

# Digital Ordering System for Restaurant Using Android

Ashutosh Bhargave, Niranjan Jadhav, Apurva Joshi, Prachi Oke, Prof. Mr. S. R Lahane

Department of Computer Engineering,  
GES's RHSCOE

**Abstract-** Nowadays web services technology is widely used to integrate heterogeneous systems and develop new applications. Here an application of integration of hotel management systems by web services technology is presented. Digital Hotel Management integrates lots of systems of hotel industry such as Ordering System Kitchen Order Ticket (KOT), Billing System, Customer Relationship Management system (CRM) together. This integration solution can add or expand hotel software system in any size of hotel chains environment.

This system increases quality and speed of service. This system also increases attraction of place for large range of customers. Implementing this system gives a cost-efficient opportunity to give your customers a personalized service experience where they are in control choosing what they want, when they want it – from dining to ordering to payment and feedback.

We are implementing this system using android application for Tablet PC's. The front end will be developed using JAVA Android and the backend will work on MySQL database.



This project deals with Digital ordering system for restaurant. This topic includes scope of the project, project characteristics, Operating environments, Assumption and dependencies, design and implementation constraints. Scope of the project includes features that can be implemented. Design part includes the method and way of designing the product. It also explains certain constraints on designing and implementation.

## Index Terms-

- DFD: Data Flow Diagram.
- DOSRUA: Digital Ordering System for Restaurant Using
- KOT: Kitchen Order Ticket
- Android
- UML: Unified Modeling Language.

## I. INTRODUCTION

Restaurants are one of the favorite premises. With no regard to the actual reasons for visiting restaurants, customer will make orders and wait for the ordered meals. However, it is common if customers complain for not feeling satisfied about the services offered.

There are many reasons leading to the feeling of dissatisfaction including being entertained late in terms of order taking by the waiter and meals serving. The issue of being late entertained could be solved with help of the advancement in the technologies of communication. In accordance, this study initiates an integrated and networked system, with the focus is on its ability to solve the above described limitations in order taking.

This study names the system as Digital Ordering System for Restaurant Using Android (DOSRUA). In definition, DOSRUA is an integrated system, developed to assist restaurant management groups by enabling customers to immediately make orders on their own selves. This will minimize the number of minutes to wait for the meal serving.

## II. RELATED WORK

The existing system is paper based. The traditional menu cards in the restaurants are paper based. Waiters use paper to write the order of customers. The records are stored on paper. As with anything paper based, it is so easy for things to get damaged by Coffee stains etc, or paper being lost due to fire or accidents or just generally lost. There is wastage of time, money, and paper. As traditional menu cards are paper based, any changes that need to be made in the menu card will lead to wastage. As it will require reprinting of all the menu cards. Also, for small changes it is not possible to print all the menu cards again and again. There is no power to dynamically make any changes in the menu card. To access a particular record from the stack of papers is not efficient. From the customer's point of view, this system is time consuming. As, one has to wait until the waiter comes to take the order, one has to call waiter number of times till he notices it, there can be misinterpretation while the waiter is writing your order on paper, and it might be possible that you are served with a wrong dish. There has been improvements in the management

Fig1.1

of restaurants. Each waiter is assigned a group of tables, after taking orders for a table the waiters enter the orders (a list of dishes and drinks ordered by the diner or group of diners) into the system at the PC. The waiter usually knows of any dishes that are unavailable before taking an order. The system must confirm the availability of dishes. Should an item not be available the system must allow the waiter to change or even delete a customer's order. Dishes to be prepared are sent to the kitchen, drinks orders to the bar. Starters and main course orders are usually taken together. Drinks and desert orders may be taken separately. Kitchen staff sees the dish orders on their screen, prepare them in an appropriate sequence and confirm preparation to the system when complete, similarly with the bar. When a waiter sees the completion indications on his terminal he collects the items and takes them to the table. The waiter can also check on the status of dish and drink orders. At the end of the meal the waiter will have the system print a bill, and he will enter the details of payment for it. The management can give discounts. The system keeps track of the numbers of customers served by each waiter and the amount of money taken by each waiter. The management can view these statistics [2]. The next advancement was "QORDER": the portable ordering system for Android devices. Here the waiter no longer approaches the table with his notepad, but rather with the QOrder hand held device. He enters order information on the touch screen and then sends it to the kitchen in real time for processing. Simultaneously, your POS system receives the sales information for later billing. QOrder utilizes WIFI to easily reach to your most remote corner spot in your establishment. Once the guests wish to leave, the waiter prints the receipt out on his belt printer and processes payment with the handheld unit much like he would on the POS system [3].

