridge, who worked for Grid Systems Corporation (About.com, 2010). His designed laptop was referred to as the Grid Compass and was used by NASA on space shuttle programs (About.com, 2010).

Others believe that the first fully functional laptop was developed in 1983 by Manny Fernandez. His design was referred to as the Gavilan Computer (About.com, 2010).

Another suggestion as to the first laptop developed, is a laptop referred to as Osborne 1 (About.com, 2010). This particular system was designed by an ex-book

publisher called Adam Osborne in 1981 (About.com, 2010). Most historians consider this to be the first true portable laptop computer (About.com, 2010).

Despite the uncertainty about who developed the first laptop, it is quite evident that these developments paved the way for research into the development of the smaller, more effective devices that exist today.

Today the demand for mobile devices has increased substantially. Over 3 billion cellular phones alone exist today as compared to 1.2 billion computers (Dunphy, Lamont, Zupnik, & MacPherson, 2009). People are clearly making a move to the smaller, more efficient, mobile substitutes for computers.

One could blame this migration on the current high cost of energy, or the increasing need to access information at any point in time. But no matter what the reason may be, the fact still remains that these devices are taking over the computing industry. In countries such as the United States of America (USA) and England, the use of these PC substitutes plays a major role in the lives of the people. Not only are individuals using them but various industries as well (smartcomputing.com, 2000, July).

In North America for example, the police service have switched from using computer terminals that have been hardwired to their vehicles, to using smart phones, specifically the Blackberry, to serve as mediums for which information about people arrested on patrol can be cross checked in the central database (Blackberry, 2010). This provides the police force with a secure, cost effective mobile solution to the service they provide and as a result helps them be more productive. Just like this scenario, other situations have developed over the years, in which the use of PC alternatives has proved to be a more effective solution.

In Ghana, there seems to be very little awareness of the capabilities of these PC alternatives. People still seem to be glued to the notion that without a PC, data processing is impossible. From basic mathematical calculations to browsing the internet, the PC always seems to be the only choice for most Ghanaians. It is common to come across people with highly sophisticated mobile devices, but who seldom have any idea about the capabilities of these devices. The potential of mobile devices are at their maximum when they are used alongside service providers that provide support for their features. Hence, with the current variety of services provided by mobile communication service companies such as MTN, Vodafone and Tigo, there is no limit to the use of smaller devices as PC alternatives.

To solve the proposed problem of finding an alternative method to receive and process orders made online by a customer in a restaurant, instead of the conventional use of a desktop computer, one of the possible solutions would be as follows:

Develop an electronic device that performs the specific functions for receiving and processing incoming orders. This device will be interfaced with the server through one of the ports (parallel port, serial port, universal serial bus port etc.). Some form of input and output system will exist on the device to receive orders from the customer making the orders online as well as send notifications to the customer. Creating this kind of system will require knowledge of electronics as well as knowledge of some form of high level programming language capable of accessing

the computer's hardware resources. Examples of such languages include C and C++. Although this method may appear to be a good solution to the problem, it comes with a few problems:

- a. Such a system would still require the device to be constantly connected to a power source because of the possible demand for extra power and hence would still lead to high costs. Also, it will require some form of cable connection to the port of the server. These cables running through the restaurant would become clutter especially if long cables are used.
- b. If the parallel port is used for interfacing the device with the server, which will most likely be the case because it is the easiest port to use when interfacing, it would require that the cable used to connect the server and the device be relatively short since signal transmission in parallel transmission is limited to short distances. This will mean that the server would possibly have to be positioned within the ordering area of the restaurant, once again creating clutter.
- c. Creating such a system would require a lot of electronic parts, which could be quite costly to the developer.
- d. Upgrading the system when necessary would be an expensive task and would usually require that a completely new device be created.

The solution I propose is to create an application that runs on an existing mobile device, specifically a cellular phone with the added functionality to access the internet through GPRS and/or the ability to access wireless networks. The cellular phone will be interfaced with a main server through GPRS or a Wireless Local Area

Network (WLAN). The server has the responsibility of holding the database, serving as the central order processing unit as well as serving as the medium through which the system gets access to the internet. The application created will be programmed using Java Programming Language (reasons for using Java will be stated in the next section) and will be created to possess the features required to receive orders and process orders.

Reasons for deciding to use this method include:

- a. The ability to interface the phone with the server wirelessly eliminates the use of cables within the kitchen and as such removes the possibility of clutter.
- b. Creating an application for an existing mobile device brings to the fore the possibility of including several extra features and design options to the system, unlike with the method of creating an electronic device from scratch, which may limit functionality such as displaying information about orders and holding a large number of orders at a time.
- c. It is a much more inexpensive method. The most expensive part of creating this kind of system would be the cost of acquiring a phone with all the features previously stated which most phones are capable of. Developing the software for this system would be relatively inexpensive. Upgrading and maintenance of such a system would be even much less expensive compared to updating a customized device.

CHAPTER 2

2.1 METHODOLOGY

The software design methodology used in the design of the system is the "Structured Systems Analysis and Design Methodology (SSADM)". This is an approach which requires that the various modules and stages of the software system being designed be specified prior to the start of the design process. This methodology makes use of the waterfall model and relies largely on data flow diagrams. With the use of this methodology, the software system aims to fully and effectively address the following issues:

- a. **Performance:** The system should be designed to function effectively and accurately as often as possible. Slow response of the system will be avoided as much as possible.
- b. **Coupling:** This is the measure of the interdependence among software modules. Considering the different modules that exist within the entire system and the requirement for these to communicate and transfer information with each other, it is very important that this measure is high within the system.
- c. **Robustness:** The system will be designed to tolerate inputs of various kinds made by users of the system and will as much as possible be designed not to crash when invalid inputs are made. An example of such a situation will be someone entering letters and characters into fields that require numbers.

- d. **Re-usability & portability:** The system will be designed to ensure that it can handle movements from one location to another with ease. Setup of the system should be easy and straight forward. Its components should as much as possible possess the ability to be reused for other purposes.
- e. **Error, Exception Handling & Fault Tolerance:** The system should be able to handle errors effectively, with a specific description to each error that may occur. It should also possess the ability to easily recover from system crashes.
- f. **Security:** The system should ensure that only people with the right permissions are given access to certain sections. Outsiders that are not registered on the system should have no access.

