International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471
Vol. 4 Issue 1, January-2015, pp: (23-31), Impact Factor: 1.296, Available online at: www.erpublications.com

Implementation of Paperless Dining Experience using Client-Server Model with Cloud and QR Technology

Aditya Sundaresan[1], Salil D. Pol[2], Abhijeet Bhamare[3], Meenal Jadhav[4], Gaurav Dhokre[5], Tanuja Dhope[6]
[1,2,3,4,5,6]G. H. Raisoni College of Engineering and Management, Domkhel Road, Wagholi, Pune, India

Abstract: In this paper, the design and implementation of a paperless digital dining experience in restaurants using Client-Server model with Cloud and QR technologies has been proposed. This system has a dynamic database utility system which fetches all information from a centralized cloud based database. The tablet at the customer table contains the android application with all the restaurant and menu details through which the customer can place an order and the application at the admin end enables the admin to manage the services and update menu dynamically. The food ordering systems that have been proposed earlier exhibit limitations, primarily in cost effectiveness, allowing customizations and supporting real-time feedback to customers that we wish to overcome in this paper.

General Terms: Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords: Android tablets, paperless food ordering system, Distributed Computing, Cloud Computing, Google App Engine, App Scale, Recommendation algorithms.

1. INTRODUCTION

The restaurant services such as making reservations, processing orders, and delivering meals generally requires waiters to input customer information and then transmit orders to the kitchen for menu preparation. When the customer pays the bill, the amount due is calculated by the cashier. Although this procedure is simple, it may significantly increase the waiters' workload and even cause errors in menu ordering or in prioritizing customers, especially when the number of customers suddenly increases during busy hours, which can serious degrade overall service quality. With the increase in the number of emerging wireless & cloud technologies as well as the no. of restaurants and population of restaurant-goers, there is a need to enhance the working of hospitality industry. This project aims for the betterment of hospitality industry by incorporating technology that could help to improve the workflow of order management in the hotel industry. A recent survey on the utilization of technology in hospitality industries showcased that various applications based on wireless & other technologies are already in use, enabling partial automation of the food ordering process. While the age old paper based systems are cheaper to deploy and have been used widely across the globe, yet the overgrowing competition and need for perfection and swift delivery has cried out for the need of efficient ways to ensure better efficient ordering systems. Many systems have been tested and put to use for this purpose, like use of PDA(Personal Digital Assistance) and the POS(Point Of Sale) system. But these systems have disadvantages of their own which we have tried to study and propose a modified& efficient way of paperless dining experience.

The system proposed here takes away the waiters' responsibility of taking the order and delivering it to the kitchen. This ensures the waiters to focus completely on the job of delivering the food to the customer. Also the customers can get a detailed idea of the food that they are going to order. At times it happens so that the menu item is unrecognizable for the customers and they have to rely on the waiters' descriptions to get a general idea of what it might be. If they order it based on their understanding, it may happen that the customers end up dissatisfied. This calls for a return of food order which turns out to be a lengthy process and fuels dissatisfaction even further. The proposed system aims at eliminating such an inconvenience. A high quality service system should be customer centred, i.e., it should immediately recognize customer identities, favorite menus, and expenditure records to provide customer-centric services. Google App Engine(GAE) which is a service for hosting web applications can be used for free of charge execution of computer intensive applications. It's often presented as an example of a Platform as a Service (PaaS) cloud computing solution because web application developers can use it to create their applications using a set of Google tools. The applications are executed in a secure hosting environment running on a computing infrastructure in a sandbox like state that Google provides and manages. The infrastructure also offers self-load balancing and auto-

scaling capabilities i.e., it automatically adjusts the number of applications instances (virtual machines) to the request rate[1]. However, the quote used to provide optimum service to clients, was unsuitable for our proposed model. App Scale is an open source distributed software system that implements a Cloud PaaS. App scale makes cloud applications, easy to deploy and scale over disparate cloud fabrics, implementing a set of APIs and architecture that also makes apps portable across the services they employ. App Scale is API compatible with GAE and thus executes GAE applications on premise or over other cloud infrastructures, without modification [2]. We have used android operating system to design the client side application [3] with application of user interface design techniques for mobile applications [4]. In the following sections GAE and App Scale Technologies are explained further more.

