**Restaurant reservation system**

*A*

***Major Project Report***

*Submitted*

*In partial fulfillment*

*For the award of the Degree of*

## BACHELOR OF TECHNOLOGY

***In Department of Computer science and Engineering***

**SUBMITTED BY: GUIDED BY: SUBMITTE TO:**

Suhani Jain(18etccs084)

Chahat lodha (18etccs022) Aditya Maheshwari Head of department



### Department of Computer Science and Engineering TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY

RAJASTHAN TECHNICAL UNIVERSITY

**May 2022**



### TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY



## CERTIFICATE

This is to certify that this project report **“Restaurant reservation system”** is the confide work of **“Chahat Lodha, Suhani Jain”** who have carried out the project work under my supervision. I approve this project for submission of the Bachelor of Technology in the **Department of Computer Science and Engineering, Techno India NJR Institute of Technology**, affiliated to Rajasthan Technical University, Kota.

##### Mr. Aditya Maheshwari

**Head of department Assistant professor**

Department of Computer Science Department of Computer Science



## ABSTRACT

### Purpose

* 1. ***Introduction***

The Restaurant Reservation System is a simple project developed using PHP, JavaScript, and CSS. The project contains two roles one is registered user and the other is guest users. The guest users can go through the homepage which contains various sections. In order to make the reservation for a table, the user has to create an account and sign in or log in. This project makes a convenient way for customers to reserve tables online.

* 1. ***Scope***
* This can be further used in the Hospitality Industry for easy management of the reservations for the rooms and services.

### Document overview

The remainder of this document is 8 chapters, the first providing introduction of the project. It lists all the functions performed by the system. The second chapter consists of software requirements specification. The third chapter provides details about system analysis and design. The fourth chapter gives data dictionary information. The fifth chapter consists of snapshots of the complete project. The sixth chapter gives testing for the project. The seventh chapter tells about the conclusion and future enhancements of the project. The final chapter concerns with the bibliography.

## ACKNOWLEDGEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed guide **Mr. Aditya Maheshwari (Assistant Professor), TINJRIT** for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I would like to express my sincere thanks to **Faculties of CSE Department TINJRIT** for giving me this opportunity to undertake this project.

I also wish to express my indebtedness to my parents as well as my family member whose blessings and support always helped me to face the challenges ahead.

At the end I would like to express my sincere thanks to all my friends and others

who helped me directly or indirectly during this project work.

**Place : Udaipur Suhani Jain (18etccs084)**

**Date : Chahat Lodha (18etccs022)**



## TABLE OF CONTENTS

**Table of Contents**

[ABSTRACT iii](#_Toc104815118)

[ACKNOWLEDGEMENT iv](#_Toc104815121)

[TABLE OF CONTENTS v](#_Toc104815122)

LIST OF TABLES …………………………………………………………………………………………………….vii

LIST OF FIGURES …………………………………………………………………………………………………..viii

CHAPTER – 1 Introduction .......................................................................................................................................1-5

Introduction ......................................................................................................................................................................2

1.1. Purpose .....................................................................................................................................................................2 1.2. Overall Description ..................................................................................................................................................4

**CHAPTER – 2 Software Requirement Specification ..........................................................................................6-20**

2.1. Purpose................................................................................................................................................……………7

2.2. Overall Description ...............................................................................................................................................9

2.2.1. Functional Requirements definitions ..............................................................................................…………...9

2.2.2. Use Cases ...........................................................................................................................................................9

2.2.3. Non – Functional Requirements ......................................................................................................................14

2.3. Requirement Specification ..................................................................................................................................15

2.3.1. External interface specifications ......................................................................................................................15

2.3.2. Functional Requirements .................................................................................................................................15

2.4. Hardware Specification ......................................................................................................................................20

2.5. Software Specification .......................................................................................................................................21

2.6. Hardware and Software Requirements in detail ................................................................................................21

**CHAPTER – 3 System Analysis and Design ...................................................................................................22-30**

3.1. Study & Weaknesses of current system ............................................................................................................23

3.2 Requirements of new system ..................................................................................................................24

3.3. Feasibility Study ................................................................................................................................... 24

3.4. Features of new System ........................................................................................................................25

3.5. Data flow diagram ................................................................................................................................26

3.6. UML Modeling......................................................................................................................................28

3.6.1. Activity Diagram ...............................................................................................................................28

