|  |
| --- |
| Information Gathering |
|  |

Project Name: Restaurant Management System

Submitter Name: Deepak Kumar

Kumari Shikha

Shourabh Rathi

Abhishek Mittal

Date Submitted: 29/08/19

|  |
| --- |
| **About Restaurant Management System** |

The main focus of the project was to create a single working “point of management” restaurant system that acts as both a terminal for taking orders and a terminal for generating reports and making changes to employees or items on the menu. Project planning was done to define the scope of the project, assess risks, and estimate and schedule project activities and thereby lay the foundation for the execution, monitoring and control of the project.

|  |
| --- |
| **Explain different functionalities of Restaurant Management System** |

The system Restaurant Management System has following major functionalities:-

1. It is easy to use the order with all the options provided in the web page.
2. The end user can access the portal with simple registration.
3. User can access online payment for their ordered foods.
4. The webpage provides different options to choose the mode of payments.
5. User can check their order status.
6. Customer can g**i**vetheir feedback related to their orders.

|  |
| --- |
| **Analysis of** **Existing Restaurant Management System** |

The existing Restaurant Management System does not allow us to book tables online and we can’t check the Menu Online. So, it is difficult for us to know the items available at the Restaurant.

|  |
| --- |
| **Challenges in existing Restaurant Management System** |

1. **Disappointed diners.** Probably the biggest problem was that what was supposed to be a quiet, enjoyable Friday night for the customers turned into a nightmare of excuses and waiting. Many families ended up eating their main course at different times, as many people who had ordered the missing steak had to wait to be served the replacement dish. I honestly couldn’t blame any customer who wouldn’t want to patronize the restaurant again after this experience.
2. **Slow service.** The delayed service made the night feel longer for everyone. Every single order took longer than usual to go through, testing everyone’s patience and destroying the flow. By the end of the night the customers were fed up and the staff was stressed out and drained, which didn’t help the quality of service.
3. **Confusion and errors.** The fact that we had to recreate all the customers’ orders and receipts created a lot of chaos and errors along the way. On top of it, all the manual work we had to perform delayed service even more.
4. **A whole day wasted aligning data**. After recreating all of the customers’ orders and receipts (which was a time-consuming enough task), it took a whole day of aligning data to make sure that all the old and new receipts matched, and that all the customers had been billed correctly.

Synopsis Of

**“Restaurant Management System”**

Submitted By

Deepak Kumar - 1616110072

Kumari Shikha - 1616110108

Shourabh Rathi - 1616110201

Abhishek Mittal - 1616110007

Under the guidance of

**Mrs. Deepa Basantani**

In partial fulfillment of

**Bachelor of Technology**

Krishna Engineering College, Mohan Nagar

Ghaziabad

Affiliated to AKTU

August - 2019

**INDEX**

Sr. No. Topics Page No.

1 Introduction

2 Motivation

3 Objective(s) and Scope

4 Target Users

5 Technical Platform

1. **INTRODUCTION**

Major software functions of the Restaurant Management System include managing orders, inventory, and employee records, and generating reports.

Order management includes creating and deleting orders, adding and removing items from an order and closing orders. Orders should also be stored in the database to be used to calculate total sales. Inventory management includes adding new products, deleting products and updating products and resources.

The software is responsible for a number of other functions. Our software must be able to add employees, edit their information, and remove employees from the employee database. Menu items must be added, edited, and deleted from the menu item database. Items that can be ordered must be able to be added and removed from an order. All employees must be able to clock in and clock out. Servers must be able to do what all employees do as well as take orders. Managers should be able to do what all employees do and be able to edit item and employee information and generate reports. Reports that should be generated include sales reports showing sales by food category and the total sales from the start of the day. Orders should also be stored in the database to be used to calculate total sales.

1. **MOTIVATION**
2. **Disappointed diners.** Probably the biggest problem was that what was supposed to be a quiet, enjoyable Friday night for the customers turned into a nightmare of excuses and waiting. Many families ended up eating their main course at different times, as many people who had ordered the missing steak had to wait to be served the replacement dish. I honestly couldn’t blame any customer who wouldn’t want to patronize the restaurant again after this experience.
3. **Slow service.** The delayed service made the night feel longer for everyone. Every single order took longer than usual to go through, testing everyone’s patience and destroying the flow. By the end of the night the customers were fed up and the staff was stressed out and drained, which didn’t help the quality of service.
4. **Confusion and errors.** The fact that we had to recreate all the customers’ orders and receipts created a lot of chaos and errors along the way. On top of it, all the manual work we had to perform delayed service even more.
5. **A whole day wasted aligning data**. After recreating all of the customers’ orders and receipts (which was a time-consuming enough task), it took a whole day of aligning data to make sure that all the old and new receipts matched, and that all the customers had been billed correctly.
6. **OBJECTIVE(S) AND SCOPE**

**Objectives of the system are:-**

Major software functions of the restaurant management system include managing orders, inventory, and employee records, and generating reports.

Order management includes creating and deleting orders, adding and removing items from an order and closing orders. Orders should also be stored in the database to be used to calculate total sales. Inventory management includes adding new products, deleting products and updating products and resources.

The software is responsible for a number of other functions. Our software must be able to add employees, edit their information, and remove employees from the employee database. Menu items must be added, edited, and deleted from the menu item database. Items that can be ordered must be able to be added and removed from an order. All employees must be able to clock in and clock out. Servers must be able to do what all employees do as well as take orders. Managers should be able to do what all employees do and be able to edit item and employee information and generate reports. Reports that should be generated include sales reports showing sales by food category and the total sales from the start of the day. Orders should also be stored in the database to be used to calculate total sales.

