Tea Stall Management System

Second semester Project Report

BSc. Computer Science

COMSATS University ISLAMABAD

Sahiwal Campus

**Tea Stall Management System**

**FA20-BCS-122 Sidra Saleem**

**FA20-BCS-151 Aamna Zahid**

**FA20-BCS-141 Ali Raza Tareen**

11th January 2022

# **Abstract**

This report documents the process of designing, developing and testing a software system to be used in a Tea Stall; usually given the name Tea Stall management system. The Tea Stall management system is there to help communication between all workers within a stalls by minimising the probability of human errors. This report was written by Sidra Saleem, Aamna Zahid and Ali Raza Tareen as part of his 3rd Semester project.

Contents

[**Abstract** 2](#_Toc93248343)

[**Introduction** 3](#_Toc93248344)

[**The Problem** 4](#_Toc93248345)

[**Project Objective** 5](#_Toc93248346)

[**Existing Solutions** 5](#_Toc93248347)

[**Project Proposal** 5](#_Toc93248348)

[Table 1.proposed features 6](#_Toc93248349)

[Table 3 graph 6](#_Toc93248353)

[7](#_Toc93248354)

[**Modules of Tea Stall Management System** 7](#_Toc93248355)

[**System Design of Tea Stall Management System:** 8](#_Toc93248362)

[**5.diagram** 8](#_Toc93248372)

[**Entity Relationship Diagram** 8](#_Toc93248375)

[**6.Sequence diagram** 9](#_Toc93248386)

[**Design** 10](#_Toc93248387)

[**Introduction** 10](#_Toc93248388)

[**METHODS** 10](#_Toc93248389)

[**Graphical User Interface** 11](#_Toc93248390)

[**Order GUI** 11](#_Toc93248391)

[**Functional and Non-Functional Requirements of Tea Stall Management System** 11](#_Toc93248392)

[**Functional Requirements of Tea Stall Management System:** 12](#_Toc93248393)

[**Non-Functional of tea stall management system: –** 13](#_Toc93248396)

[**Non-functional necessities – are often divided into 2 main categories:** 13](#_Toc93248397)

[**Conclusion of Tea Stall Management System** 14](#_Toc93248398)

[**Bibliography** 14](#_Toc93248399)

# 

# **Introduction**

This system helps the tea stall owners to take orders and to deliver them easily. This system helps the customers to place orders easily and to make payments.

This system has all the features like making changes in orders, knowing the status of an order, making payment in online mode if the customer wants. This system also has the option to add different industries into the system so that the employees working in any company can make the order and as some companies have rules to pay for their employee’s order so, the payment can be taken from the company on the monthly basis.

Only those customers who are verified by the industry can make the orders from the company’s name. This system has the option to make changes to the menu and customers can choose the order from available items. This system also has the option to manage staff who is working for the tea stall. Staff’s attendance can be saved on this system and their monthly salary can also be calculated on this system.

# **The Problem**

According to a research article written by Horizons [7], in 2006 within the UK there was just over 26,000 tea stalls with 7 million food served that year. As this tea stalls sector was worth £7.61 billion, any Tea Stall generating a good business reputation could lead to the making of a very successful and profitable business. The problem for many businesses is to ensure that they not only attract new customers but to ensure they maintain their existing clientele. It has been argued many times that an existing customer is worth more to a business than a new customer as the cost to attract a new customer can be up to five times the cost to retain an old customer. An online article by Paul Lemberg [9], discusses the pros and cons of this argument. Within the Tea Stall sector, a customer is likely to return to the tea stalls in the future if they received an excellent customer service as well as appetising food. However, if they had to wait for an unreasonable amount of time or there was a mistake in the order, it’s very unlikely the customer would return. Therefore a solution to this problem would be to minimise mistakes within the order and bill, and help eradicate delays as well as encouraging team work and communication within the team. The next section will go into the objectives of the proposed solution.

# **Project Objective**

The objective of this project is to build an electronic tea stalls management system using all of the skills and techniques from the field ensuring that no common development mistakes are reproduced. Project management is critical to all software engineering projects and keeping to a project plan will be of similar importance. One of the main objectives of any business is to maximize profit by increasing efficiency and decreasing overheads1 without compromising customer satisfaction. Currently, many tea Stall use a paper-based system to communicate between the tea stalls and kitchen which can be shown to be one of the least efficient approaches. Even though this approach is implemented in successful profitable tea stalls, there are several problems which could be seen as reducing the tea Stall efficiency:

• Miscommunication caused by handwriting.