2. LITERATURE SURVEY

A In[1], the system provides online menu-ordering and reservation-making functions along with a personal menu recommendation service. The customers are immediately identified via RFID(Radio Frequency Identification)-based membership cards and then actively recommend the most appropriate menus for customers according to their consumption records. For new customers, the recommendations are provided based on meal popularity and then customers' preferences are created and stored in the back-end database. For long-time customers, Multiple-Criteria Decision-Making (MCDM) approach is used by the service staff, as detailed in Section III, to infer items preferred by customers or items close to those preferred items based on customers' preference data stored in the system. This system, therefore, helps service providers increase their customer interactions and provides fast and thoughtful services. In this system, the waiter uses a PDA to take customer orders and then sends the order wirelessly to the kitchen server. The menu is then displayed on the display system built in the kitchen, according to which the chefs prepare the menu. The restaurant manager can also use the system to view statistics of the current inventory, sales records, staff information, and other information. Once the customer has finished the meal, the cashier uses the RFID-based PDA to identify the membership ID to check out the bill.

In the system which is proposed in the paper[5] and paper[6], the user interface of the system is built with Visual C# 2005 and embedded Visual C++. The database is built on a Microsoft SQL Server 2005 for server management and statistic reporting. This system shows certain limitations which we intend to lay our focus on and make an attempt to come over in the proposed system. Some of them being, each customer must visit at least five times to enjoy the benefits of the customer-centric service that the menu recommender provides and the customers who visit less than six times are recommended the most frequent ordered menus that are similar to those of the new customers. In this system, the customers do not receive the menu recommendations if they have privacy concerns. Also, here, when new menus or new criteria are introduced, the recommender system needs to collect enough data to retrain and rebuild the system and the model must be modified if new criteria are introduced, which in turn will affect relationships, dominance values, strength of preference, and customer's dominance degree.

Irrespective of the drawbacks of the system, certain lessons can be learned from it. For instance, the service procedure must be identified with care and precision step by step before developing the e-restaurant. Moreover, user interface of the proposed recommending system must be designed based on the viewpoints of waiters to ensure ease of use and simplicity in design. Easy up gradation of the traditional restaurant by following the framework of the intelligent e-restaurant with inexpensive equipment must be ensured. In a nutshell, this study constructed an intelligent e-restaurant system using RFID, WLAN, database technologies, and a menu recommender to offer customer-centric service to enhance customer service quality and improve restaurant industry competitiveness [5].

In [7] the system architecture which covers three main areas of the restaurant: the serving area, the restaurant owner's working desk (cashier table), and the kitchen. Conceptually, it is built on four main components, namely, the mobile application on the smart phones for customers to make order, the web-based application and server on the laptop for restaurant owner to keep track and respond to received customers' orders and customize menu information, the database for restaurant owner to store order details, and updated menu information and The wireless infrastructure to support networked communications. This architecture promotes portability as the set up can easily accommodate all kinds of restaurants including the ones without the internet access.

The CWOS-RTF(Customizable Wireless Ordering System With Real Time Feedback) has been developed using both programming languages, ASP.Net and VB.Net. The database of the system has been designed using Microsoft Access 2008. The above described paper presents a customizable wireless food ordering system with real-time feedback to customers. Instead of using PDAs to interface with customers, smart phones are used to provide necessary interfaces for customer to view and order menu. With private login system, customers can view and make order and receive updates in real-time and collect receipts right from the smart phone itself. It allows restaurant owners to manage orders from customers instantaneously whenever he or she logged in into the system. This system shows the capabilities of wireless communication and smart phone technology in fulfilling and improving business management and service delivery [7].

The following section shows the existing systems and their comparative analysis.

2.1 Comparative study

The manual or paper based systems have been persistent in the restaurant business since ages. Here the customer chooses the meal from a menu card provided to him or her at the table. The waiter notes down the order and issues a KOT (Kitchen Ordering Ticket) to the staff in the kitchen. The chef makes the order as per the KOT and the waiter dispatches it to the desired table [8]. With the advancement in technologies, the PDAs replaced the papers where the waiters usually noted the orders down. Using PDAs, the waiters placed the order which as relayed to the kitchen. PDAs however need prior awareness and some training [9].