**Modules in the System:-**

The Restaurant Management System can be divided into two modules, namely:

**Module 1: End User:**

This module is responsible for doing all these functions:

1. It can Book Tables.
2. It can check the Menu.
3. It can choose between different Meal Courses.

**Module 2: Admin:**

This module is responsible for doing all these functions:

1. It can confirm the Tables that are Booked.
2. It can add New Items in the Meal Courses.
3. **TARGET USERS**

The following users are identified as the targeted user of the system:-

1. Every user who don’t want to wait for dining.
2. Users who are shy to go to restaurant to see menu items and to choose between them.
3. **TECHNICAL PLATFORM:**

|  |  |
| --- | --- |
| **Operating System:** | Windows XP,Windows7 |
| **Language Requirements**: | Java EE, HTML, JSP |
| **Web Browser:** | Google Chrome, Mozilla Firefox, IE ver. 7 and above |
| **Database Technology:** | MySQL |
| **Tools & Development:** | Apache Tomcat 9.0.01 Server, Eclipse IDE |

*RESTAURANT MANAGEMENT SYSTEM*

project Understanding Document

*VERSION 1.0*

*29/08/2019*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | *Deepak Kumar, Kumari Shikha, Shourabh Rathi, Abhishek Mittal* | *08/29/19* | *Mrs. Deepa Basantani* | *08/29/19* |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**TABLE OF CONTENTS**

[1 introduction 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc218)

[1.1 Purpose of Project 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc219)

[2 project And Product Overview 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc223)

[3 Scope 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc226)

[3.1 Objectives 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc227)

[3.2 High-Level Requirements 5](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc234)

[3.3 Business Rules and Special considerations 5](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc263)

[3.4 Major Deliverables/Milestones 5](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc265)

[4 Duration 5](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc293)

[4.1 Timeline 5](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc294)

[6 project understanding approval 6](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\02_CF_Project_Understanding_Template_Version1.0.doc#_toc298)

# introduction

Major software functions of the Restaurant Management System include managing orders, inventory, and employee records, and generating reports.

Order management includes creating and deleting orders, adding and removing items from an order and closing orders. Orders should also be stored in the database to be used to calculate total sales. Inventory management includes adding new products, deleting products and updating products and resources.

The software is responsible for a number of other functions. Our software must be able to add employees, edit their information, and remove employees from the employee database. Menu items must be added, edited, and deleted from the menu item database. Items that can be ordered must be able to be added and removed from an order. All employees must be able to clock in and clock out. Servers must be able to do what all employees do as well as take orders. Managers should be able to do what all employees do and be able to edit item and employee information and generate reports. Reports that should be generated include sales reports showing sales by food category and the total sales from the start of the day. Orders should also be stored in the database to be used to calculate total sales.

## Purpose of Project

The main focus of the project was to create a single working “point of management” restaurant system that acts as both a terminal for taking orders and a terminal for generating reports and making changes to employees or items on the menu. Project planning was done to define the scope of the project, assess risks, and estimate and schedule project activities and thereby lay the foundation for the execution, monitoring and control of the project

**Benefits of the Project:**

The system includes the students, parents, tutors. Benefits to each of these are described below.

* **Front of house** solutions enables the guest experience to be the best possible. These systems are seen by customers and include reservations and point of sale.
* **Back of house** solutions are responsible for the retail and manufacturing aspects and provide information about your restaurant’s labor and raw materials. Examples of back of house restaurant software include employee scheduling and inventory.
* **Loyalty program** solutions provide customers with incentives to return to your restaurant or spend more per visit. Gift cards and guest programs are types of loyalty program solutions.
* **Human resources** solutions enable the restaurant to employ and retain the most qualified employees. [Training](https://www.deputy.com/blog/customer-service-training-ideas-to-help-build-brand-advocacy) and terminating employees are part of this solution.

# project And Product Overview

The main focus of the project was to create a single working “point of management” restaurant system that acts as both a terminal for taking orders and a terminal for generating reports and making changes to employees or items on the menu. Project planning was done to define the scope of the project, assess risks, and estimate and schedule project activities and thereby lay the foundation for the execution, monitoring and control of the project

**Product Features:**

* *It is easy to use the order with all the options provided in the web page.*
* *The end user can access the portal with simple registration.*
* *User can access online payment for their ordered foods.*
* *The webpage provides different options to choose the mode of payments.*
* *User can check their order status.*
* *Customer can g****i****ve**their feedback related to their orders.*

# Scope

Major software functions of the restaurant management system include managing orders, inventory, and employee records, and generating reports.

Order management includes creating and deleting orders, adding and removing items from an order and closing orders. Orders should also be stored in the database to be used to calculate total sales. Inventory management includes adding new products, deleting products and updating products and resources.

The software is responsible for a number of other functions. Our software must be able to add employees, edit their information, and remove employees from the employee database. Menu items must be added, edited, and deleted from the menu item database. Items that can be ordered must be able to be added and removed from an order. All employees must be able to clock in and clock out. Servers must be able to do what all employees do as well as take orders. Managers should be able to do what all employees do and be able to edit item and employee information and generate reports. Reports that should be generated include sales reports showing sales by food category and the total sales from the start of the day. Orders should also be stored in the database to be used to calculate total sales.

## High-Level Requirements

The following table presents the requirements that the project’s product, service or result must meet in order for the project objectives to be satisfied.

| Req. # | I Requirement Description |
| --- | --- |
| Admin Login | To Provide access to the Admin to add new items in the Menu. |
| Booking | To book table for the user by taking his Details. |
| Availability Of Tables | Tables are limited to be booked. |
| Network Connectivity | The user should have fast internet connection and should be  supported by an advanced browser. |
| Availability Of New Items | Only valid member can access to add new Items. Whenever he  Wants to access he should login along with the Admin Login and Password. |

## Major Deliverables/Milestones

[Provide a list of the major deliverables/milestones that will be completed by the end of this project. A deliverable is any unique and verifiable product, result or capability to perform a service that must be produced in order to complete a process, phase or project. A milestone is a key performance indicator that is typically reported to executives to indicate the project’s progress.]

| Major Deliverable | I Deliverable Description |
| --- | --- |
|  |  |
|  |  |
|  |  |

# 

# Duration

## Timeline

[Provide an estimate of the project duration (e.g., 18 months). You may provide a high-level timeline for the project if information is available at this time. This time estimate will be further refined in the Planning Phase of the project. If applicable, also state the expected life of the product. An example of a high-level timeline is provided below.]