• Unmanageable order logging.

• Inefficient tea stalls -kitchen communication.

• Difficult order tracking and time management.

• Difficult stock management.

• Limited statistical output.

By introducing an electronic tea stalls management system these problems can be avoided or improved leading to an increase in profits.

# **Project Proposal**

The aim of this project is to create a tea stall management system that can incorporate the benefits of all the existing solutions but without any of the drawbacks as well as including many new features.

A list of proposed features can be found in table 1.

Many of the existing solutions to POS (Point-of-Sale) systems are sold with the required expensive hardware so for any business looking to work to a budget, the more enriched software solutions are just out of their range.

|  |  |
| --- | --- |
| Feature | Motivation |
| Automated stock control | Real-time view of ingredient stock levels so only the meals with enough ingredient stock can be sold. |
| Menu option and preference selection. | Flexible meal options available for the customer |
| Advanced discount function. | Calculating the best price for the customer |
| Order alerts | Kitchen and bar staff in direct communication with waiters allowing the kitchen to notify the waiter that service is required. |
| Flexible GUI design | Software capable of being used on any sized screen and so must have a flexible design. |
| Order logging. | All orders logged for future query generation |

## Table 1.proposed features

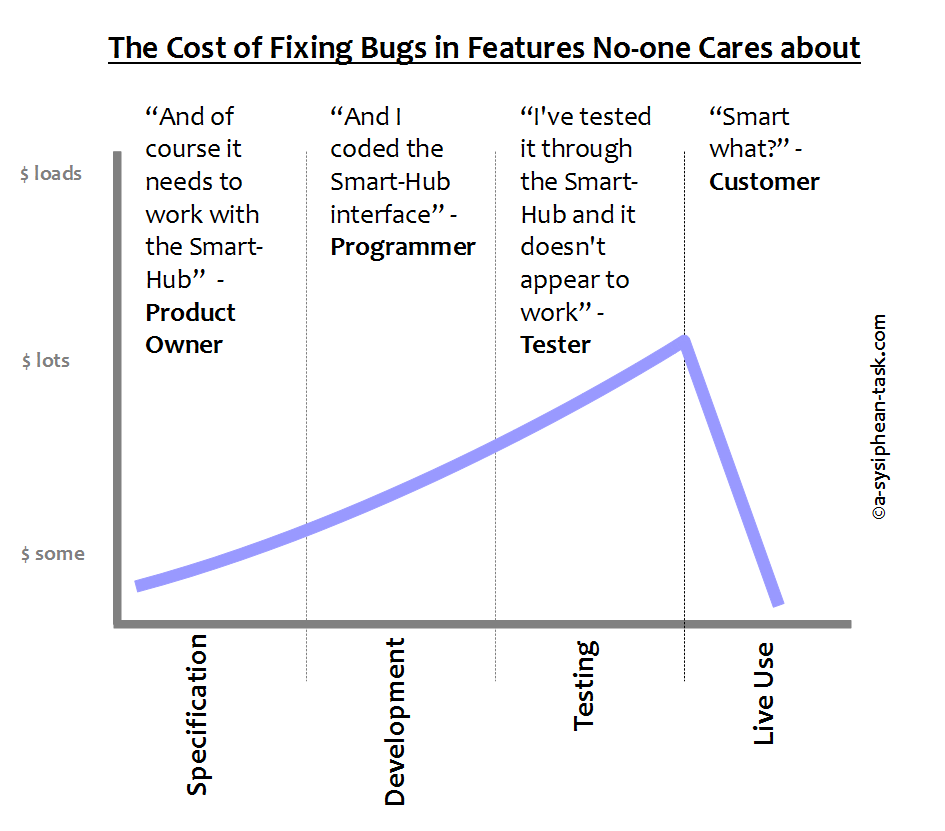
****

Table 3 graph

****

# **Modules of Tea Stall Management System**

Our system has only two main interfaces. One is for the admin and the other one is for the user, the here user can be a customer, an industry, or a tea seller.

Along with these, we have other interfaces which can be discussed as below:

## **Admin**:

This module can only have one account and this account has all the privileges which a user account might not have. First of all, an admin account is created, and then if the admin verifies a registering user, then his account will be created.

**Tea Stall:**

This module consists of tea sellers. The option to register as a seller or a customer is given at the registration time. This module contains the options to add items to the menu, managing staff, adding customers and taking payments, etc.