2.2 APPROACH

Functional Requirements

Functional requirements of the system are as follows:

- a. **Receive Orders:** The system should be able to receive orders made by customers
- b. **Approve Orders:** The system should be able to approve orders made by customers
- c. **Disapprove Orders:** The system should be able to deny orders made by customers

- d. **Manage Users:** The system should be able to manage all users of the system.
- e. **Security:** The system should be able to provide secure access to users of the system.
- f. **Manage Customers:** The system should be able to manage customers of the system and provide a communication medium between customers and administrators.
- g. **Make Orders:** The system should allow customers to make orders from offsite.
- h. **Tracking:** The system should be able to track order information of orders that have already been executed.
- i. **Menu Management:** The system should allow the management of the food menu.
- j. **Comments and complaints management:** The system should allow management to view anonymous comments and complaints made by customers on services rendered.
- k. **Notification:** The system should enable management send customers notifications with information concerning promotions, new developments, and other details management would like customers to know.
- l. **Freezing accounts:** The system should enable management temporarily disable customer accounts. This would be required for customers who default on orders.

- m. **Specifying customer location:** The system should allow customers who would require delivery, accurately specify their location to ensure that orders are delivered accurately, quickly and easily.
- n. Should be able to alert customers of decisions made on their orders.
- o. Should be able to notify users of errors in data input

Non-functional Requirements

Non-functional requirements of the system are as follows:

- a. The system should use as little electrical power as possible.
- b. The system should provide a reliable medium for the exchange of information.
- c. The system should involve a maximum of one computer in its implementation.
- d. The system should be implemented without the need for the existence of a computer in the order receiving area.

Detailed Description Of System Functionality

The system consists of three major components or areas. These are:

- a. **Mobile application:** The mobile application will be developed using the Java 2 Micro Mobile Edition Language. It will be interfaced with the server over a WLAN or the internet and will constantly communicate with it. The application will be developed to handle the following functions:
 - i. Setting up access to server: To ensure that the system can easily be setup on a network, there will be a setup section that lets the user type in the

location of the associated files on the server on which the web application will be setup. If the location does not exist, an error message is displayed when the user tries to login. If the location is right, it is stored on the phone, and will stay there as default until the server or network needs to be changed. A detailed description of how this will function will be given later in the report.

- ii. Login: There will be a login section that ensures that only employees with the duty of managing the orders are given access to the system.
- iii. Receiving orders: Orders made by customers will be displayed on the application's interface in the form of a combination of the Order Number and the Item Ordered. The application will alert the employee of the received order by outputting a sound. Monitoring of incoming orders will be done using a thread that will be given the responsibility of constantly checking the database for changes. A defined change in the database state will indicate an order made.
- iv. Approving orders: Orders received will have two possible actions that the user could perform on them. One of these is to approve the order. Approving the order will send an indication to the customer that the order has been received and will be processed.
- v. Disapproving orders: The other possible action that the user could perform on a received order will be to disapprove the order. This may be done in exceptional circumstances such as when a particular item has been sold out for the day or when the destination for a particular delivery is out of the restaurant's range of access. Disapproving the order once again sends an indication to the customer indicating that the order made cannot be filled.

- vi. Alerting the customer of a completed order by Short Messaging Service (SMS): When an order has been completed, the user should be able to indicate or mark the order as completed. Doing this should automatically send a message to the customer's cellular phone by SMS, telling him or her that the order is ready for pickup.
- vii. Printing a map through a Bluetooth enabled printer: If the order is supposed to be delivered, instead of sending a message to the customer, the system will print out a map through a Bluetooth enabled printer, with a marker on the map pointing to the customer's current location. This will help easily locate the customer when delivering. Another method would be to provide cellular phones to employees in charge of delivering so that they can view the maps on the on them. This would however be an expensive method to implement.
- viii. Printing a receipt through a Bluetooth enabled printer: When an order is picked up by a customer, the user of the system indicates that the order has been picked up by the customer and this will once again automatically print out a receipt through a Bluetooth printer. The receipt will help customers verify that what they order is exactly what they are getting, and also that the price they are being asked to pay is accurate.
- ix. Providing a report: The application will be able to generate a report at any point in time that gives details of the day's orders, including the time of order, the type of order made, and the customer id or customer name of the person that ordered it. Accounting information such as the total number of orders, the total revenue etc. will also be available to employees. Information such as

the number of different orders coming from a particular area will also recorded.

b. **Web Application:** There are two main kinds of users that have access to the web application. These are the managers and the customer. Each kind of user is directed to a corresponding section when logged in.

- **ADMINISTRATOR SECTION:** This section comes with the following functions:
 - i. **Add users:** This allows the administrator to create any of the three kinds of users for both the mobile application and the web application. (customer, manager, order manager).
 - ii. **Edit user details:** Allows the administrator make changes to the details of users in any of the user categories.
 - iii. **View users:** This displays all registered users on the system and provides the ability to search for a specific user.
 - iv. **Remove users:** Allows the administrator to permanently remove a user's account existence.
 - v. **Customer Management:** This provides the administrator with the ability to perform various functions pertaining to the customer. These functions include, sending notification messages to customers, freezing customer accounts and viewing comments and complaints made by customers.
 - vi. **Add menu item:** This gives the administrator the ability to add items to the current food menu on display.
 - vii. **Delete menu item:** This gives the administrator the ability to delete items currently on the menu.
 - viii. **View menu item:** This displays all items in the current menu.