System Development

Completed

Developed Prototype

Requirements Analysis

Completed

Project Plan Completed

System Development

Completed

Developed Prototype

Requirements Analysis

Completed

Project Plan Completed

23/08

24/08

26/08

31/08

# project understanding approval

The undersigned acknowledge they have reviewed the **Project Understanding Document** forthe **RESTAURANT MANAGEMENT SYSTEM** project. Changes to this document will be coordinated with and approved by the undersigned or their designated representatives.

[List the individuals whose signatures are desired. Examples of such individuals are Business Steward, Project Manager or Project Sponsor. Add additional lines for signature as necessary. Although signatures are desired, they are not always required to move forward with the practices outlined within this document.]

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | DEEPA BASANTANI |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

APPENDIX A: REFERENCES

[Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.]

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| *Restaurant Management System* | *RMS is used to check Menu of Restaurant online and to book tables online.* | *<URL or Network path where document is located>* |

APPENDIX B: KEY TERMS

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Credit/Debit Card** | means any credit/debit card issued on the account to the holder and must be a valid card. |
| **VISA/MasterCard** | types of cards issued by the banks which may be used for making payments. |
| **Transaction** | Transfer of monetary value from one bank account to another. |
| **Web Portal** | website bringing together information from diverse sources in a unified way. |
| **Authentic** | valid things, refers to the truthfulness of origins, attributions and commitment. |
| **Contemporary** | relating to the current world |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Software Requirements Specification

For

Restaurant Management System

Table of Contents

1. Introduction

1.1 Purpose

1.2 Intended Audience and Reading Suggestions

1.3 Project Scope

1.4 Product Feature

1.5 User Classes and Characteristics

1.6 Technologies to be used

1.7 Design and Implementation constraints

1.8 User documentation

1.9 Assumption and Dependencies

2. System Features

2.1 Registering new user in Database

2.2 Secure Login to the interface

3. External Interface Requirements

4. Other non-functional Requirements

5. Other Requirements

# Introduction:

## Purpose

The main focus of the project was to create a single working “point of management” restaurant system that acts as both a terminal for taking orders and a terminal for generating reports and making changes to employees or items on the menu. Project planning was done to define the scope of the project, assess risks, and estimate and schedule project activities and thereby lay the foundation for the execution, monitoring and control of the project

## Intended Audience and Reading Suggestions

This document is intended to provide a clear picture of the system for the users i.e., Admin and Customer. And the SRS document got divided into sections which are classified as the scope of the project, the overall description about the system, the system features, external interface requirements as well as the non-functional requirements.

## Project Scope

Major software functions of the restaurant management system include managing orders, inventory, and employee records, and generating reports.

Order management includes creating and deleting orders, adding and removing items from an order and closing orders. Orders should also be stored in the database to be used to calculate total sales. Inventory management includes adding new products, deleting products and updating products and resources.

The software is responsible for a number of other functions. Our software must be able to add employees, edit their information, and remove employees from the employee database. Menu items must be added, edited, and deleted from the menu item database. Items that can be ordered must be able to be added and removed from an order. All employees must be able to clock in and clock out. Servers must be able to do what all employees do as well as take orders. Managers should be able to do what all employees do and be able to edit item and employee information and generate reports. Reports that should be generated include sales reports showing sales by food category and the total sales from the start of the day. Orders should also be stored in the database to be used to calculate total sales.

**1.4** **Product Features**

It is easy to use the order with all the options provided in the web page.

The end user can access the portal with simple registration.

User can access online payment for their ordered foods.

The webpage provides different options to choose the mode of payments.

User can check their order status.

Customer can g**i**vetheir feedback related to their orders.

## 1.5 User Classes and Characteristics

The user classes will be User, Administrator.

## 1.6Technologies to be used

## Programming languages:

**JAVA EE**: Java Enterprise Edition is a programming platform— part of the Java Platform-for developing and running distributed multi-tier architecture Java applications, based largely on modular software components running on an application server.

**HTML:** Hyper Text Markup is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document and to supplement that text with interactive forms, embedded images, and other objects.

**JSP:** A **C**lient**-**side scripting language used to create dynamic web content and user interface.

**MySQL:** The data of this application can be stored in the database.

**Tools and Development Requirement:**

**Apache Tomcat 9.0.01 Server:** Apache Tomcat is a Servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

**Eclipse:** Eclipse is an integrated development environment used in computer programming, and in 2014 was the most widely used Java IDE in one website's poll. It contains a base workspace and an extensible plug-in system for customizing the environment.

## 1.7Design and Implementation Constraints

The design of the software included both architectural and subsystem design. The architecture of software defines the major software subsystems and the dependencies and interrelationships among subsystems. Architectural styles define a vocabulary for different classes of architectures.

## 1.8User Documentation

* The user should be familiar with the Digital Marketing related terminology like Order/Payment.
* The user should be familiar with the Internet.

## 1.9 Assumptions and Dependencies

* The details related to the products, payment and service transaction provided online.
* Administrator is created in the system already.
* Roles and tasks are predefined.
* Roles and responsibilities are already established.

# System Features:

# 2.1 Registering new user in Database

**2.1.1 Description and Priority**

This feature will enable the new user to enter his/her basic information in the database, so that the generation of profile for them may be done.

**2.1.2 Stimulus/Response Sequences**

This form will consist of basic fields such as Name, Username, E-mail Id, availability. There are two buttons: Register and Reset. Register will submit the data to the database at the server tier, and as expected Reset will reset the input values of all the fields.

**2.1.3 Functional Requirements**

The most important requirement here is to input values in the database and store them there for future use. To implement the security, the user has to enter and the id is generated during every order.