## **Customer:**

This module is different from the tea-seller module and a user registered as a customer cannot register as a seller. This module contains the options to make orders from tea sellers, adding items into order, managing accounts, making payments to tea Sellers’s and adding the industry in which he works, etc.

## **Items:**

This module contains all the items that the seller is selling. Tea seller can make changes into the list of items he sells. As an order contains many items, all the items that can be seen in the history of the order is old.

## **Order:**

This module is available in both the interfaces, one of the customers and the other of the seller. The customer can see the items he has ordered in order, can check the status of the order made, and can also see the old orders made by him as it saves the order history. In the tea seller’s module, he can see the order made by the customer and can update the status of the order and he can also the orders made by the customer in the past.

# **System Design of Tea Stall Management System:**

* The system needs to be operated by some authorized person as wrong hands can make it irresponsible.

## **Entity Tea stall:**

Every User detail needs to be stored in a proper manner with very needed attributes. All the other tasks can be performed only if the user is authenticated.

**Some of the features of the user are as follows:**

## **Primary key User id:**

The user id is system generated and unique, which can be referenced in any other entity. This mostly for recognizing it in a unique way anywhere in the program.

## **Username:**

Username is taken from the user and fed into this it is properly validated so that no mistake happens.

## **Number:**

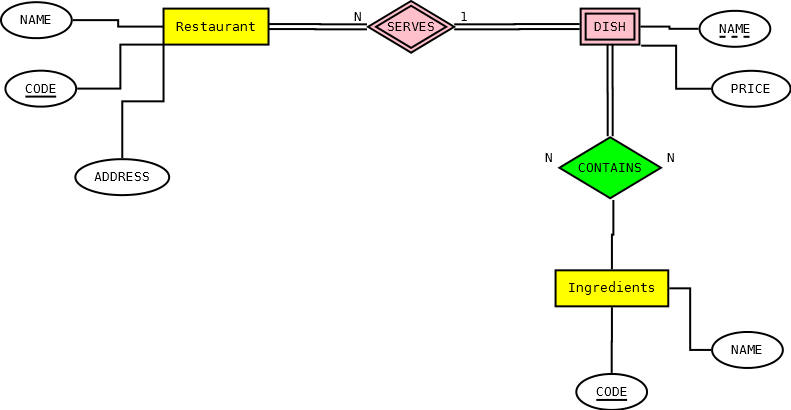
The phone number is taken here to keep the member updated and the confirmation of the event is also managed through this method.

## **Payments:**

In this section, the seller can see all the remaining payments he has to take from customers or from companies and can also see the payments they have to make to employees and others.

## **Address:**

The address of the user is required at the time the user is registering itself on the system. The address can be the shop address or the home address.

****

## **4. Erd of tea stall**

## **Entity Customer:**

The customer is the entity that requires it to order the items he wants from a tea seller if he wants to make payment to the seller. If any customer has any query, then there is also an option to put your query and it will be answered by your service provider.

### Entity payment:

This module is present in both the interfaces but the options that are provided are different. In the customer’s interface, there is an option to pay the bill using their debit card or by net banking. Customers can also see their payment status after paying their bills. In the Tea seller’s module, there is an option to take the bill.

## **5.diagram**

**Some features of payment entity are explained further:**

## **Number:**

The phone number is taken here to keep the member updated and the confirmation of the event is also managed through this method.

#### **Address**:

The address is required because if the customer is not working in any industry and is taking services then the bill of the orders can be taken from the given address if required. So, the address of the customer is required.

# **Entity Relationship Diagram**

An entity relationship (ER) diagram is a modelling language used to represent a type of semantic data model of a system. The ER diagrams are often used to represent a relational database and its requirements in a top-down fashion usually defined as the database schema. The database schema for this database has been split into two ER diagrams (Figures 4.3 and 4.4). Figure 4.3 graphically shows the objects and their relationships that are contained within a meal. The meal object will be made of at least two ingredients that can be either a normal ingredient or a prepared ingredient. Note, a prepared ingredient is a collection of ingredients used to either group commonly used ingredients or to group optional ingredients. Each ingredient will have a default and manual measurement with the default measurement entered on input of the ingredient and the manual measurement entered if the meal ingredient link requires a different amount. Also, each ingredient will be part of a generic ingredient object as there are many ingredients that are the same item but packaged in a different way at a different price. This allows the database to be in Boyce-Codd normalised state. An example of this would be the drink Coca-Cola which can be bought by bottle, can or draught, thus are the same item but packaged differently at a different price and amount. Finally each ingredient and prepared ingredient can be part of a category allowing optional ingredients to be interchanged with other ingredients in the same category. Figure 4.4 graphically shows the relationships for the menu, order and offer objects. The menu consists of a date time relationship that provides the intervals to when the menu is active and a menu section relationship that contains the colour variables and items under that particular menu section. The order consists of one to many suborders with the suborder consisting of one to many items. The order stores all the ingredients within each item and also the replaced ingredient if that optional ingredient was replaced. The offer consists of a date time relationship that provides the intervals to when the offer is active and a offer section relationship that contains the sets required by the offer.