POS , during computerization, later becoming electronic point of sale or EPOS is the place where a retail transaction is completed. It is the point at which a customer makes a payment to the merchant in exchange for goods or services. At the point of sale the retailer would calculate the amount owed by the customer and provide options for the customer to make payment. The merchant will also normally issue a receipt for the transaction. The POS in various retail industries uses customized hardware and software as per their requirements. Retailers may utilize weighing scales, scanners, electronic and manual cash registers, EPOS terminals, touch screens and any other wide variety of hardware and software available for use with POS. For example, a grocery or candy store uses a scale at the point of sale, while bars and restaurants use software to customize the item or service sold when a customer has a special meal or drink request[10].

Table 1. Comparative Study

Method	Advantages	Disadvantages	Requirements
Manual/Paper based[10]	Cheap	Easy to tamper & Possible wastage of Resources/Man-power	1.Pen 2.Paper 3.Carbon paper for copy
PDA (Personal Data Assistant)[11]	1.Improves order response 2.Easy to use	1.Limited System 2.Restricted to Staff only	1. PDA 2.Supported H/W
POS (Point-Of-Sale)[12]	1.Secure control over business flow 2.Capacity to optimize response time 3.Some system also offer business logic to improve business practice	1.Complicated to use 2.Might require training for staff 3.Modular design means system is brought in modules which may increase cost	1. PC 2.PDA/Tablet 3.Supporting S/W & H/W

3. PROPOSED SYSTEM

The proposed system takes away the waiter's responsibility of taking the order and delivering it to the kitchen. The customer orders using the tablet provided to them at their table. The customer gets menu recommendations based on his choices and previous orders. The admin verifies the order using the admin side system which is displayed on the kitchen display unit. The ready order is dispatched by the waiter and admin does the work of taking it away from the kitchen display. will take care of the business logic, book keeping and the menu editor. The GAE is deployed using the App Scale which allows offline deployment if needed.

Figure 1: Proposed system model

The Client Tablet and the Admin PC communicate with the API end points of the Google App Engine which holds the app logic and the master Database. The GAE is hosted on the App Scale cloud which contains the App Server, the Load Balancer and the Slave Database. The Master Database is responsible for handling the queries and data operations whereas the slave database holds the data.

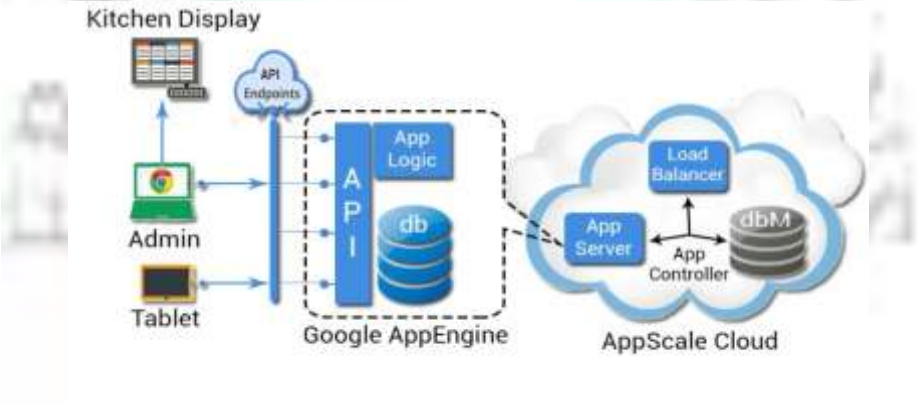

Figure 2: Project Implementation

GAE is a service for web hosting applications. The applications are executed in a secure hosting environment running on a computing infrastructure that Google provides and manages. The infrastructure offers load balancing and auto-scaling capabilities i.e., it automatically adjusts the number of applications instances (virtual machines) to the request rate. Web application developers can use it to create their applications using a set of Google tools [1].

