RESTAURANT AUTOMATION SYSTEM

TEAM NAME : **CometGrub**

DATE : September 29, 2019

Avani Sah

Harshita Rastogi

Kritika Manchanda

Rahul Shivakumar

Siddharth Chandrasekar

SUMMARY

Restaurant management automation is an application to restaurant management system. It handles the flow and storage of orders placed by a customer.

On a busy day in a restaurant, we might see people waiting more than usual, waiters forgetting the orders and even serving wrong orders. When restaurant wants to change the order, printing a menu every time does not look like a feasible option. The traditional hand waving method/paper pen method is confusing and often leads to complaints and customer dissatisfaction. To overcome these obstacles, an automated solution is needed which reduces manpower while keeping the productivity and customer satisfaction at high.

Firstly, the project begins with requirement analysis keeping all the stakeholders in the loop. Secondly, iterations are introduced which constitute tasks delegated to team members. These iterations can be treated as sprints. The features are designed, developed and tested through each iteration. The process is designed using diagrammatic representations for better insights. The implementation is discussed using the diagrams and pseudocode for modules. The modules are developed and integrated followed by each module being tested separately as well as in integration with others. Thirdly, if time permits, secondary features which would be defined in future scope during requirement analysis are developed and tested.

The application consists of 3 primary actors: chef, manager and customer. Each of them have certain responsibilities and permissions. The use cases include preparing orders, taking orders, updating menus, ordering food and making payments.

The project is implemented using Java as front end and MySQL as backend tool. MySQL. MySQL is a freely available open source relational DBMS that uses structured query language for adding, retrieving and managing data in the database. We have used JDBC connectivity. JDBC helps connect our java front end to our database backend. The project will be built on the Eclipse IDE. The process will result in a high performing system which is easily scalable and can run numerous parallel operations efficiently.

PROJECT RELEVANCE

- **Relevance or importance of problem**

There are a few reasons why 'Restaurant Automation System' is important. Stating below:

- ★ **Workforce Management**: Workforce management in the conventional manner involves filling up spreadsheets, paper schedules, printed contracts, paper-based timesheets, etc. All these are quite time-consuming and extremely troublesome. Also, hard to manage if it is a large workforce. But an automated restaurant management solution tends to be 100 times more efficient as it automates the entire spectrum of employee scheduling and customer handling.
- ★ **Inventory Control:** This is a major expense for a restaurant. And, managing an inventory on paper or just with man-power can be tiresome and inefficient. Also, is bound to be full of human errors. However, an automated inventory control can help with this issue.
- ★ **Tracking Sales and billing:** A large-scale restaurant handles high volumes of cash and credit cards daily. Restaurant automation system, unlike the traditional cash register can track all sales down to the last penny with minimum human errors. With this kind of insight, they can adjust the menu accordingly.
- ★ **Meal Ordering:** Unlike the conventional methods, customers can order using automated machines without long queues at the cash counter. Also, as soon as an order is paid, it can directly be added to the manager and chef's portal.

- **Background information to educate the reader**

A restaurant generally comprises of four departments that work co-dependently.

First, **workforce management department** would handle the employees and their work-related and official information, their taxes, their time-sheets, etc.

Second, **inventory control**; this department handles all the inventory used in a restaurant, this includes ordering, storing and managing inventories.

Third, **financial department**; this department would handle customer billings, inventory purchases and payroll of employees.

Fourth, **meal ordering and preparing department**; this department handles the ordering, meal preparation and delivering the meals. This department includes chef and his workers, and customer service that takes orders face-to-face or on-call (in some cases).

All these departments work together to manage a restaurant efficiently. An automated restaurant management system would help managing all these departments in a single desktop/web application with minimal human errors and effort.