But there are still many areas which are not closely looked at. Like, making dynamic changes in the menu card, to get rid away from the heap of paper based records, to assure the customer that he'll be served with what he has ordered, to get the customer feedback on record.

Some of the existing system's are mentioned below:

- PixelPoint

PAR PixelPoint Company uses this software for managing the restaurant. The system consists of the company's software and hardware. This network system is compatible to TCP/IP, enabling information sending through both wireless and conventional networks [4].

- LRS Restaurant Server Pager Starter Kit

This system improves the food-ordering service quality in restaurants and reduces the waiting time of clients. The on-site paging system is used at UHF frequency or the frequency range of 467 MHz for sending the order data [5].

- Billpro Pocket® and Billpro POS for Restaurant

This system receives a client's order and makes a list by means of the designed client's template in the kitchen. The food ordering device is portable. The waiter takes the client's order and sends it to the client's template in the cook room[6].

- Implementation of Network-based Smart Order

System The Smart Order System in Restaurants (SOSIR) has been modified to take order from the client's table through RS-232 signal, which is sent to the cashier counter. The cashier counter system is connected to a database. When the clients' orders are sent the cashier counter system will screen and prioritize the orders before sending the information to the kitchen for the chef to cook [1].

**Table 2.1: Comparison of Existing Systems.**

	Pixel Point	LRS	Smart Order	DOSRUA
Wireless Network	YES	YES	NO	YES
Touch Screen	YES	NO	NO	YES
Digital Menu	NO	NO	NO	YES
Status of Order	NO	NO	NO	YES
Group Order	NO	NO	YES	YES

III. PROJECT SCOPE

In current formal dining environments, some form of physical static menu is utilized to convey the available food and beverage choices to customers. Said menus are generally paper based and hence impose restrictions on the textual real estate available and the ability a restaurateur has to update them. This document specifies the requirements for a restaurant paper menu and ordering replacement strategy to alleviate the problems associated with the current archaic method. Three related concepts are encompassed by the general scope of the Restaurant Menu and Ordering System. The first pertains to the replacement of paper-based menus using an electronic format, the second relates to a complementary electronic strategy for the front of house handling of a customer's order and the third surrounds the process of transferring said electronic orders to the kitchen for preparation. It should be noted that while the suggested strategy incorporates the use of various hardware components, the primary focus of the presented SRS relates to the constituent software elements.

The following are the features which can be a part of the proposed system:

Ordering, Waiting, Billing, Table Reservation, Home Delivery, KOT, Advertisement.

**3.1 User Classes and Characteristics**

The end-users of the DOSRUA fall into three primary categories, unskilled, partly skilled and highly skilled. Unskilled user: The users of the tablets at the table are walk-in customers and should therefore be assumed to have no relevant prior skills

or education other than basic abilities to operate an automated system; no more complex than a mobile phone. Partly skilled user: The users of the tablets and displays are managers and chefs respectively and they should be able to use the system and further be able to train others with minimal training themselves. They must be able to explain all elements of the user interfaces except the server. Supervisors also fall into the same category, though they will have to learn other sections of the system (refunds etc); these should not be of notably greater complexity than the standard functions. This class of user would be expected to have a high-school certificate education or equivalent. Highly skilled user: The initial installation and configuration of hardware and the constituent proposed system components (especially the server) is guaranteed to require someone with notable computer experience, including extensive experience with network and operating systems to complete it. The software should not be needlessly complex, but it is still expected not to be entirely 'plug and play'. This class of user is expected to have a graduate certificate or equivalent, as well as extensive computer experience.