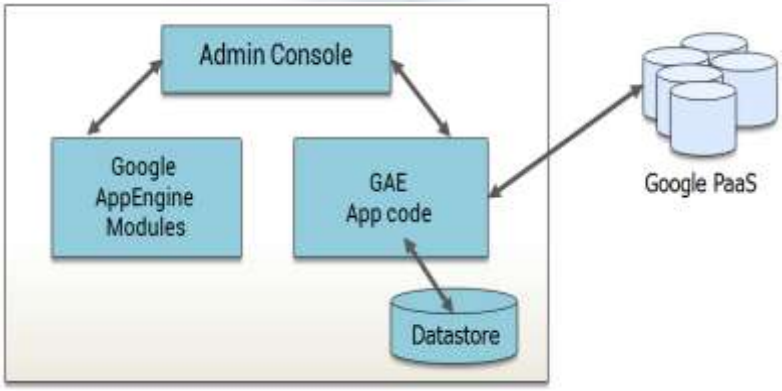

Figure 3: Google App Engine

The GAE consists of the admin console, GAE Modules, the GAE App code and the master Database. In this system, as stated earlier, the GAE is deployed on the App Scale. The features of App Scale which makes it usable in our project are as follows [2]: It has a Production-ready platform with full GAE compatibility. App Scale provides unlocked deployment options. It provides customer choice of services for API plug-ins. It provides portable and hybrid app deployment across clouds and services. It is open, free, customizable execution environment for web apps. It provides automatic configuration, deployment, & scaling and fault tolerance of apps and their service.

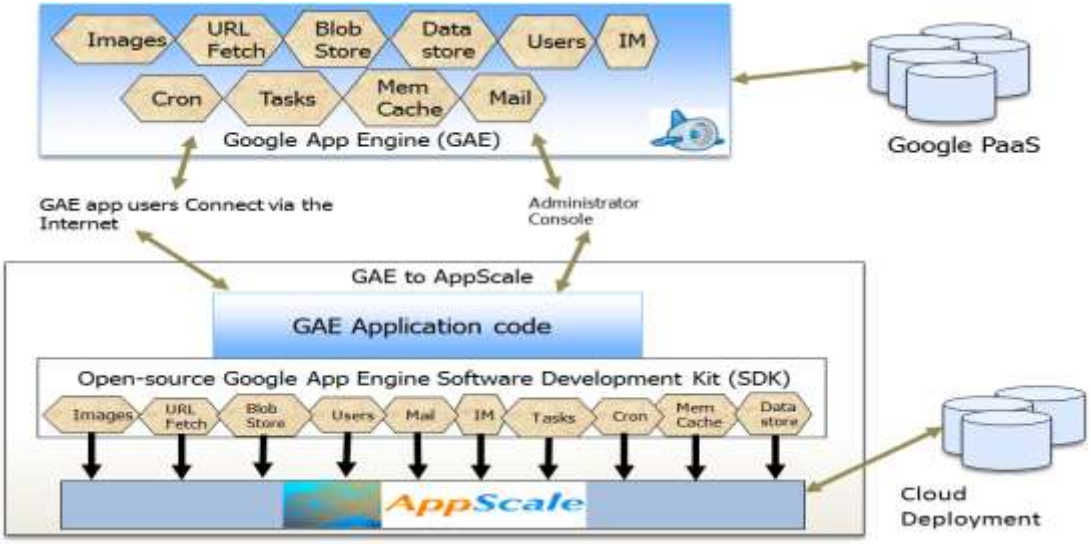

Figure 4: App Scale Deployment

The GAE is deployed using the App Scale as shown in the figure above. The App Scale uses its own corresponding modules for various functions using the GAE Application code and the GAE is connected via the internet.