- **CUSTOMER SECTION:** The main functionality that this section provides is providing the customer with the ability to make an order. Making an order involves the following steps:
 - i. View the menu and decide on the item to order.
 - ii. Make the order.
 - iii. Customers specify if they would require the order to be delivered to them. If it is to be delivered, using Google maps, the user will specify their location on the displayed map, and submits the order. Customers will also be able to specify their locations on their profiles at specific times so that whenever they make an order they will not have to always specify their location on map. Specifying the location on a map will be ideal for this since accurate addressing in Ghana is a challenge.
 - iv. The customer receives an order number which will be used to verify their order.
 - v. If the item selected cannot be prepared, a notification is received and the customer has to select a different item.
 - vi. Customers can leave any comments and complaints they may have about services received anonymously.
- c. **Database:** The database serves as a communication medium between the web application and the J2Me application. By monitoring changes in the database state, both the web and J2Me applications are able to provide notifications and communicate significantly close to real time. It also holds records of customers, employees, orders etc.

2.3 DETAILED SYSTEM DESCRIPTION

The data flow diagram in Fig. 1 shows how the flow of data will occur within the web application and the conditions necessary for this flow.

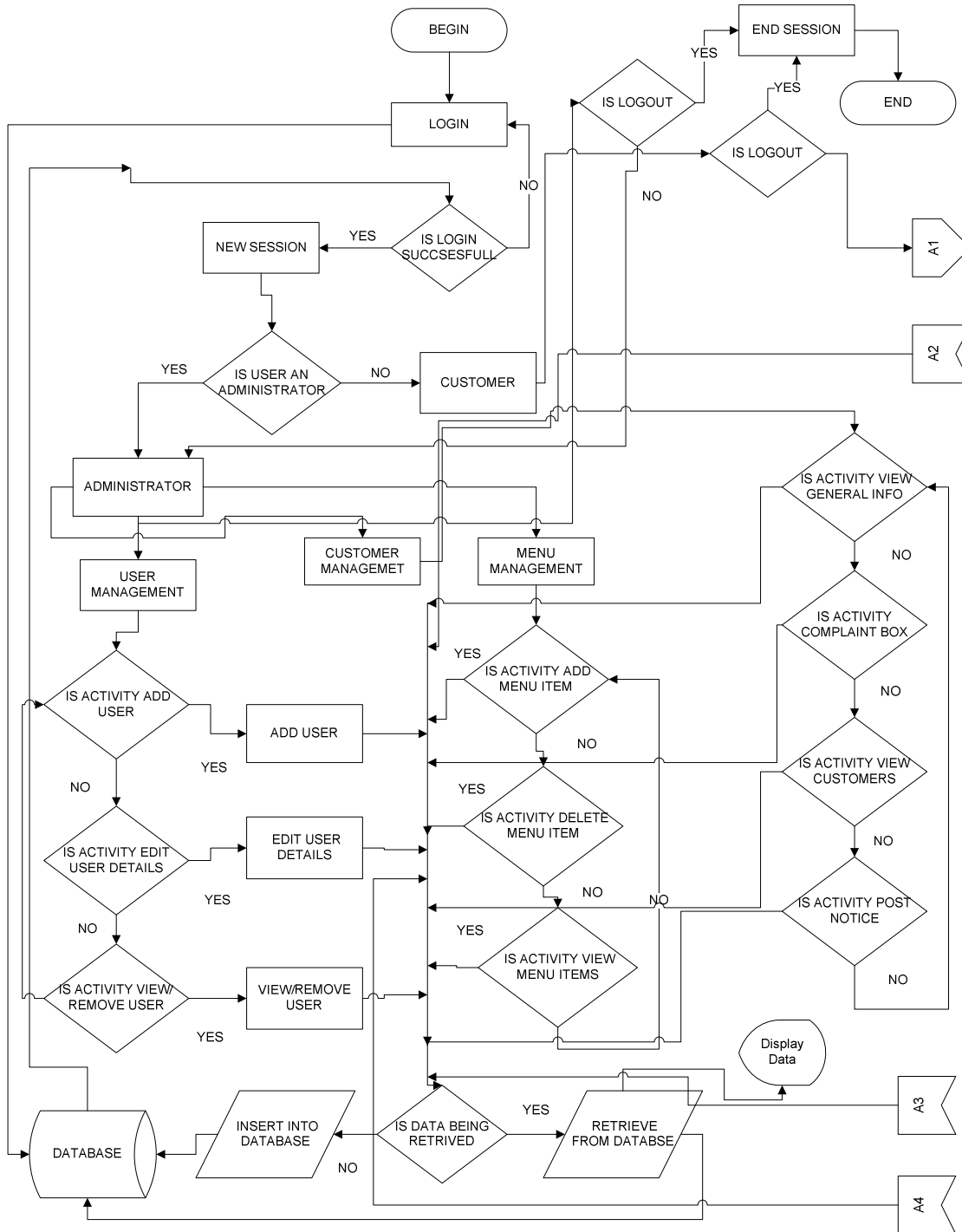


Fig. 1

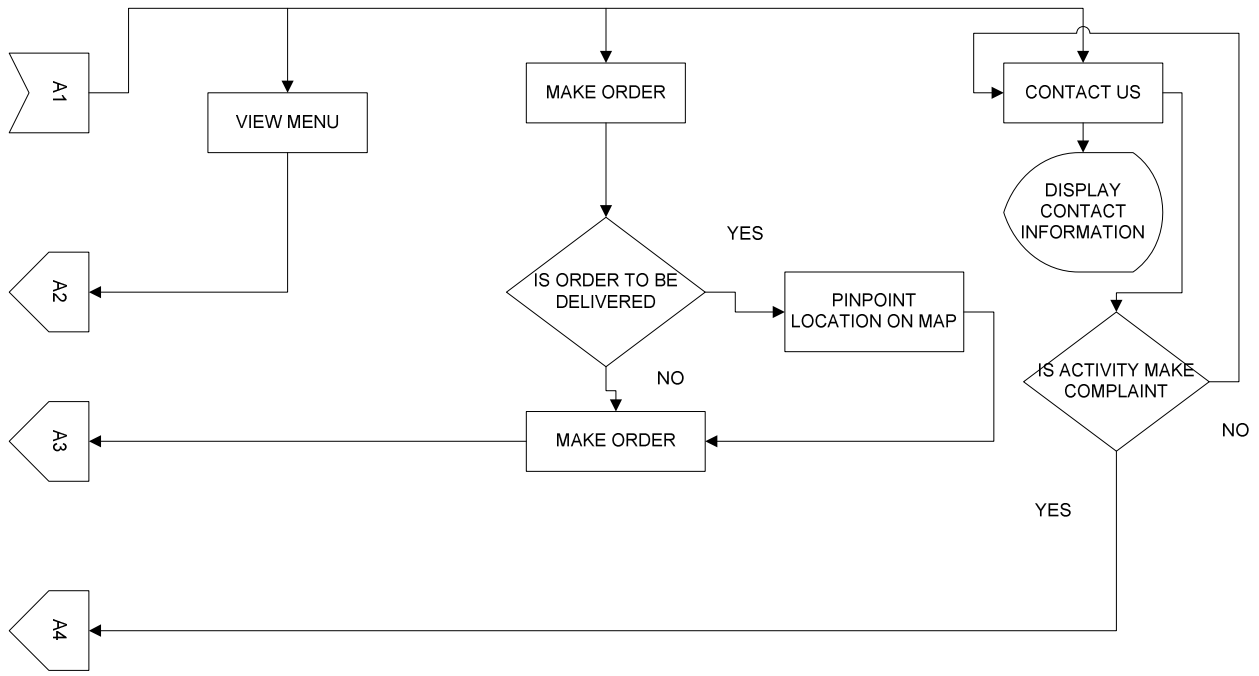


Fig. 1 (cont.)

The data flow diagram in Fig. 2 shows how the flow of data will occur within the mobile application and the conditions necessary for this flow.

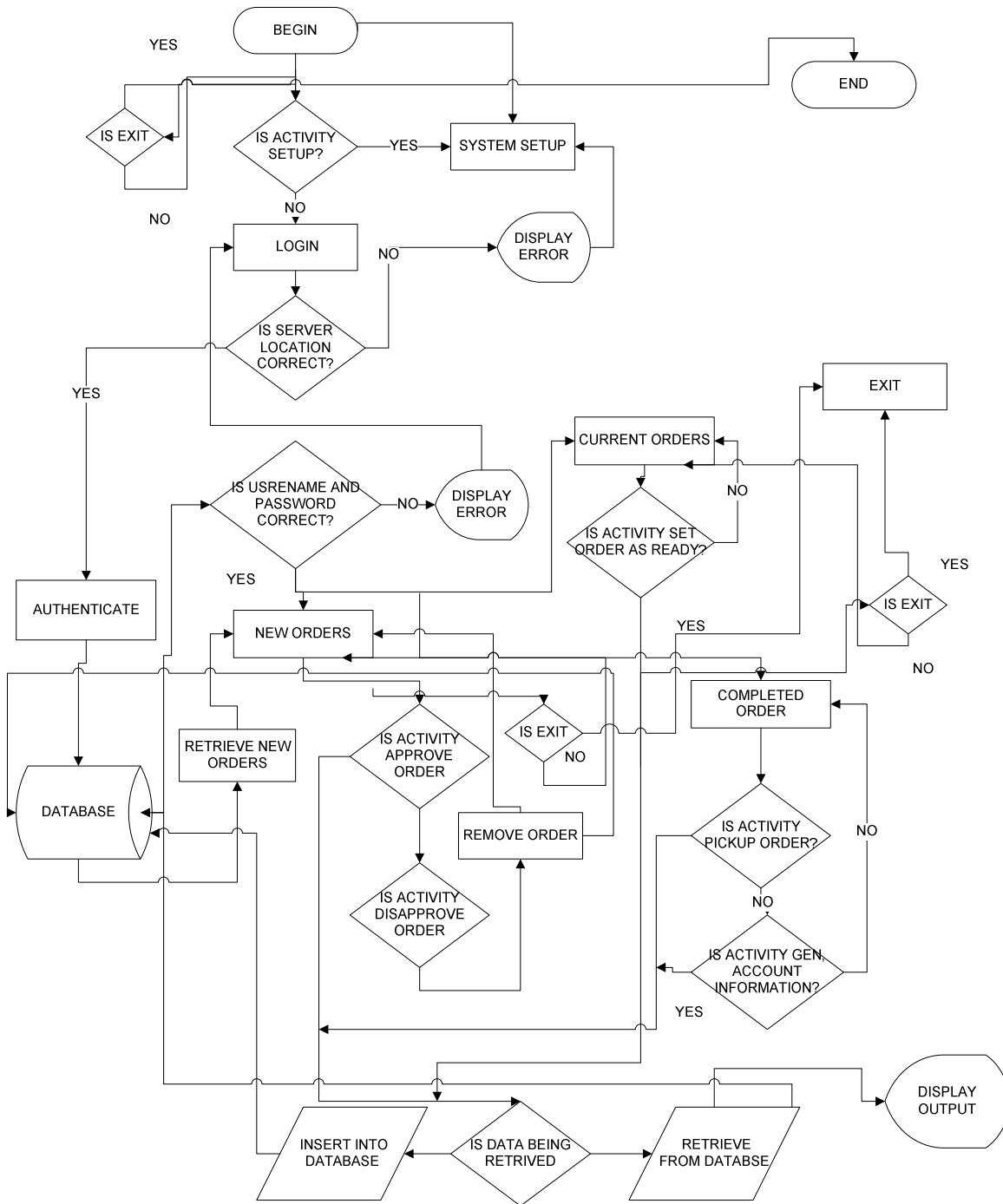


Fig. 2

System Architecture Design

The Software system will be created based on the following diagram.