## **Entity Admin:**

Admin is only a single user who can verify the user who is trying to register on this prison management system. If the admin verifies the user, then only the user can see the details and can make changes according to the given privileges. No other user can register as admin as there is only one account for admin.

## **Username:**

Username is taken from the user and fed into this it is properly validated so that no mistake happens.

## **Password:**

The password is required to login into the system and the password must contain a special character, a numeric character, and an uppercase letter.

## **Logs:**

As the admin manages all the accounts so he can see all the history of all accounts. In customer’s account log contains the items bought, date, amount, and the name of the seller from which the items are bought. In case, the customer made the transaction to the wrong seller, then it helps both the customer and the seller.

## **Verification:**

If the admin verifies a registering user only when a user can get itself registered on the system.

#### **Admin-id**:

Admin can keep anything as id because there is only one admin account and it does not matter what the id is.

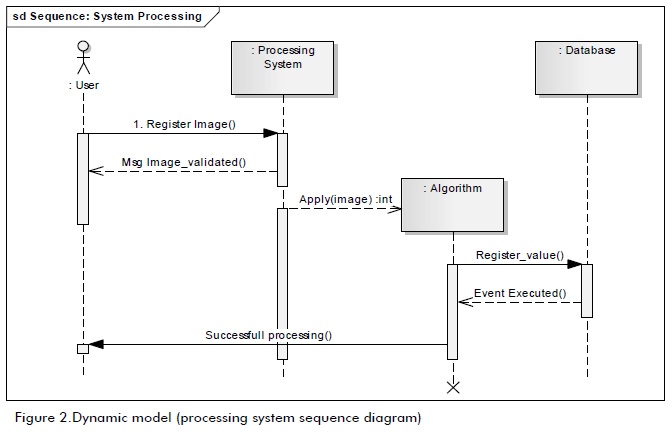
## **Entity Order:**

This entity talks about all the orders that are made by the customer to the seller or that are completed by a seller. The customer might want to pay the amount together for all the orders made. So, it is easy to look at all the orders made and pay the amount to the seller. This entity also helps the seller as the seller delivers the orders to many customers so he can easily calculate the total quantity he bought and the total quantity he sold.

### Items:

From this, the customer and the seller can know which items are delivered and which are remaining and can also know how much quantity is delivered.

## **6.Sequence diagram**

****

# **Design**

## **Introduction**

This project has been designed using numerous diagrammatic techniques. Recall from Section 2.6, that the most general modelling language to describe both the structure and behaviour of a software system is Unified Modelling language (UML).

Use case diagrams have already been used in the requirements analysis as a way to graphically overview the order process within the system. Other diagrams from the UML family are used in the design stage to show the structure and behaviour of numerous sophisticated design features.

## **METHODS**

RDBMS are one of the most popular data storage methods out in the market and offer many advantages including:

• Fast data extraction using structured query language (SQL).

• Good management of data and security through the management system.

• Good level of data consistency.

• Advanced features including functions and triggers.

• Requirement of a data model to be developed; leading to long term cost effectiveness.

In industry, there are numerous expensive highly functional RDMBSs including Oracle and SQL Server that are very popular and offer technical support. However, there are also numerous open-source solutions with many adjudged to be as good or better and are becoming even more popular with small scale software systems.

# **Graphical User Interface**

The graphical user interface (GUI) is the only component of the system that the user interacts with therefore is of great importance. The design had to be simple, clear and concise but whilst also showing all of the required features. The main objective was to create a GUI that allowed the user to get the order completed in the minimum time possible. This was judged by the time taken to complete the order as well as the amount of clicks required to get from the start to the finish. In total, there are 3 different GUIs within the tea stall management system with each GUI requiring a different design specification.