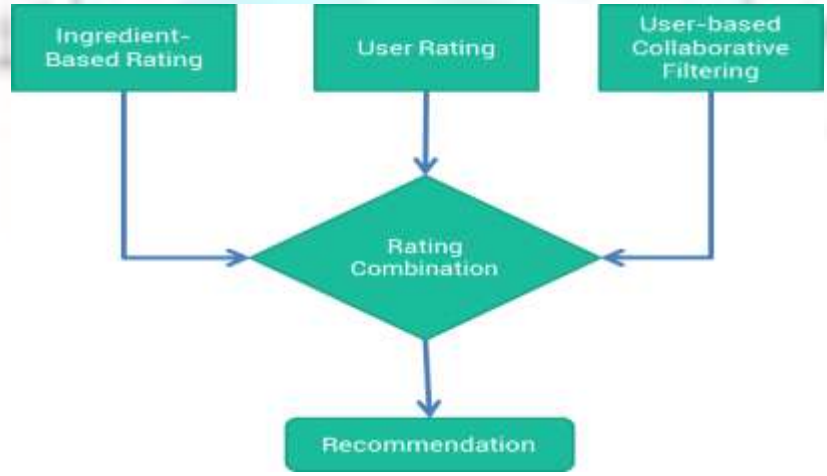

Figure 5: Recommendation system

The recommendation system combines three recommendation paradigms: If the customer is a first timer at the restaurant, the user rates the ingredients and based on that the user is recommended the menu items which involve the use of the ingredients favoured by the user. If the customer is a frequent visitor at the restaurant, the customer gets recommendations as per his/her past ratings. This helps them to make proper judgement about the food they wish to order. If the other customer shows the same liking pattern as the customer, the menu liked by that other customer is also suggested to the customer. It gives a wider perspective to the customer on what to order and the dishes that he/she may like. Recommendation systems are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item. Recommender systems have become extremely common in recent years, and are applied in a variety of applications. The most popular ones are probably movies, music, news, books, research articles, search queries, social tags, and products in general. However, there are also recommender systems for experts, jokes, restaurants, financial services, life insurance, persons (online dating), and Twitter followers.

Each type of system has its own strengths and weaknesses. Recommender systems are a useful alternative to search algorithms since they help users discover items they might not have found by themselves. Interestingly enough, recommender systems are often implemented using search engines indexing non-traditional data. Recommender system is an active research area in the data mining and machine learning areas.