### 3.1.1 Operating Environment

Android Operating system is an open source operating system. There are thousands and thousands of developers are there at sites trying to make android a better a operating system. There are so many eyeballs looking over the code every day. So the loopholes are quickly patched and fixed. Therefore android is secured. It always encourages your creativity. Unlike the iphone OS, Android user interface has been constantly refining and over the years. With Android 4.0 Google has made the user interface much more polished and modern. Apple charges people who want to develop applications for the App store \$100/year, while Google only charges Android developers \$25. So android prevails.

### 3.1.2 Design and Implementation Constraint

The proposed system should be written in an object oriented language with strong GUI links and a simple, accessible network API. Front end can be designed by using Rapid Application Development Tool (Indigo Eclipse). The system must provide a capacity for parallel operation and system design should not introduce scalability issues with regard to the number of tablets or displays connected at any one time. The end system should also allow for seamless recovery, without data loss, from individual device failure. It is worth noting that this system is likely to conform to what is available. With that in mind, the most adaptable and portable technologies should be used for the implementation. The system has criticality in so far as it is a live system. If the system is down, then customers must not notice, or notice that the system recovers quickly (seconds). The system must be reliable enough to run crash and glitch free more or less indefinitely, or facilitate error recovery strong enough such that glitches are never revealed to its end-users.

### 3.1.3 Assumptions and Dependencies

The implication is that the target hardware will provide a capacity for standalone program/application deployment and not require customized embedded firmware to be written. It is further assumed that tablet PCs of sufficient processing

capability and battery life will be utilized. The SRS assumes that none of the constituent system components will be implemented as embedded applications. The surface computers employed by the system should facilitate being utilized/left on for extended periods (sufficient for daily use) and that they are programmable in the same fashion as x86 architecture computers. Finally, it is further assumed that the deployment environment is capable of supporting an IEEE 802.11 wireless network for system communication. The maximum distance of transmission is within 50-100 meters, about the range of Wi-Fi.

## 3.2 System Features

Tablet on table:-

- There will be a tablet on each table.
- This will allow the customers to browse the food items for the time they wish.
- This will allow the customers to browse the food items the way the customer wish.

Customer feedback:-

- Customer can enter the feedback about the service and the food served.
- This helps the Restaurant owner to analyse the service and make necessary changes if needed.
- This also helps the Customer's to decide a particular food item with a positive feedback.

Searching Item:-

- Customer can search a particular food item according to name, price, category etc.
- This saves a lot of time of customer to order an item.

Offers for Customer:-

- The Restaurant owner can post various offers on tablet.
- This will help the customer as well as the restaurant owners.

Attractive Presentation:-

- The Menu is organized in an attractive way.
- There are images of every food item which will make the view of customers more clear about how the food will look like after delivery.
- here is an attractive use of Various themes and colour schemes.

Sorting an Item:-

- The food items will be sorted according to price, season and user ratings.
- This helps the customer to find or select a food item which has a good rating and which is liked by a many customers.
- This also helps the Restaurant owner to make changes in a particular food item if it has low ratings which improves the quality of food.

Time to Serve:-

- The menu includes the approximate time to be served of a particular food item.
- This will help the customer to select the food item accordingly.

Modifiable Menu:-

- The menu can be modified by the Kitchen manager.

- The items which are not available in a particular time period are not displayed on the menu card.