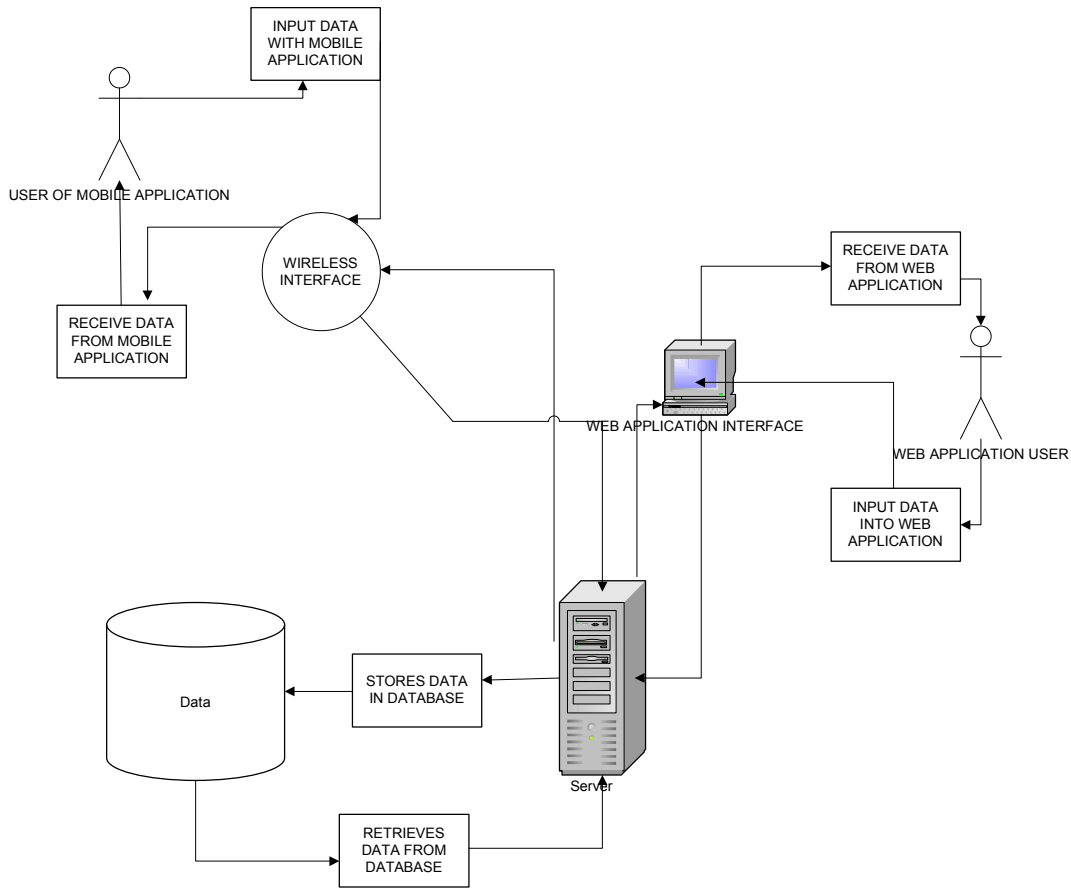


Fig. 3

According to the diagram, data input from either the mobile application or the web application will be transmitted to the server. The server will process the data and if necessary store it into the database. A response will be transmitted back to the corresponding source of data input if necessary. Likewise, if some particular kind of information is needed, the server is accessed via either the mobile application or the web application; a request is made to the server and then data is retrieved and

sent back to the user. Transmission of data to and from the server by the mobile application will be done via a wireless medium.

Interface Design

To create an interface for the web application that could be interacted with effectively, it was important that the right colors were used to ensure proper visibility. The clarity of the text on the pages was ensured as well as alignment of text fields and labels. The interfaces of all user categories were created to be similar in appearance. The look, feel, and layout of the mobile phone application however depended on the J2ME implementation on phone that was used. Each phone type had a different implementation of the J2ME interface. Figure 1 and Figure 2 (See appendix) show some of the images that represent each user category's interface on the web application.



Fig. 4 – Example of Administrator interface



Fig. 5 – Example of customer interface

Database Design

The main operations that are carried out by the system rely extensively on the availability of data which is either yet to be stored in one location, or already stored in that location. Modern trends have shown that most reliable forms of data storage depend on a database of some sort. These are evident even from data storage giants such as Google, Yahoo and MSN (including Bing). All these entities use databases in their daily transactions and still use it till date because it works for them.

This software system, also being data dependent, and seeking a need to be secure, fast, efficient and reliable, also uses a database. The majority of operations carried out involve data insertion, data retrieval and data editing/updating. These are all

operations that are best maintained by a database and as such find it necessary to use one.

Choice of Database Management System (DBMS)

Restaurant operators are usually cautious when it comes to spending money. In an effort to create a system that is less of a strain on the finances of a restaurant's management, it is important that minimizing cost comes first.

As such, to ensure that the system developed would pose little problems when being financed in terms of acquisition, installation, running and maintenance, it was evident that MySQL would be the best choice when it came to selecting the appropriate Database Management System.

The commercial Enterprise version has no up-front licensing costs, which means lower capital outlay. It is much lower especially when compared to high-end proprietary databases like Microsoft SQL Server and Oracle, who have a per-CPU license cost anywhere from \$25,000 to \$40,000 (Mysql, 2010). Support for the Enterprise Edition also costs less, leading to savings of up to \$5000 annually (Mysql, 2010).

Hardware requirements for MySQL are also very modest and as such can be run on a variety of systems. The one caution however is to ensure the availability of RAM since all live data storage is done in memory.

MySQL is also very well supported with a dual-licensing scheme that provides access to support directly from the creator of the database. MySQL AB provides support and maintenance services as part of its roughly \$3000 annual subscription

fee (Mysql, 2010). A separate fee gives customers priority 24 hours a day, seven days a week support.

The scalable nature of MySQL also makes it possible to easily adapt the system to the current needs of a restaurant. The system can be run from very large systems to very small systems the size of a pocket book.

MySQL also comes prepared with support for the most languages such as Ruby, ASP and PHP (sun.com, 2010).

The software application being developed is also implemented in the PHP language and according to the developers of applications of this platform, the speed of the PHP server is optimized when used in unison with MySQL as its DBMS. The level of integration that can be issued when PHP is used with MySQL makes it an obvious choice for use as the DBMS.

All these features within this comparably inexpensive software make it ideal to the application on.

Tables

From the functional requirements and design specifications, relationships were drawn which were used to build the database of the system. The tables that came up as a result are:

- Complaints: Anonymously stores comments and complaints made by customers about services. Complaints are made. It is a stand-alone table.
- Menu: Stores the current list of foods on the menu, along with their prices.
- Notice: Stores notices made by the manager to all or specific customers.
- Order: This contains information about orders to customers.