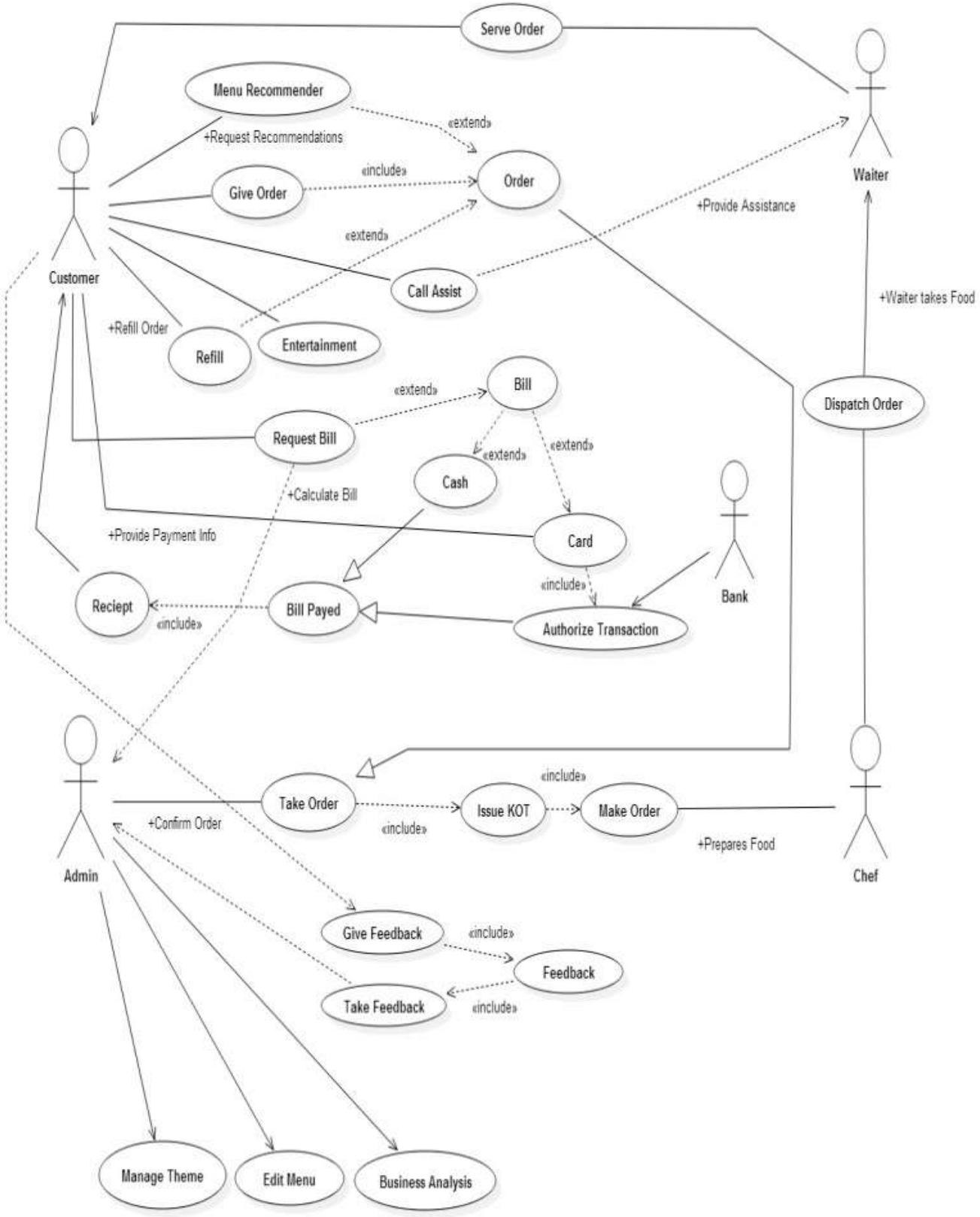

Figure 6: Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the case.

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role (also known as "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time.

The actors in the system are customer, admin, waiter and chef. The customer interacts with the menu recommender, give order, refill, entertainment, bill pay and Give feedback use cases. The waiter interacts with the serve order use case. The admin interacts with the take order, bill accept, take feedback, manage theme, business analysis, edit menu and issue KOT use cases. The chef is involved with the make order use case.

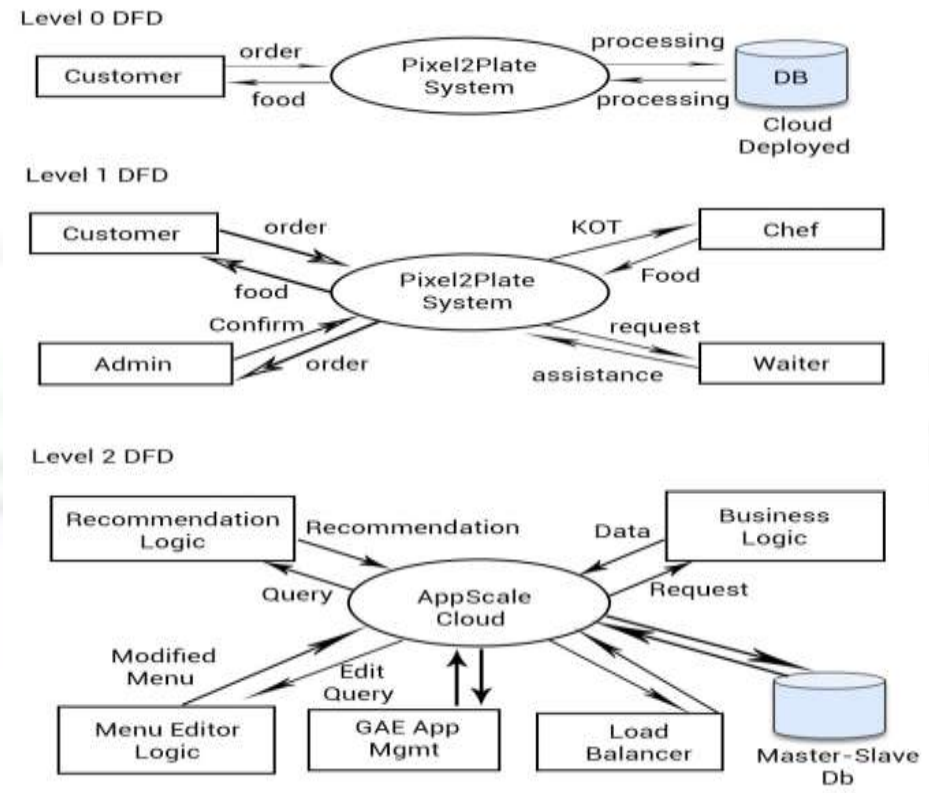

Figure 7: Data Flow Diagram

In level 0 of the Data Flow Diagram (DFD) the Pixel2Plate (P2P) system takes order from the customer and gives it for processing to the database. The database gives result to the P2P system and the system P2P dispatch food to the customer through the waiter.