**2.2 Secure Login to the interface**

**2.2.1 Description and Priority**

This feature will enable the user to have a secure and simple login to the system. To avoid handling a large number of errors and exceptions this feature will enable the user to provide only a limited number of inputs having constraints upon them and if there are any errors the system will notify the user about them.

**2.2.2 Stimulus/Response Sequences**

It will consist of three basic fields Username, Password and Id. There are two buttons: Login and Forgot password. Login will submit the entered data for approval followed by access, and Forgot password button will change the details of the user.

**2.2.3 Functional Requirements**

The most important function is to only grant access to users that are listed in the database. The customer will provide the information on who will be allowed access. To implement the security, the web page must check the database to see if the Username, Password and Id are valid. If they are not, the user will receive an “Enter correct username, password and id” as a response.

1. **External Interface Requirements:**

# User Interfaces

The user interface is screen shown on the browser. The Home screen of the Web-Portal is where user can register and login. The portal screen acts as an interface to provide services to the user which are to be availed from the database.

# Hardware Interfaces

A minimum of 40GB of HDD, with Pentium IV processor, a minimum of 256MB of RAM

so that a suitable OS (Windows XP) may be installed, and a reliable internet

connection is required for the client side/user side so that may be accessed easily.

## Software Interface

**The system uses:**

**JSP:** Java Server Pages. It is a technology that helps software developers serve dynamically generated web pages based on HTML, XML and other document types; uses java programming language.

**Servlet:** Java web-containers which holds actions to be performed; a Servlet a java programming language class used to extend the capabilities of servers that host applications access via a request response programming mode.

d**. Communications Interfaces**

Internet connection and Browser are required in order for several functions to be

executed such as downloading. The system uses the following browsers:

* + - Google Chrome.
    - Mozilla Firefox.
    - Internet Explorer and so on.

# Other Non-functional Requirements:

# Performance Requirements

Some Performance requirements identified is listed below

* 1. The database must be support-many orders.
  2. Can support many users at the same time.
  3. High speed internet.

## Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Specific requirements in this are could include the need to:

* + - 1. Keep specific log or history data sets.
      2. Check data integrity for critical variables.
      3. Communication needs to be restricted when the application is validating the users.
      4. Providing Authentication.

**c.** **Software Quality Attributes**

There are a number of attributes of software that can serve as requirements. It is important that required attributes should be specified so that their achievement can be objectively verified. The following terms provide a partial list of examples

* **Portability**

Some of the attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include: Java is used to develop the product, so it is easiest to port the software in any environment.

* **Maintainability**

The user will be able to reset all options and all stored user variables to default settings.

* **Reliability**

Some of the attributes identified for the reliability is listed below:

1. All data storage for user variables will be committed to the database at the time of entry.

2. Data corruption is prevented by applying the possible backup procedures and techniques.

* 1. **Usability requirements**

Some of the usability requirements identified for this system are listed below:

* + - 1. A logical interface is essential to an easy to use system, speeding up common tasks.
      2. Error prevention is integral to the system and is provided in a number of formats from sanity checks to limiting free-text input.

* 1. **Availability**

Cached data will be rebuilt during every startup. There is no recovery of user data if it is lost. Default values of system data will be assigned when necessary.

# Other Requirements:

**Immediate Feedback:**

The System must try to answer all the queries of the user and it should provide immediate feedback after getting any request from users. The system must provide the illusion to the user that they are in contact to administrator of the web-site.

**Make the Interface Simple as Possible:**

The System must provide the simple and easy interface for beginners and also provide facilities for technical peoples who are using the system. The interface must be simple as possible.

**Reduced Time:**

To perform any task time is one of the important factors to consider. If the system not utilize properly time than the entire aim of system is fails and the system is fails to reach its goal, so time take to process all these activities should be less but the output should be effective.

Appendix A: Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | Collection of all the information monitored by this system. |
| Administrator | Is a person responsible for maintaining one or many website the duties of the webmaster may include ensuring that the web servers, hardware and software are operating correctly, designing the website, generating and revising web pages, replying to user comments, and examining traffic through the site. |
| HTML | Hypertext Transfer Protocol is a transaction- oriented client/server protocol between a web browser & a Web Server |
| HTTPS | Secure Hypertext Transfer Protocol is a HTTP over SSL (secure socket layer). |
| SRS (Software Requirements Specification) | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Customer |
| IDE | An integrated development environment (also termed integrated design environment, integrated debugging environment or interactive development environment) is a [software application](http://en.wikipedia.org/wiki/Software_application) that provides comprehensive facilities to [computer programmers](http://en.wikipedia.org/wiki/Computer_programmer) for [software development](http://en.wikipedia.org/wiki/Software_development) |
| Email | Electronic mail, commonly known as email or e-mail, is a method of exchanging  digital messages from an author to one or more recipients |

# Appendix B: Analysis Models

Under the analysis model, we analyze the system to check the following:

1. Whether it meets the requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics.

To perform these analyses of the model, the following testing is to be implemented:

**Unit testing**: Unit testing, also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

**Integration testing**: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang").

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

**System Testing**: system testing is done to ensure whether the system meet all the requirements stated in the SRS.

System testing is performed on the entire system in the context of a [Functional Requirement](http://en.wikipedia.org/wiki/Functional_requirements) Specification(s) (FRS) and/or a [System Requirement](http://en.wikipedia.org/wiki/Requirements_analysis) Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

The system testing is categorized into three:

#### Alpha testing

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

**Beta testing**

Beta testing comes after alpha testing and can be considered a form of external [user acceptance testing](http://en.wikipedia.org/wiki/User_acceptance_testing). Versions of the software, known as [beta versions](http://en.wikipedia.org/wiki/Beta_version), are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or [bugs](http://en.wikipedia.org/wiki/Computer_bug). Sometimes, beta versions are made available to the open public to increase the [feedback](http://en.wikipedia.org/wiki/Feedback#In_organizations) field to a maximal number of future users.