### 3.3 External interfaces Requirements

#### 3.3.1 User Interfaces

User Tablets:

This type of the tablets is especially for the use of normal users coming in the restaurant. These tablets will consist of the whole menu of the restaurant. They will be enabled with the Wi-Fi connectivity. The items in the menu are non-editable for these types of tablets. So, the user can not interfere in the menu and make changes in it. The tablets should be able to display all the items of the menu with sufficient visibility. Customer from any layer of the society should be able to handle and operate all the functions easily.

Manager Tablets:

These tablets are especially for the use of the restaurant manager. The manager should be able to control the function of whole restaurant from a single tablet. He can access any tablet and should be able to make changes to the menu. Like he can change price of particular item or he can disable particular item which is not available at that particular time.

Display at Kitchen:

These are present at the kitchen near chef so that he should be able to see what a particular has ordered. All the ordered items are displayed on the screen giving the table number below. They should be sufficiently large to be seen by chef at a reasonable distance. Chef should be able to denote a particular item that is ready.

#### 3.3.2 Hardware Interfaces

There are three external hardware devices used by the proposed system, each related to a user interface. These devices are the wireless tablets and the displays. All the devices must be physically robust and immune to liquid damage and stains. The devices (with the possible exception of displays) must also have good industrial design aesthetics, as they are to be used in place of normal restaurant tables and notepads and will be in direct contact with customers. The devices behave as 'terminals' in the sense that they never have a full system image, do not store data and are not used for the core logic of the system. However, they should be fully capable tablets that can use textual data from the server along with local UI/interpretation code to display UI elements and take input. All order and transaction records should be stored on the server, not these tablets. The performance of dumb terminals over an area the size of a restaurant is likely to be unacceptable. In all the cases, the hardware device takes information from the proposed system and processes the information to display. It also provides user input information to the proposed system.

#### 3.3.3 Software Interfaces

We will require interface with a JSP/Servlet that stores the information necessary for our system to operate. The JSP/Servlet must be able to provide, on request and with low latency, data concerning the restaurant's menu, employees (and their passwords) and available dietary requirements. Additionally, it should take and archive data provided to it. This data will include records of all orders and transactions (system states and state changes) executed. JSP/Servlet must store all data such that it can be used for accounting, as well as accountability.

#### 3.3.4 Communication Interfaces

The DOSRUA will interact with a WiFi to maintain communication with all its devices.

#### 3.4 Non-Functional Requirements:

This subsection presents the identified non-functional requirements for the subject of proposed system. The subcategories of non-functional requirements given are performance safety, security and software quality attributes.

**Table 3.1 Non-Functional Requirements**

Description
A manger password used for tablet login must have a bit strength of at least 64bits.
A manager password used for tablet login must be changed every three months.
A manager shall only be able to log into one tablet at any given instance of time.
The display shall not require an user to log-in.

#### 3.4.1 Performance

Performance requirements define acceptable response times for system functionality.

- The load time for user interface screens shall take no longer than two seconds.
- The log in information shall be verified within five seconds.
- Queries shall return results within five seconds.

#### 3.4.2 Safety

Table presents the identified non-functional safety requirements that directly relate to the entire subject proposed system.

#### 3.4.3 Security

Table 3.4.3 presents the identified non-functional security requirements that directly relate to the entire subject proposed system.

**Table 3.2 Security**

The system shall log every state and state change of every user tablet and display to provision recovery from system failure.
.
The system shall be capable of restoring itself to its previous

state in the event of failure (e.g. a system crash or power loss)
The system shall be able to display a menu at all timesto facilitate Manual order taking should the need arise.
The system shall utilise periodic 30-second keep alive messages between tablets and the server to monitor tablet operational status.
The system shall flag tablets that fail to send timely keep alive messages as non-operational and disassociate the assigned waiter from the tablet

- 4] A complete database is stored in a single cross-platform disk file.
- 5] Supports terabyte-sized databases and gigabyte-sized strings and blobs.
- 6] Small code footprint: less than 350KiB fully configured or less than 300KiB with optional features omitted.
- 7] Faster than popular client/server database engines for most common operations.
- 8] Simple, easy to use API.
- 9] Written in ANSI-C. Bindings for dozens of other languages available separately.
- 10] Well-commented source code with 100% branch test coverage.
- 11] Available as a single ANSI-C source-code file that you can easily drop into another project.
- 12] Self-contained: no external dependencies.
- 13] Cross-platform: Unix (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT) are supported out of the box.
- 14] Easy to port to other systems.
- 15] Sources are in the public domain. Use for any purpose.