In level 1 of the DFD, the interaction of the P2P system has been shown with the four actors namely customer, admin, chef and the waiter. The customer issues order through the P2P interface provided on the tablet. The admin receives the order from the system and he confirms the order using the admin interface. The system issues a KOT to the kitchen display for the chef to act on. The Chef prepares the food according to the order which is dispatched to the customer by the waiter. The waiter also accepts request and provides assistance.

In level 2 DFD the lowest level of abstraction is shown. The App Scale cloud acts as the central processing unit of the system. The recommendation logic takes in query and gives out the recommendation based on the query. Business logic takes in the requests and gives out processed data as per the request. The menu editor corresponds to dynamic menu editing options and takes in edit query and gives out the modified menu accordingly. The GAE app management involves with the management functionalities of the GAE. When the number of queries increases, the load balancer balances the load by increasing the number of servers. The master and the slave databases involve storage and retrevial of data and work on SQL queries.

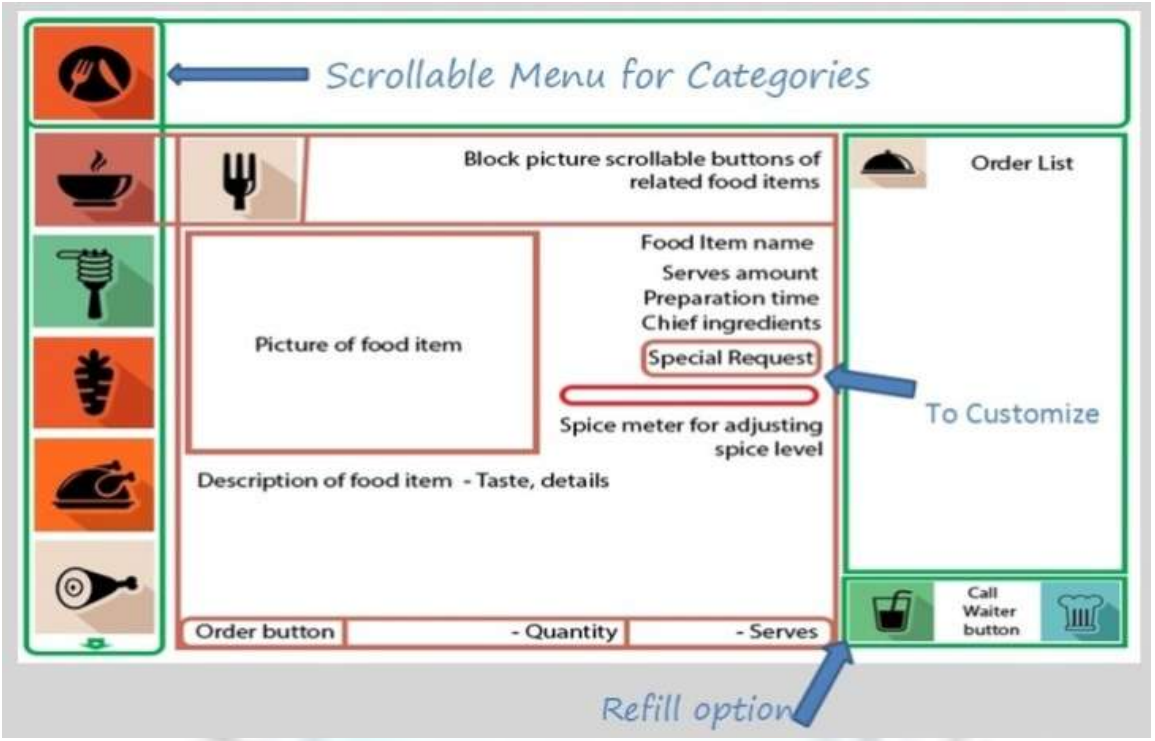

Figure 8: Tablet UI Prototype

The client UI has been designed such that it follows a simplistic design which will enable even the naive users to efficiently use the system with ease. The UI has special features like the spice meter and call for assistance. The spice meter enables the user to adjust the spice content in the food they wish to order. The call for assistance calls the waiter to the table on which the user requires assistance.