**Acceptance testing**

Acceptance testing performed by the customer, often in their lab environment on their own hardware, is known as [user acceptance testing](http://en.wikipedia.org/wiki/User_acceptance_testing) (UAT). Acceptance testing may be performed as part of the hand-off process between any two phases of development.

# Appendix C: Issues List

*The problems that might occur with the software product are:*

* It might occur that a customer searches for a particular product but the content is not available.
* During order the product an error might occur.
* The portal is dependent on web services. Though we make every effort to ensure that services are provided on time but there is no guarantee. The problems that might occur are:
  + Some web pages are lost. The user is looking for a specific Web page but try as they might, they can't find it.
  + Web pages load slow or incorrectly. The user found the Web page he wanted but it took forever to load or things are jumping around on the page while loading.
  + **JavaScript Errors.**

Forms are completely broken. After clicking submit button, an error might occur.

* Big security vulnerability. Someone to steal your login information and hack into your account
* Broken Registration Process
* Site won't load. Websites are supposed to work fine whether you type in the "www" or not. But an error might occur.

**SOLUTION ARCHITECTURE DOCUMENT**

**~RESTAURANT MANAGEMENT SYSTEM**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version*** | ***Date*** | ***Author*** | ***Reason for Change*** |
| *1.0* |  | *Abhishek Mittal, Deepak Kumar, Kumari Shikha, Shourabh Rathi* |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1 General Information 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\05_CF_Solution_Architecture_Template_Version1.0.doc#_Toc219544091)

[1.1 Introduction 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\05_CF_Solution_Architecture_Template_Version1.0.doc#_Toc219544092)

[2 J2EE – Solution Architecture 4](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\05_CF_Solution_Architecture_Template_Version1.0.doc#_Toc219544093)

[2.1 High Level Architecture 5](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\05_CF_Solution_Architecture_Template_Version1.0.doc#_Toc219544094)

[2.2 Description 6](file:///C:\Users\Abhishek%20Mittal\Desktop\Restuarant%20Management%20System\Templates\05_CF_Solution_Architecture_Template_Version1.0.doc#_Toc219544095)

# **General Information**

The J2EE platform is a multi-tiered system. A tier is a logical or functional partitioning of a system. When the developers are not disciplined, the display logic, business logic and database logic are muddled up and/or duplicated in a

2- tier client server system.

The advantages of the multi-tier architecture are:

* Forced separation of user interface logic and business logic.
* Business logic sits on small number of centralized machines (may be just one).
* Easy to maintain, to manage, to scale, loosely coupled etc.

Each tier is assigned a unique responsibility in a 3-tier system. Each tier is logically separated and loosely coupled from each other, and may be distributed.

The advantages of a 3-tiered or n-tiered application: 3-tier or multi-tier architectures force separation among presentation logic, business logic and database logic. Let us look at some of the key benefits:

* **Manageability**: Each tier can be monitored, tuned and upgraded independently and different people can have clearly defined responsibilities.
* **Scalability**: More hardware can be added and allows clustering (i.e. horizontal scaling).
* **Maintainability**: Changes and upgrades can be performed without affecting other components.
* **Availability**: Clustering and load balancing can provide availability.
* **Extensibility**: Additional features can be easily added.

## **Introduction**

The **Goal** of the proposed system is to create a web–based application with the following capabilities**:**

* **Performance and scalability:** System should be efficient and scalable.
* **Code and Design Reuse:** Code reuse decreases the cost of development and increases the stability of the code. Also, following the best design practices and architectural and design patterns will enable us to reduce the risk of inconsistent design.
* **Logical Functional Decomposition:** Every class in the design will have a clearly defined responsibility to play in the application. This will result in an application, which is easier to understand, maintain and extend.
* **Minimize Network Traffic**. Avoid transmitting unnecessary and redundant data. To achieve this goal, we would be using value objects to pass the data to and from the web layer to the business layer and the DB layer.
* **Maintainability:** This architecture results in a system that would be easy to maintain. The configuration would be controlled primarily by configuration file, which would control the application flow and logic.
* **Reliable:** The developed system must be reliable enough to avoid unexpected behavior of the system and each of the methods, procedures and functions doing Insert, Update, Delete, Create Table or Select must include error management.

# **J2EE – Solution Architecture**

J2EE Solution Architecture in enterprise architecture is a kind of architecture domain, that aims to address specific problems and requirements, usually through the design of specific information systems or applications. The solution architecture is required to implement solutions to meet business requirements and ensures alignment with the Enterprise Architecture.

## **High Level Architecture**

HttpServletRequest

HttpServletResponse

HTML, DHTML,

JavaScript,

CSS

Client Tier

Oracle 10g Database

Service Tier

JSP,

Servlets

Web Container

Web Server

Presentation Tier

**RESTAURANT MANAGEMENT SYSTEM Web Portal Architecture**

## **Description**

The target application has the following layers:

* HTML pages are built using HTML editor and Dreamweaver acting as the presentation layer and will be developed by HTML tags, JavaScript and style sheets.
* **HTML**-Hyper Text Markup Language (HTML) is the main [markup language](http://en.wikipedia.org/wiki/Markup_language) for [web pages](http://en.wikipedia.org/wiki/Web_page). HTML elements are the basic building-blocks of web pages.
* **JavaScript**-A scripting language developed by Netscape and used to create interactive Web sites.
* **Style Sheets**-It is used for describing the [presentation semantics](http://en.wikipedia.org/wiki/Presentation_semantics) (the look and formatting) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language).
* Business Logic Layer will be developed by using Servlets and Java Server Page (JSP)s.
* **JSP**-Java Server Page (JSP) is a technology for controlling the content or appearance of Web pages through the use of [servlet](http://searchsoa.techtarget.com/definition/servlet)s, small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it.
* **Servlet-** A Servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP( the HyperText Transfer Protocol).
* **MySQL**- It is MySQL [grid computing](http://searchdatacenter.techtarget.com/definition/grid-computing) product group including (among other things) a database management system ([DBMS](http://searchsqlserver.techtarget.com/definition/database-management-system)) and an [application server](http://searchsqlserver.techtarget.com/definition/application-server).