- **Previous related work by others**
  - **Jolt:** It is a tablet-based software used to manage daily operations for brands like Chick-fil-A, McDonald's, and Buffalo Wild Wings. Owners and managers get a real-time look at daily operations from their phone, and in-store tablets keep staff accountable and on task. Jolt includes a training library, drag-n-drop schedule builder, text notifications, a time clock, and more. Trusted in thousands of restaurants across the globe, Jolt has helped over 300,000 users complete over 700 million tasks.
  - **Restaurant365:** Restaurant365 is a cloud-based restaurant platform designed for multi-unit restaurant locations, combining key restaurant operational functions. Restaurant365 provides greater control over food and labor costs, simplifies managing multi-units, and automates the accounting and reporting processes. With powerful integration to POS, banking, payroll and vendors R365 provides robust reporting and analytics. In addition, manage inventory, scheduling, menu engineering, recipe costing.

- **SynergySuite for Restaurants:** SynergySuite gives multi-unit restaurants the insight and tools they need to manage the back office. Our cloud-based, mobile-first platform simplifies back-of-house management for brands like Tropical Smoothie Cafe, Beef O'Bradys and Costa Vida. SynergySuite customers save an average of 4-6% on food and labor costs, using SynergySuites streamlined modules for inventory, purchasing, recipe costing, food safety, scheduling, cash management, human resources and business intelligence.

❖ **Detailed Problem Description**

One of the main issues with managing a restaurant is communication. For example, a chef in the kitchen may be unaware of the sequence in which the orders are made, and this may lead to customers receiving their orders late. For a small-scale restaurant operating with minimal staff this may not be a very large issue. However, when the number of customers is large, and the efficiency of the kitchen operations is low, it becomes more relevant. Another issue faced by restaurants is the management of paper records for every transaction. Proper maintenance and documentation of receipts, orders, ingredients, bills, etc. is a time consuming and cumbersome process.

The problem that this system aims to solve is to allow the actors in the restaurant environment to interact with one another and maintain records of their interactions in a simple time-efficient way. Communication between the different verticals within the organization must be dealt with, and all transactions must be recorded.

SCOPE

The scope of the project is very broad in terms of the manual work it replaces at multiple levels. A few of them are as follows:

- The current scenario of booking a table by calling involves a human intermediating, but the application takes care of it for the person, thereby avoiding any schedule clash or any human errors.

- Currently, since the order placing or even the table is via call, the user does not have a clear idea of what to choose and there is no certainty either. With a properly built system, we can avoid any kind of communication gap and show the user all the necessary details without any ambiguity.
- The whole system, starting from the user placing an order which is being served by the chef in the restaurant managed by the Manager is centrally managed by the application.
- The application can be used by the restaurant for the online service or even in-house service.

OBJECTIVES

We can briefly classify the objectives of the project into the following:

- Service oriented.
- User oriented.
- Finance oriented.

Ø **Service oriented:**

- The project aims at developing a software which can satisfy the customers with the service.
- To be able to handle high number of orders at any point of time.
- To be able to consistently perform (both qualitatively and quantitatively) with less human interference.
- To automate the whole process.

Ø **User oriented:**

- To improve the communication between client and management without any gap.
- To improve the user experience by delivering a user-friendly and easy to understand system.
- To reduce the time needed to process / take an order.

Ø **Finance oriented:**

- Like all other systems, the project aims at reducing the cost of maintaining the restaurant.
- To be able to reach as many users as possible.
- To be able to adapt and improve based on the user feedback.

Specifications:

The project aims to have the below but not limited to features:

- Central login system for users, chef and the management.
- Have a user-friendly UI with menu from which users can choose from.
- Take orders from the users and pass it forward to the chef.
- Calculate the final bill for the user for an order.
- Payment options for the users to pay for their order.
- Maintaining all the data in a database.

Constraints / Limitations:

- A high performing system with capabilities of running an object-oriented software along with GUI and consistent network connection.
- Project will be built using Eclipse IDE, SQL DB and the system must be capable to run the software on the console.
- System must be capable of running parallel operations – to avoid scalability issues.
- A strong database which can provide secure authentication into the system.
- The application is best suited with a UI – either mobile application or web page; Implementation of the same is unsure due to time constraint.
- System must provide fault assurance and recovery in case of system failure.