Figure 9: Admin UI Prototype

The admin UI is simplistic in nature as well so that even a non technical admin can handle it efficiently. The admin UI follows the windows 8 style of tile design for as thetic layout.

4. Conclusion

This study integrated Android Tablet Application, Cloud Deployment, MySQL Database, Google App Engine, and App Scale technologies to implement an e-restaurant to support customer-centric service. This system can actively recommend menus and enable dynamic menu updation to provide customer-centered services. Experimental result reveals that the proposed system has the potential for practical application and can be promoted in restaurants.

There is a huge potential scope for the system to be expanded for improved efficiency and ease of deployment and use in terms of the aspects like involving an inventory management system where the admin can manage and get details about the items available and needed which will ease the administration of the restaurant even further, The Adaptive Database Schema Design for Multi-Tenant Database Management, Improving User Experience based on feedback, Future improvements in Recommendation system & Business Analytics, Provide features in Theme manager for better experiences for customer, Custom ROM using AOSP for enhanced security and reliability, etc.

5. Acknowledgments

This work is supported by Geogreenapps IT Solutions, Pune.

6. References

[1]. Maciej Malawski, Maciej Kuz´niar, Piotr Wójcik and Marian Bubak. © 2013 IEEE. How to Use Google App Engine for Free Computing. Published by the IEEE Computer Society 1089-7801/13/$31.00.
[2]. Chandra Krintz. © 2013 IEEE. The AppScale Cloud Platform. Published by the IEEE Computer Society, 1089-7801/13/$31.00.
[3]. Margaret Butler. © 2011 IEEE. Android: Changing the Mobile Landscape. Published by the IEEE CS n 1536-1268/11/$26.00
[4]. P. H. J. Chong, P. L. So, P. Shum, X. J. Li, and D. Goyal. NOVEMBER 2004. Design and Implementation of User Interface for Mobile Devices. At IEEE Transactions on Consumer Electronics, Vol. 50, No. 4.
[5]. Ching-su Chang, Che-chen Kung, Tan-hsu Tan. Development And Implementation Of An E-restaurant For Customer-centric Service Using WLAN And RFID. At Proceedings Of The Seventh International Conference On Machine Learning And Cybernetics, Kunming – IEEE.
[6]. Tan-hsu Tan, Ching-su Chang, Yung-fu Chen. Developing An Intelligent E-restaurant With A Menu Recommender For Customer-centric Service. At IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 42, No. 5.
[7]. Noor AzahSamsudin, Shamsul Kamal Ahmad Khalid,mohdFikryAkmalMohdKohar, ZulkifliSenin, Mohd Nor Ihkasan. A Customizable Wireless Food Ordering System With Real-time Customer Feedback. At IEEE Symposium On Wireless Technology And Applications (ISWTA).
[8]. IMH Mumbai FYB http://ihmmumd-m09-12.blogspot.in/2010/02/kitchen-order-ticket-kot-kot-is-written.html
[9]. Mishul Prasad, Scornavacca, E.. Lehmann, H. Using wireless personal digital assistants in a restaurant: impact and perceived benefits. Sch. of Inf. Manage., Victoria Univ. of Wellington, New Zealand.
[10]. William M. Brobeck, John S. Givins, Jr., Philip F. Meads, Jr., Robert E. Thomas; United States Patent 3,946,220. uspto.gov.
[11]. Sang Hyun Choi, Young-SeonJeong, and Myong K. Jeong. SEPTEMBER 2010. A Hybrid Recommendation Method with Reduced Data for Large-Scale Application. At IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, VOL. 40, NO. 5.
[12]. Jiacai Ni, Guoliang Li, Lijun Wang, JianhuaFeng, Jun Zhang, and Lei Li. SEPTEMBER 2014. Adaptive Database Schema Design for Multi-Tenant Data Management. At IEEE transactions on knowledge and data engineering, VOL. 26, NO. 9.