**Client Tier**

* Client tier represents Web browser.
* The client tier makes requests to the Web server.
* The Web server, who will be serving the request by either returning static content if it is present in the Web server or forwards the request to either Servlet or JSP in the application server for either static or dynamic content.

**Presentation Tier**

* Presentation tier encapsulates the presentation logic required to serve clients.
* JSP and Servlets which are the web container elements forms the presentation layer.
* A Servlet or JSP in the presentation tier intercepts client requests, manages logons, sessions, accesses the business services, and finally constructs a response, which gets delivered to client.

**Service Tier**

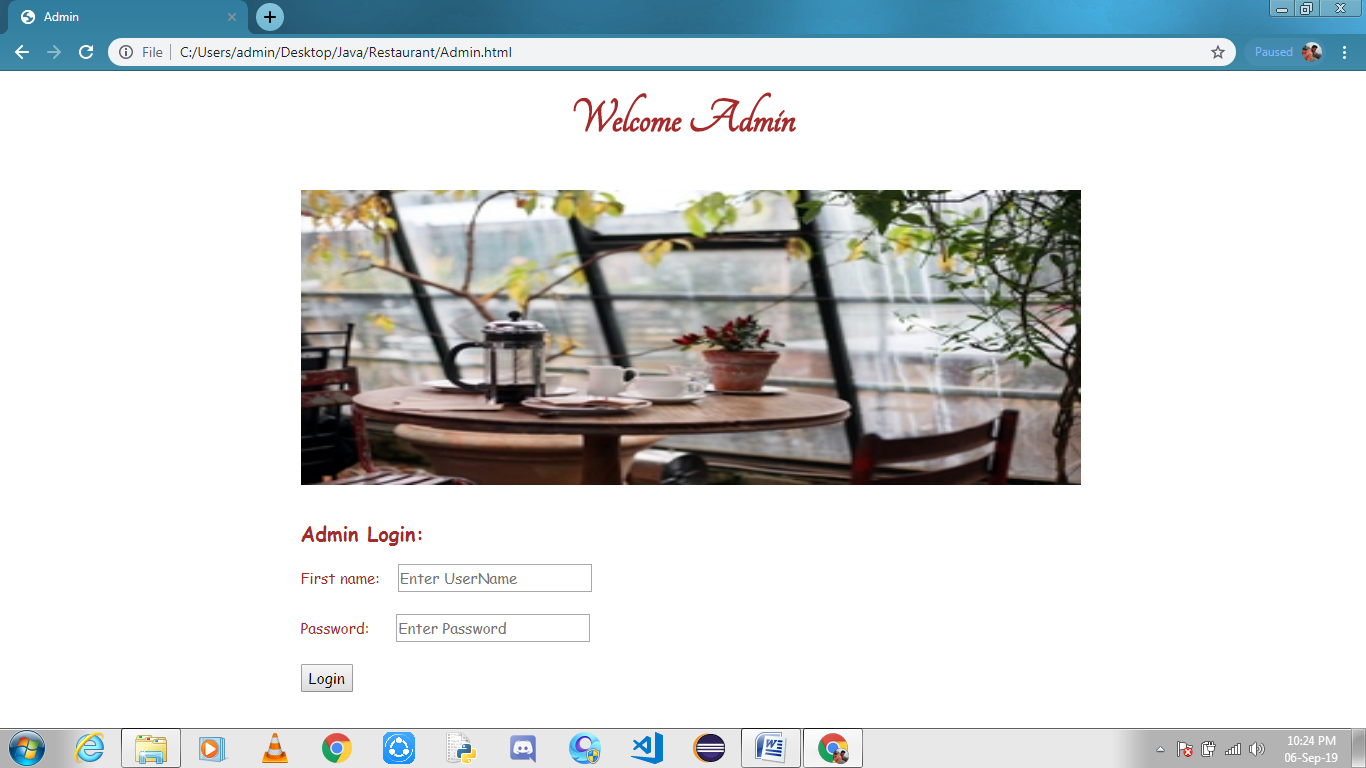
* This tier is the external resource such as a database.
* It is responsible for storing the data.
* This tier is also known as Data Tier or EIS (Enterprise Information System) Tier.
* It uses Java Beans classes to access Oracle 10g database.
* The data access will be done through Java Database Connectivity classes which enable the Oracle database to be connected to the code and operations may be performed on it.