#### Software Quality Attributes:

- 1) Coding in Android is very beneficial from the developer's point of view. There is availability of large number of documentation. Also, it could be easily run on tablet as well as any Android phone.
- 2) Using Android is very flexible as the developed product could be deployed on tablets as well as android driven mobile phones which as available in abundance.
- 3) Though maintenance is required, it is negligible.
- 4) The devices on which Android run are highly portable.

### 3.5 Other Requirements

#### 3.5.1 DB requirements:-

The database required for this system is SQLite database for storing details on the tablet itself. It also needs a database on the server which is handled by JSP and SQL.

So what basically is SQLite?

SQLite is a relational database management system contained in a small (~350 KiB) C programming library. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it. SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. OS like Android, Web browsers like Mozilla etc. SQLite has many bindings to programming languages.

#### Features Of SQLite

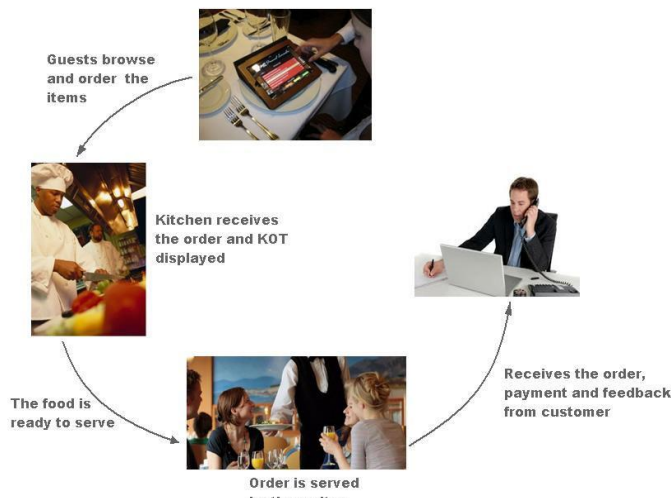
- 1] Transactions are atomic, consistent, isolated, and durable (ACID) even after system crashes and power failures.
- 2] Zero-configuration - no setup or administration needed.
- 3] Implements most of SQL

#### Suggested Uses For SQLite:

Application File Format: Rather than using fopen() to write XML or some proprietary format into disk files used by your application, use an SQLite database instead. You'll avoid having to write and troubleshoot a parser, your data will be more easily accessible and cross-platform, and your updates will be transactional. Database For Gadgets: SQLite is popular choice for the database engine in cellphones, PDAs, MP3 players, settop boxes, and other electronic gadgets. SQLite has a small code footprint, makes efficient use of memory, disk space, and disk bandwidth, is highly reliable, and requires no maintenance from a Database Administrator. Website Database: Because it requires no configuration and stores information in ordinary disk files, SQLite is a popular choice as the database to back small to medium-sized websites. Stand-in For An Enterprise RDBMS: SQLite is often used as a surrogate for an enterprise RDBMS for demonstration purposes or for testing. SQLite is fast and requires no setup, which takes a lot of the hassle out of testing and which makes demos perky and easy to launch

## IV. SYSTEM ARCHITECTURE

Fig. 4.1: System flow



System architecture of project could be described as:

When the customer enters the restaurant, he would surf on the tablet to order his menu. He could also surf quickly if he has already decided upon what to order. He would click the item he wants to order and after he is sure he wants each item in the list, he would click confirm. The confirmed order would be displayed on the display screen in the kitchen. After the chef has completed preparing the item, it would be notified to the customer. After the customer has completed eating the food, bill would be directly displayed on his tablet as well as managers system.