APPROACH

On studying the problem description we see that different user-types such as managers, waiters, customers, etc. access the system which stores data about their transactions with one another. Based on this information, it is clear that a client-server architecture would be an ideal option for the design of this application.

The necessary steps to be taken are:

I. Develop a sign-up system on the client side that allows the users to access the system.
II. Create the server-side database system and establish a communication channel between the client and server
III. Develop the client's basic functionalities of viewing, inserting, modifying and deleting data in the server.
IV. Develop additional modules for user-type specific actions in the system.

For the development of the solution, we will be following the Agile methodology with multiple development iterations. Each iteration would involve the end-to-end development of one or more of the above-mentioned steps. Every iteration of the project will be subject to evaluation and testing.

Over the course of the development process several team meetings will be held. In each of these meetings all the developers would discuss their current progress and analyse how well the solution solves the problem. This will be determined by test results, performance feedback and by checking how well the developed modules handle the functional requirements. In the event that the majority of the development team feels that the current development trajectory isn't heading towards the project goal, then the subsequent iterations would be modified to accommodate the changes discussed in the meeting.

Below is one possible scenario along with the alternative approach that may be taken if and when it happens:

- Suppose the restaurant handles a very low volume of customers and if the restaurant runs with minimal staffing, then the usage of a server-side database becomes a very costly and inefficient option. A more efficient approach in this case would be to maintain a single POS system that runs the restaurant automation application. The storage of data can be handled using the host computer's file system. The advantage of switching to this approach is that it does not require any major changes to the application logic. This change can be made at any stage of the project's development. It is basically a light-weight alternative.

PROJECT MANAGEMENT

- Project Duration

  The entire project will be completed in the approximate time frame of around 1 month and 15 days.

- Iterations (Milestone)

  - **Iteration 1**

    - Flexible Login System along with the UI for the customers, manager and chef includes registration for new users and login for old users using the right credentials.
    - Create a relational database maintaining the login details.
    - Connectivity using JDBC.

  - **Iteration 2**

    - Detailed menu items for the user to choose from.
    - User can choose the items from the menu and then place an order.
    - The placed order must reach the Chef, which must be visible when the Chef logs on to the portal.

  - **Iteration 3**

    - Calculate the total price of the order placed by the customer.
    - Introducing various payment methods.

- Maintaining all the data in the database.

- **Schedule estimation of all tasks:**

Name of the Task	**Time Taken**	**Completed by**
Login System	8 hours	October 5
Relational Database	5 hours	October 8
JDBC Connectivity	3 hours	October 12
	ITERATION 1 COMPLETED	
Create Menu Items	4 hours	October 18
Customer Order	3 hours	October 23
Send Order to Chef	3 hours	October 28
	ITERATION 2 COMPLETED	
Calculate Total Order	3 hours	November 4
Add Payment Methods	6 hours	November 12
Maintenance of Database	5 hours	November 16
	ITERATION 3 COMPLETED	

- **Task Distribution**

Task	**Team Members**
Login System	Rahul, Harshita, Kritika
Relational Database	Avani, Siddharth,Rahul
JDBC Connectivity	Harshita, Kritika, Avani
Create Menu Items	Siddharth, Rahul, Harshita
Customer Order	Kritika, Avani, Siddharth
Send Order to Chef	Rahul, Harshita, Kritika
Calculate Total Order	Avani, Siddharth, Rahul

Add Payment Method	Harshita, Kritika, Avani
Maintenance of Database	Siddharth, Rahul, Harshita

- **Task Links**

- Login System - Eclipse/Netbeans connect with RDBMS using JDBC connectivity
- List of menu items attached with the Main page of the system
- Customer order linked with the Chef's profile and notified
- Total Cost Calculation leads to the next page which is the Payment Method page

DELIVERABLES

➢ **ITERATION 1 (Due date: October 13)**

- Design the Login page
- Create Login page which has option to login as a Customer, Manager or Chef.
- Code and implement the Login page
- Do Testing on the Code
- Peer Review of the Code
- Additional Changes if needed
- Code Acceptance and Finalization