- OrdersInfo: Contains information about all processed orders.
- Users: Contains information about the users of the application.

The Fig. 6 gives more detail to the above tables.

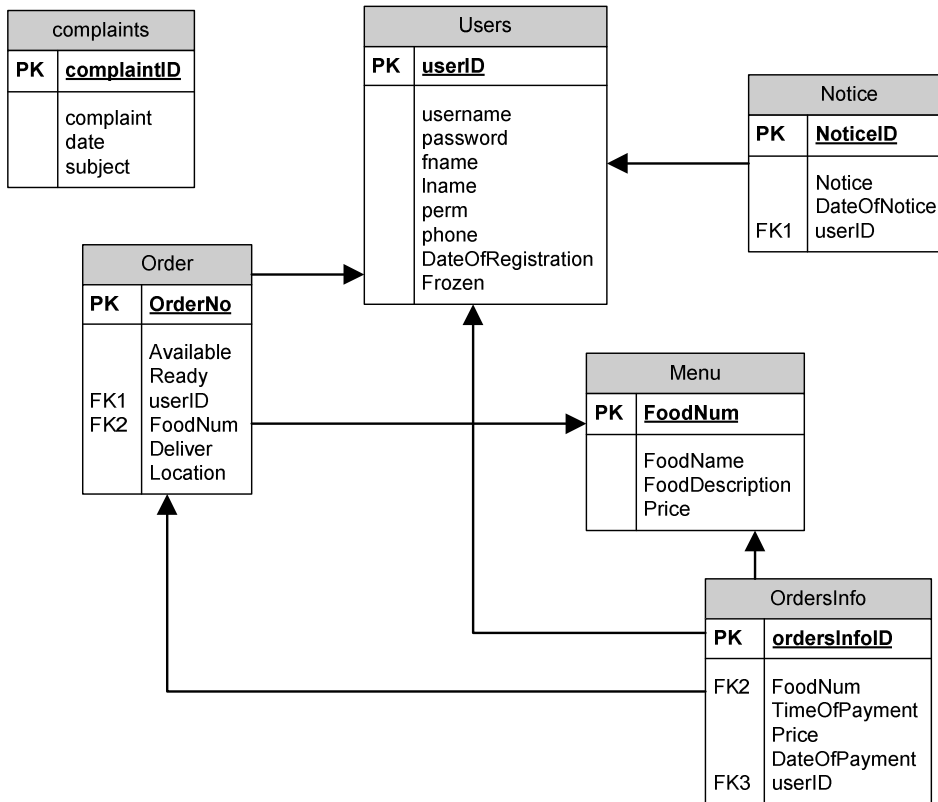


Fig. 6 – Database diagram

CHAPTER 3

3.1 IMPLEMENTATION

The development tools used for the development of the system were

1. Languages

- a. PHP: For server side programming, Mysql works best when used alongside php.
- b. JavaScript: For client side scripting.
- c. HTML
- d. CSS: For designing the interface.
- e. Ajax: This prevents the refreshing of the entire page and as such reduces the load time on a page.
- f. J2Me: Java is present on most mobile phones today. This makes J2me the ideal language to use to create an application specified for mobile phones. By doing this it creates a situation where the application can be run on a variety of mobile phone types.

2. Database

- a. MySQL

3. Utilities

- a. Apache web server
- b. Notepad++

- c. PhpMyAdmin
- d. SqlYog
- e. XAMPP
- f. Adobe Photoshop
- e. Zend Studio for Eclipse

Implementation of different modules or System Units

Database: The database was the first unit to be developed in the implementation process. This was created using SqlYog, an application suited for the development of databases. The database development process began with the listing of the various kinds of data that the system would have to store. These were then grouped to form relations. The relations were then normalized to reduce redundancy and limit the possibility of any anomalies. However, room was made for changes to be made since it was known that during the development of the other units, changes to the database might have to be made.

Web Application Section: The web application section was the next unit to be developed. Ajax was mostly used in its development. Server side programming was done using PHP, while the client side was done using HTML and JavaScript. The first step in its development was to create the login section to ensure that the right users were granted access. Each user's username and password was crosschecked with those in the database and if the match was successful, the user was redirected to his or her designated destination or page; otherwise an error message was displayed and the user was denied access. The Administrator section was then created, followed by the Customers section, then the mobile application section.

The customer section had a feature that required the implementation of Google maps (Google.com, 2010). This feature was to allow customers pinpoint their current location using a map. They then submit the order. The coordinates of the customer's location was then retrieved from the map and stored in the database to be viewed later. See Item 1(See appendix) for the code used to generate the map and to add features such as markers that could be moved across the map.