## V. TECHNICAL SPECIFICATION

The technologies which are used to implement the system are:

- 1) Android version 2.2.3 (Smart Phone) and Android version 2.2 – 4.0 for Tablets is required.
- 2) Java SE 6 Programming Language is used to develop the software.
- 3) Eclipse Indigo is used as a Rapid Application Development Tool (RAD) or as an Integrated Development Environment (IDE) for coding the software.
- 4) JSP/SERVLET is used for Remote Database Access from the main system of the restaurant.
- 5) SQLite is a light weight Database which is going to be used for database access from handheld device or the tablet.

### 5.1 Advantage

- Wastage of paper is avoided as our implementation is working just on tablet and does not need any paper work. e.g.-For taking the order, we are not using papers. Also, our menu card would be digitized.
- A customer going into restaurant does not has to wait for the waiters to take the order. As soon as he occupies a seat, he would order whatever he needs.
- As soon as the order is ready, it would be notified to the customer. So, there would not be any issue of late delivery in spite of the food being ready.

### 5.2 Disadvantage

- Tablets would cost us more as they are more costly the simple paper. Hence, it would cost us more. Though it

**Correspondence Author** – Ashutosh Bhargave,

would be a onetime investment, it would certainly be more costly.

- If we compare our system with traditional paper based system, more maintenance would be needed. Some technical assistance would also be needed.

## 5.3 Application

We are going to implement our system in restaurants to ease the management of the Restaurant and also give a technical touch which would help atomize the working of Restaurant.

## VI. CONCLUSION

The proposed system would attract customers and also adds to the efficiency of maintaining the restaurant's ordering and billing sections.

## ACKNOWLEDGMENT

Special Thanks to our Head of Department Prof N.V Alone, for his valuable suggestions and encouragement.

## REFERENCES

- [1] M.H.A. Wahab, H.A. Kadir, N. Ahmad, A.A. Mutalib and M.F.M. Mohsin, "Implementation of network-based smart order system," International symposium on Information Technology 2010.
- [2] Cormac O'Connell, Restaurant Assignment.
- [3]"QOrder" The portable ordering system for Android devices.
- [4]PAR PixelPoint "PixelPoint POS Brochure", Available <http://www.pixelpointpos.com>
- [5]Advanced Analytical, Inc (October 2004) "LRS Restaurant Server Pager", Available [http://foodsoftware.com/Product\\_0132.asp](http://foodsoftware.com/Product_0132.asp)
- [6]Billpro Pocket® and Billpro POS for Restaurant Available: [http:// billpro-pos.apponic.com/](http://billpro-pos.apponic.com/)
- [7] GHIRS: Integration of Hotel Management Systems by Web Services
- [8]Wei-Meng Lee , Beginning Android Application development by "wrox"
- [9] Mark Murphy , Beginning Android 3,by "Apress".

## AUTHORS

**First Author** – Ashutosh Bhargave, Department of Computer Engineering, GES's RHSCOE, [ashubhargave@gmail.com](mailto:ashubhargave@gmail.com)

**Second Author** – Niranjan Jadhav, Department of Computer Engineering, GES's RHSCOE, [niranjanjadhav25@gmail.com](mailto:niranjanjadhav25@gmail.com)

**Third Author** – Apurva Joshi, Department of Computer Engineering, GES's RHSCOE, [apurva.joshi91@gmail.com](mailto:apurva.joshi91@gmail.com).

**Fourth Author** – Prachi Oke, Department of Computer Engineering, GES's RHSCOE, [prach.oak@gmail.com](mailto:prach.oak@gmail.com)

**Fifth Author**- Prof. Mr. S.R Lahane, Department of Computer Engineering, GES's RHSCOE, [shivaji\\_lahane@rediffmail.com](mailto:shivaji_lahane@rediffmail.com)