## **Order GUI**

The order system GUI has the most user interaction through the means of a touch screen. Hence usability and user-friendliness of the GUI was of the highest priority. Therefore the specification was as follows:

• Chronological order of steps; either from left to right, top to bottom or a mixture of both.

• Minimum clicks from start to finish.

• Meal items with optional images.

• Ability to fit on any monitor size.

• Maximum space usage with readable font.

• Colour coordinated meal buttons to display the stock status.

# **Functional and Non-Functional Requirements of Tea Stall Management System**

## **Functional Requirements of Tea Stall Management System:**

The functional requirement of this is that it does what it is meant for. A functional requirement describes what a software system should do, while non-functional requirements place constraints on how the system will do so. Functional requirements specify a function that a system or system component must be able to perform.

It can be documented in various ways. The most common ones are written descriptions in documents and use cases.

A few of its functional requirements are as given below: –

## **User Data Should Be Fed into the System:**

This system is doing that properly in the user entity. After fetching the data of the customer, the bill can be made and sent to the customer.

## **Admin Can Add the Users:**

Admin can verify and add the user, which is they are doing with this system. He can also delete the users if required.

# **Non-Functional of tea stall management system: –**

Non-functional necessities square measure the other demand than practical necessities. These square measures the necessities that specify criteria which will be wouldn’t to choose the operation of a system, instead of specific behaviors.

# **Non-functional necessities – are often divided into 2 main categories:**

* Execution qualities, like security and usefulness, that square measure evident at the runtime.
* Evolution qualities, like liabilities, maintainability, flexibility, and quantifiable, that square measure embodied within the static structure of the code.

Non-functional teal stall management system necessities place restrictions on the merchandise being developed, the event method, and specify external constraints that the merchandise should meet. Our project qualifies all the criteria of functional and non-functional accordingly and the system is up to mark performance vice.

Unlike ancient style wherever the goal is to form the thing or application physically enticing, the goal of interface style is to form the user’s interaction expertise as straightforward and intuitive as attainable – what’s typically known as user-centered style.

Where smart graphic/industrial style is daring and eye-catching, smart interface style is commonly delicate and invisible.

**Keep things simple and consistent: –**

Simple and Harmonic way making UI is very intuitive and needs to follow.

**Make good use of typography: –**

The typography is taken care very strictly as the need of the system.

**Use colour and contrast properly: –**

Color combo of lite and dark is a good way emphasise and done well in this system.

**Consider feedback messages: –**

The feedback form is a very good way taking feedback of forms and improving the system.

# **Conclusion of Tea Stall Management System**

Finally, in the tea stall management system, all the hard work is done for tea stall management system is here. It is software that helps the tea sellers to increase their customers, increase their work, and increase their income source. This system helps the customers to make orders and make payments of their orders easily.

The User Interface of it is very friendly and can be easily used by anyone. This system has different user interfaces for the different user types. It also decreases the amount of time taken to write details manually and in other modules. In the end, it can be said that it is serving all its purposes very smoothly and efficiently for which it is made.

# **Bibliography**

[1] Ian Alexander. Computing & control engineering. Volume 14(Issue 1):22–26, April 2003.

[2] Ezee Burrp. Software Features. Accessed on 14 October, 2009. <http://www.ezeeburrp.com/> features.html.

[3] HSQL db. Database comparison. Accessed on 13 September, 2009. <http://hsqldb.org/images/> imola\_retrieve.jpg.

[4] Business Dictionary. Stakeholder. Accessed on 2 April, 2010. <http://www.businessdictionary>. com/definition/stakeholder.html.

[5] Geoffrey Elliot. Global business information technology: an integrated systems approach. Pearson Education, 2004.

[6] Beefeater Grill. Beefeater Grille. Accessed on 24 September, 2009. <http://www.beefeater.co>. uk/ourFoodAndDrink.

[7] Horizon. tea stall Sector Overview. Accessed on 4 May, 2010. <http://www.caterersearch>. com/Articles/2006/05/12/306737/ tea stall -sector-overview.htm.

[8] Java JDBC. Java SE Technologies - Database. Accessed on 3 May, 2010. <http://java.sun.com/> javase/technologies/database/.

[9] Paul Lemberg. Which is Better: New customers or repeat business?. Accessed on 12 October, 2009. http://www.businessknowhow.com/marketing/new-customers.htm.

[10] Agile Manifesto. Principles behind Agile Manifesto. Accessed on 20 September, 2009. http: //www.agilemanifesto.org/principles.html.