**USE CASE :-**

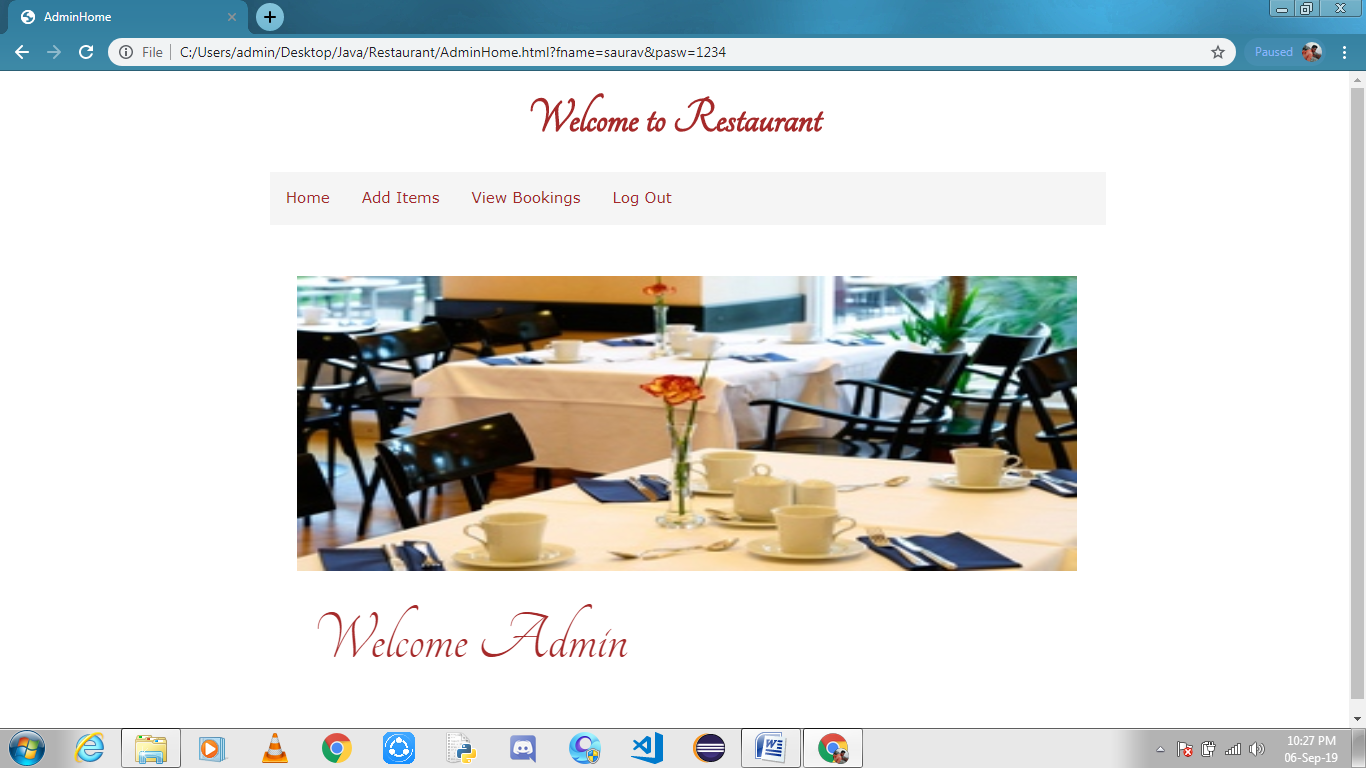
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | | | **Description** | | **Created By** | **Date** |
| ***1.0*** | | | ***First version of the system*** | | ***Deepak Kumar,***  ***Kumari Shikha,***  ***Abhishek Mittal,***  ***Shourabh Rathi*** |  |
| **General Description** | | | | | | |
| *A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.*  *A use case (or set of use cases) has these characteristics:*   * *Organizes functional requirements* * *Models the goals of system/actor (user) interactions* * *Records paths (called scenarios) from trigger events to goals* * *Describes one main flow of events (also called a basic course of action), and possibly other ones, called exceptional flows of events (also called alternate courses of action)* * *Is multi-level, so that one use case can use the functionality of another one.*   *Use cases can be employed during several stages of software development, such as planning system requirements, validating design, testing software, and creating an outline for online help and user manuals.* | | | | | | |
| **Actors *<an entity that can interact with a system, invoking some behavior>*** | | | | | | |
| 1. Administrator 2. End User */ Customer* | | | | | | |
| **Preconditions *<the state(s) the system can be in before this use case starts>*** | | | | | | |
| 1. The System is functioning properly 2. The System should be connected to the Internet 3. The System should have access to the database 4. The Actor should have knowledge of working on internet. 5. The Actor must have a valid E-mail id. | | | | | | |
| **Use Case Diagram <*a* *specific sequence of actions and interactions between actors and the system being discussed*>** | | | | | | |
| **Restaurant Management System (Major) Use Case Diagram** | | | | | | |
| **Basic flow of events: <*a* *specific sequence of actions and interactions between actors and the system being discussed (happy path Or basic course of action)*>** | | | | | | |
| **Line** | | **System Actor Action** | | **System Response** | | |
|  | | Administrator Manages (Deletes Food items, Modifies food Details) | | Customer | | |
|  | | Administrator Updates | | New food Items, Price of foods | | |
|  | | Administrator Uploads | | Details of special items | | |
|  | |  | |  | | |
| **Exceptional flow of events: *<an alternate sequence of actions and interactions between the actors and the system (alternate course of action)>*** | | | | | | |
| 1 | Administrator Updates for Customers. | | | New arrivals, Todays specials | | |
| 2 | Administrator manages | | | Customers | | |
| 3 | Administrator manages (moderate’s comments) | | | Comment Customers on food. | | |
| **Post Conditions <*the state(s) the system can be in when this use case ends>*** | | | | | | |
| The logout operation must enable the logout of the user from the current session.  The feedback provided by the user must be entertained in the system if applicable.  The maintenance of the system must be done regularly to keep the system running.  The system must be monitored regularly in order to avoid slow responsiveness and working of the system. | | | | | | |
| **System Messages *<all system generated messages that the system will prompt for the user>*** | | | | | | |
| 1 | Login-Login successful or Login Failed: Re-enter your username and/or password. | | | | | |
| 2 | Search-Search Results not found. Please modify your search. | | | | | |
| 3  4  5 | Register-Successfully registered or you are already a member.  Upload-Successfully uploaded or failed to upload or slow internet connection.  Download-Successfully downloaded or failed to download due to slow internet connection. | | | | | |
|  | | | | | | |
| **Appendix *<List of any supporting documents attached>*** | | | | | | |
| 1. <http://searchsoftwarequality.techtarget.com/definition/use-case> 2. <http://en.wikipedia.org/wiki/Use_case> 3. Fundamentals of Software Engineering – Rajib Mall | | | | | | |