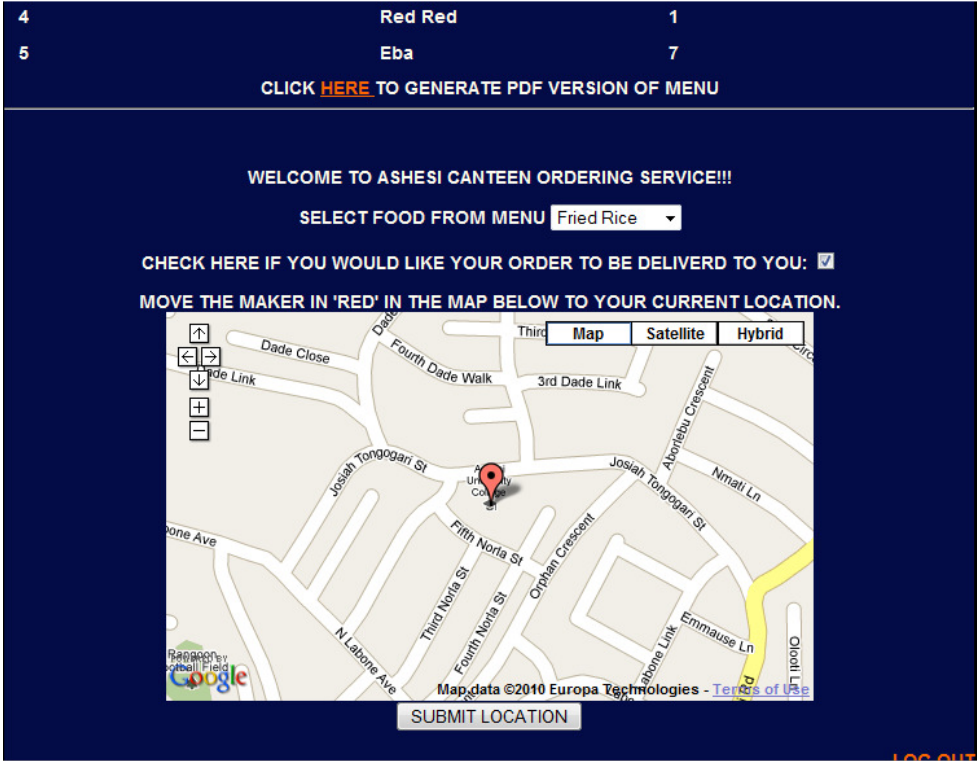


Fig. 7

A system to monitor the database was finally created that would generate an alert whenever a change was made. This was implemented successfully by creating an Ajax function that implemented the timer feature of JavaScript to check the database after specific time periods.

Mobile Application Section: The J2ME application created for the mobile phone was developed using Net Beans. First step of development was to come up with the functions that would be used to send data to the server and also to receive data from the server. Using these functions I was able to implement all the displayable sections that displayed data from the database. I also created a thread that has the responsibility of checking the database and alerts the user with three beeps when an order was received (Fig. 8). The J2ME code for this feature is shown in Item 2 (See Appendix).



Fig. 8 – J2ME application showing recently received order

A feature was also implemented that ensured that the server address for the system was accurate. To do this, at the first system use, the server address had to be specified on the phone. This was then stored in the record store of the phone and set as a default address. Whenever a user tried logging in, the server address

stored in the record store was attached to the URL and if it was invalid an exception was caught and an error message indicating an invalid address was displayed. Most of the other features were implemented successfully.

Some of the difficulties faced were:

1. How customers would specify their location accurately enough on the web application, so that orders could be delivered to them. This was however solved by the use of Google Maps.
2. Another challenge encountered was finding a way to create an alert system that would work in real time and not create a situation that would cause slow functioning of the system. This was overcome creating a multithreading application.

CHAPTER 4

4.1 RESULTS

Integration and Testing

Unit Testing

Web Application: The web application was deployed on a server on numerous occasions and thoroughly tested. This was to ensure that all the various sections of the web application worked well when actually deployed. The testing revealed that some of the queries used in the application did not function when deployed using a different database name. This was quickly rectified. It was also realized that the section of the application that dealt with Google Maps took quite a while to load. This was however blamed on slow internet speed indicating that in order for the system to run effectively, a good internet connection would be required.

Mobile Application: The J2me application was deployed onto a Nokia 5800 for the first testing. No problems were encountered during the deployment process. The application was packaged into a JAR file, and transferred onto the phone. The application was then installed and run for the first time on the phone. It was realized that images that initially looked large on an emulator, looked quite small on the physical mobile phone. This was however not a problem since it did not hamper the application's functionality.

The second phone used for testing the application was a Nokia 5130. Just as in the Nokia 5800, deployment of the application was successful. Testing on this phone revealed larger looking images.

The difference in the appearance of images on both phones was blamed on the difference in the dimensions of the respective screens. While the Nokia 5800 had a larger screen, the Nokia 5130's screen was much smaller, therefore creating the impression that the image on the Nokia 5800 was smaller than the image on the Nokia 5130. Every other feature that was supposed to work both dependent on and independent of the server worked well.

System Testing

The system was deployed on a server and a Nokia 5800 for testing. This was a success during the first testing; no problems were encountered. All functions that were completed at that time worked well, including the feature that allowed the user specify the server location.

User Acceptance Test

Different people were exposed to the system and a few usability issues were recorded and resolved. No major problem was identified.

4.2 CONCLUSION

Not every feature proposed was implemented; the feature not implemented was the ability of the system to print through a Bluetooth printer

However, those that were implemented were successful at performing the functions that a computer deployed for the same purpose would perform.

This project was able to draw attention to the fact that the use of computers is not the only way by which data processing can be done in some industries. In fact, it proves that more efficient methods exist for low level data processing. Methods that are less expensive to implement and at the same time, that provide features that might not even exist for some computers. Based on the targeted objective of creating a cost effective, feature filled, and effective online ordering system for restaurants, it has been demonstrated that cheaper yet effective devices are available that could replace the use of computers in the Ghanaian industries.

The use of technology should not be limited to computers. There is a wide variety of state of the art technological developments that could be explored in the world today. It is suggested that, industries or individuals should make use of these developments. From the project, a complex system was created that would work on mobile phones that are today, referred to as basic. The question now is "How much more could be done if the system was designed to run on much more complex devices capable of doing much more?" Given more time, the next step would be to develop similar applications on much more advanced devices in an attempt to utilize the more advanced features that they possess. Other features that could be added are:

- a. A feature that allows frequent customers specify their location once based on a particular time of day. Hence depending on the time of day an order is made by that customer, the system will be able to determine their location.
- c. Advertising based on data collected on the purchasing habits of customers.