➢ **ITERATION 2 (Due Date: October 29)**

- Decide upon the various menu items and categorize them
- Design the list of menu items
- Code and implement the addition of menu items
- Cumulate the chosen items as the customers order
- Code to show the customer order in the chef's profile (to-do items)
- Do Testing on the Code
- Peer Review of the Code

- Additional Changes if needed
- Code Acceptance and Finalization

➢ **ITERATION 3 (Due Date: November 17)**

- Update prices of the menu items
- Code to calculate the total cost of the order
- Code to add various payment methods
- Do Testing on the Code
- Maintain all the data in the database
- Confirm the connectivity of the database with the system
- Peer Review of the Code
- Additional Changes if needed
- Code Acceptance and Finalization

TEAM QUALIFICATIONS

Avani Sah

Pursued by Bachelors in Computer Science, then worked in the technical industry for a year and 9 months and worked on two products during this tenure. I have worked on these projects using Java, HTML, CSS, Angular JS, Angular 5, JavaScript, Python, IBM DataTool, Jinja Framework. Have used Eclipse and Microsoft Visual Studio for most of my coding with Java, HTML, etc. And, Jinja for Python.

During my undergraduate studies, I worked on a management project which managed platform independent online tests and had a feature 'dynamic difficulty levels', where as per the answers of the user, the difficulty level of upcoming questions changed.

Resume: https://docs.google.com/uc?export=download&id=1fzUeDLTlMUEs1TSqGQNJVTwj5m1LZg6z

LinkedIn: https://www.linkedin.com/in/avanisah/

Harshita Rastogi

I have experience of working as a software developer in Object Oriented Languages such as C++ and Java which will be beneficial in the completion of this project. Moreover, I have built a software which handles the real time student-faculty data of a training center which provides flexibility in the class schedule of the student and faculty. I have worked on the projects in Netbeans and Eclipse. Apart from C++ and Java, I know C, Python, SQL, HTML, CSS and Javascript. Most of my academic undergraduate projects include working in Machine Learning, Deep Learning and Natural Language Processing.

Resume: https://drive.google.com/file/d/10K_LzZAfA9Jq-hePw-vqsoJYmeM6zQAr/view?usp=sharing

Kritika Manchanda

I have worked on numerous large scale releases in the industry for 3 years and I bring to the table my ability to solve problems in a timely and efficient manner. My technical expertise includes extensive knowledge on C/C++. I have used Eclipse for most part of my work. I have worked in the data storage and networking domains.

I designed and implemented a library management system in my Undergraduate studies. It kept a record of all the books in the library, checked out books and amount dues.

Resume: https://drive.google.com/file/d/1RKCL4EtribKqA_oH1gexRDHcE68UsEO0/view?usp=sharing

Rahul Shivakumar

I am an experienced Software Engineer with a demonstrated history of working in IT with a sound financial domain knowledge. I have hands on experience for 3 years in Java, SQL and JavaScript. As part of my academic projects, I have also worked on Android and Arduino.

I designed and implemented an Online seat matrix system during my UG which managed the process of selecting universities for the students based on their rank in competitive exam and the number of seats available in the department for a university.

Resume: https://drive.google.com/file/d/14SzaSKQ5zvUPW-CQ1Dev7WY3lpO8ZUno/view?usp=sharing

Siddharth Chandrasekar

Resume: https://1drv.ms/b/s!ApDFq4eXLme-9BDFLb42-zpMzj7x?e=4ZLgAk

LinkedIn: https://www.linkedin.com/in/siddharth-chandrasekar-533624145/

I have worked on several end-to-end software development projects throughout my Undergraduate studies in Computer Science. I built a web-based University asset-management system from scratch, following the complete SDLC process. The system is currently in use at my previous college. Most of my development experience has been using Javascript and related web technologies, C#, C++ and Python. I do not have much experience working with Java, but I have learnt the basics of the language in a laboratory course. I am proficient in Object-Oriented Programming concepts, which will definitely be useful in the development of this application.