**ER DIAGRAM:-**

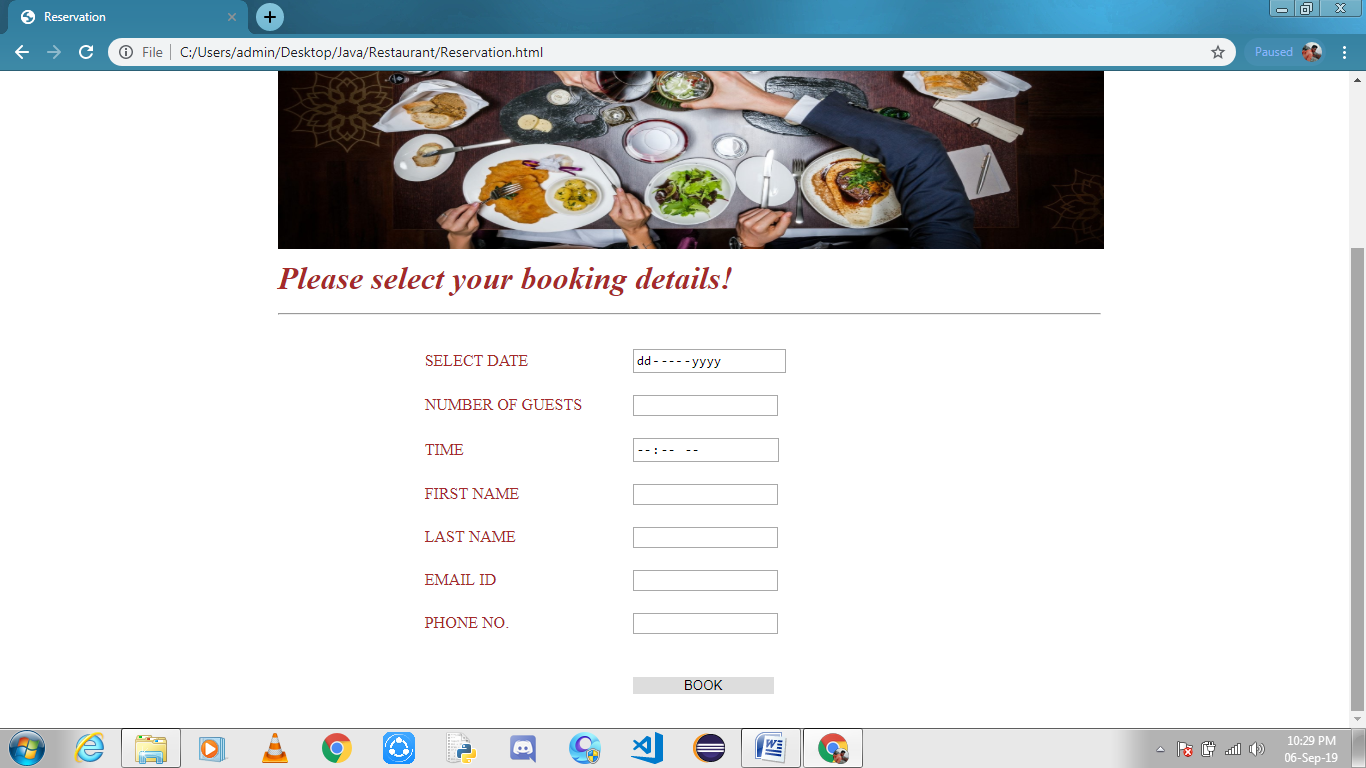
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Version** | | | **Description** | | **Created By** | **Date** | |
| ***1.0*** | | | ***First Version*** | | ***Deepak Kumar,***  ***Kumari Shikha, Abhishek Mittal, Shourabh Rathi*** | ***07-09-2019*** | |
| **General Description** | | | | | | | |
| *The ER Diagram designed here gives us the basic idea about the functionality of the RESTAURANT MANAGEMENT SYSTEM, the idea provided helps us in understanding the relations between the tables and what all attributes are needed their as the information, what will be managed by the administrator, what is the functionality of the system and the modules.* | | | | | | | |
| **ER Diagram** | | | | | | | |
| **RESTAURANT MANAGEMENT SYSTEM**  **1.** **ADMIN**  1.1ADMIN Manages Bookings and Add Items    download      2**. CUSTOMER**  shilpa1 | | | | | | |
| **Basic flow of events: <*a* *specific sequence of actions and interactions between actors and the system being discussed (happy path Or basic course of action)*>** | | | | | | | |
|  | | | | | | | |
| **Line** | | **System Actor Action** | | **System Response** | | | |
|  | | Administrator Manages(Add Items, Confirm Bookings) | | Item Added, Booking Confirmed | | | |
|  | | Customer | | See Menus, Book tables | | | |
|  | |  | |  | | | |
| **Post Conditions <*the state(s) the system can be in when this use case ends>*** | | | | | | | |
| 1. Login-You are successfully logged in to your account page. 2. Login-Home page of RESTAURANT MANAGEMENT SYSTEM (login state) of the (user, administrator, anonymous user). 3. Payment-The payment received by the administrator.   4 Logout-The Admin of the Website is in logout state or out of his/her home page. | | | | | | | |
| **System Messages *<all system generated messages that the system will prompt for the user>*** | | | | | | | |
| 1 | Login-Login successful or Login Failed: Re-enter your username and password. | | | | | | |
| 2 | Items-Surf items. | | | | | | |
| 3 | Registration-Successfully registered a table.. | | | | | | |
| 4 | Confirmation-Confirmed if table available otherwise rejected by Admin.. | | | | | | |
| 5 | Add Items-Successfully Added items by Admin. | | | | | | |
|  |  | | | | | | |
| **User Interface Screens: *<a prototype screen for this use case to give proof of concept or detailed level depending on the importance of the screen design to the user>*** | | | | | | | |
| **Screen ID: RESTAURANT MANAGEMENT SYSTEM INDEX PAGE** | | | | | | | |
|  | | | | | | | |
| **Screen ID: RESTAURANT MANAGEMENT SYSTEM – ADMIN LOGIN PAGE** | | | | | | | |



**Screen ID: RESTAURANT MANAGEMENT SYSTEM ADMINHOME PAGE**

****

**Screen ID: RESTAURANT MANAGEMENT SYSTEM BOOKING PAGE**

****