APPENDIX

ITEM 1- Google Map generation code

```
<script
src="http://maps.google.com/maps?file=api&v=2&sensor=false&key=
    ABQIAAAAsHu_kNipzhawM6zLNNuznxT2yXp_ZAY8_ufC3CFXhHIE1NvwkxTSxz6BDEoJn4p
    7rvwX-dVIDFCWsA">

</script>

<script type="text/javascript">
    function createMap() {
        //Checks if the browser is compatible
        if(GBrowserIsCompatible()) {
            document.getElementById('map').style.width='500px';
            document.getElementById('map').style.height='300px';
            //creates a map and places it within the tag with the id
            //'map'
            var map = new GMap2(document.getElementById("map"));
            var center = new GLatLng(5.571994860527281, -0.1731070876121521);
            //centers the map to specific coordinates and at a height //level of 16
            map.setCenter(center, 16);
            //creates the marker
            var marker = new GMarker(center, {draggable: true});
            //creates an eventListener that specifies what should be //done when the
            marker begins to be dragged
```

```

        GEvent.addListener(marker, "dragstart", function() {
            map.closeInfoWindow();
        });

//creates an eventListener that specifies what should be //done when the
marker stops being dragged
GEvent.addListener(marker, "dragend", function(overlay,latlng) {
    if (overlay) {
        map.setCenter(overlay, 16);
        document.getElementById("coor").value=overlay;
    }
});
map.addOverlay(marker);

//Adds controls to the map
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
    }
}
</script>

```


Item 2 - Code for alerting user on mobile application of received order

```
class PrimeThread extends Thread    {
    //gets the total number of orders currently in the database
    int total=canteen.total;
    PrimeThread() {
    }
    public void run() {
        try{
            int tot;
            //Retrieves the total number of orders in the database
            String
f=httpCon("http://" +canteen.ip+"/pro/java/getnumberoforders.php");
            tot=Integer.parseInt(f);
//if the total number of orders retrieved now is greater than the total
//retrieved later it indicates that an order has been made
            if(tot>total){
                String g=httpCon("http://" +canteen.ip+"/pro/java
/getnumberoforders.php");
                tot=Integer.parseInt(g);
                canteen.setTotal(tot);
                //Beeps 3 times
                Manager.playTone(109, 400000, 100);
                Manager.playTone(109, 400000, 100);
                Manager.playTone(109, 400000, 100);
                newOrders=null;
            }
        }
    }
}
```

```
        switchDisplayable(null, getNewOrders());  
    }  
}  
catch(Exception e){}  
}
```

REFERENCES

About.com. (2010). History of Laptop Computers. Retrieved March 20, 2010, from About.com: <http://inventors.about.com/library/inventors/blaptop.htm>

Appleton, B. *A Software Design Specification Template*. Retrieved November 3, 2009, from: <http://www.cmcrossroads.com/bradapp/docs/sdd.html#SEC2>

Aranda, N. (2007, March 27). *A Brief History of Mobile Computing* . Retrieved March 20, 2010, from ezinearticles.com: <http://ezinearticles.com/?A-Brief-History-of-Mobile-Computing&id=505215>

Blackberry. (2010). *Cape Breton Police- Case Study*. Retrieved March 23, 2010, from Blackberry Website: <http://na.blackberry.com/eng/newsroom/success/capebretonpolice.jsp>

BusinessWeek. (2010). *Workplace Computing: PC Alternatives*. Retrieved March 20, 2010, from BussinessWeek.com: http://images.businessweek.com/ss/10/01/0104_workplace_pc_alternatives/index.htm

Computerworld. (2009). *IT managers shop around for PC alternatives*. Retrieved March 20, 2010, from Computerworld.com: http://www.computerworld.com.au/article/341227/it_managers_shop_around_pc_alternatives/

ControlDraw Ltd. (2009). *Software Design Specification*. Retrieved 11 2, 2009, from Defining Requirements for Process Automation:

<http://www.urswriter.com/index.html?SoftwareDesignSpecification>

Dogbevi, E. K. (2009, October 19). *Report forecasts 15 million mobile phone users in Ghana by end of 2009*. Retrieved March 10, 2010, from ghanabusinessnews website: <http://www.ghanabusinessnews.com/2009/10/19/report-forecasts-15-million-mobile-phone-users-in-ghana-by-end-of-2009/>

Dunphy, K., Lamont, M., Zupnik, S., & MacPherson, E. (2009, March 09). *Mobile Computing*. Retrieved March 20, 2010, from <http://people.stfx.ca/nfoshay/Published%20Documents/INFO420/Mobile%20Computing%20Presentation%20lamont%20et%20al.ppt>.

EdrawSoft. (2009). *Flow Chart Definition - Types of flowcharts, How to draw flow chart and guideline for drawing a flowchart*. Retrieved 11 6, 2009, from Edraw Flowchart: <http://www.edrawsoft.com/Flowchart-Definition.php>

Google.com.(2010).Google Maps Apireference.Retrieved March 13, 2010, from Google Code:<http://code.google.com/apis/maps/documentation/reference.html>

GoWorkAbroad. (n.d.). *PDA or Smartphone: Adequate Substitute for a Laptop?* Retrieved March 20, 2010, from [goworkabroad.co.uk](http://www.goworkabroad.co.uk):
<http://www.goworkabroad.co.uk/pda-smartphone-substitute-laptop.html>

King, R. (2009, January 5). *Companies Seek PC Alternatives*. Retrieved March 20, 2010, from BusinessWeek.com:

http://www.businessweek.com/technology/content/dec2008/tc20081231_178982.htm

Ludwig Consulting Services, L. (2008, 8 22). *Requirements Terms and Definitions*. Retrieved 11 1, 2009, from Ludwig Consulting Services, LLC website:
<http://www.jiludwig.com/Definitions.html>

Malan. R, Bredemeyer. D. *Functional Requirements and Use Cases*. Architecture Resources for Enterprise Advantage. Bredemeyer Consulting. Retrieved November 1, 2009, from: http://www.bredemeyer.com/pdf_files/functreq.pdf

Malan. R, Bredemeyer. D. *Defining Non-Functional Requirements*. Architecture Resources for Enterprise Advantage. Bredemeyer Consulting. Retrieved November 1, 2009, from: http://www.bredemeyer.com/pdf_files/NonFunctReq.PDF

Mysql. (2010). *Top Reasons to Use MySQL*. Retrieved March 18, 2010, from Mysql website: <http://www.mysql.com/why-mysql/topreasons.html>

smartcomputing.com. (2000, July). *Break free from the PC*. Retrieved March 15, 2010, from smartcomputing.com:
<http://www.smartcomputing.com/editorial/article.asp?article=articles/archive/G0807/30g07/30g07.asp>

sun.com. (2010). *Mysql-Features*. Retrieved March 18, 2010, from Sun website:
<http://www.sun.com/software/products/mysql/features.jsp>