3.6.2. Context Diagram ...............................................................................................................................30

**CHAPTER – 4 Data Dictionary ..........................................................................................................31-36**



4.1. Table Details ........................................................................................................................................32

4.2. E-R Diagram ........................................................................................................................................36

**CHAPTER - 5 Screen Shots ...............................................................................................................37-48**

5.1 User Side ……………………………………………………………………………………………..38

**CHAPTER – 6 Testing .......................................................................................................................49-55**

**CHAPTER – 7 Conclusion and Future Enhancements ..................................................................56-58**

7.1. Limitations .........................................................................................................................................57

7.2. Future Enhancements ........................................................................................................................58

7.3. Conclusion .........................................................................................................................................58

**CHAPTER – 8 Bibliography .................................................................................................................59**

**List of Tables**



|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
| Table 2.1 | Glossary | 7 |
|  | **Use Cases** |  |
| Table 2.2 | Access Home page | 15 |
| Table 2.3 | User login or signup | 15-16 |
| Table 2.4 | Manage Reservations | 16-18 |
|  |  |  |
|  | **Database Design** |  |
| Table 4.1 | Admin Login | 32 |
| Table 4.2 | Category | 33 |
| Table 4.3 | Request Details | 33 |
| Table 4.4 | Contact Us Details | 33 |
| Table 4.5 | Property Details | 34 |
| Table 4.6 | Testimonials | 34 |
| Table 4.7 | Financial Applications | 34 |
| Table 4.8 | Employment Applications | 35 |
| Table 6.1 | Test report with test data | 54-55 |



**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
|  | **Use Cases** |  |
| Figure 2.2.2a | Access Home page | 10 |
| Figure 2.2.2b | Manage Content | 11 |
| Figure 2.2.2c | Admin Login | 12 |
| Figure 2.2.2d | User selects, filter, view, rent or buy property | 13 |
|  | **Data Flow Diagrams** |  |
| Figure 3.1 | USER DFD Level 2 | 26 |
| Figure 3.2 | Admin (DFD Level 2) | 27 |
|  | **UML Modelling** |  |
|  | *Activity Diagrams* |  |
| Figure 3.10 | Activity Diagram for Admin | 26 |
| Figure 3.11 | Activity Diagram for User | 27 |
| Figure 3.12 | Context Diagram | 30 |
| Figure 4.1 | E-R Diagram | 36 |



***CHAPTER – I INTRODUCTION***



**Introduction**

##### Purpose

###### Introduction

The Restaurant Reservation System is a simple project developed using PHP, JavaScript, and CSS. The project contains two roles one is registered user and the other is guest users. The guest users can go through the homepage which contains various sections. In order to make the reservation for a table, the user has to create an account and sign in or log in. This project makes a convenient way for customers to reserve tables online.

###### Scope

This can be further used in the Hospitality Industry for easy management of the reservations for the rooms and services.

###### References

* <https://www.w3schools.com/html>
* <https://www.javatpoint.com/javascript-tutorial>
* <https://github.com>

###### Document overview

The remainder of this document is 8 chapters, the first providing introduction of the project. It lists all the functions performed by the system. The second chapter consists of software requirements specification. The third chapter provides details about system analysis and design. The fourth chapter gives data dictionary information. The fifth chapter consists of snapshots of the complete project. The sixth chapter gives testing for the project. The seventh chapter tells



about the conclusion and future enhancements of the project. The final chapter concerns with the bibliography.

This document is meant for describing all the features and procedures that were followed while developing the system.

This document specially mentions the details of the project how it was developed, the primary requirement, as well as various features and functionalities of the project and the procedures followed in achieving these objectives.

A restaurant reservations system helps you manage all your bookings digitally so you can reduce those dreaded no shows and ensure you’re maximizing all your available tables.

Beyond getting diners in the door, a restaurant reservation system also offers many other benefits for your venue such as direct communication with guests, capacity tracking, digital waitlists, and more. So whether you’re struggling with no-shows, trying to fill tables on slow nights, or wondering why new customers aren’t turning into regulars, a restaurant reservation management system can help.



##### Overall description

* We have developed a website for Restaurant Reservations operations in order to make it more accessible and much organised which solves a number of problems faced by consumers.
* This website provides easy way to find tables and make reservations without much hustle.

###### Functional requirements definitions

Functional Requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Nonfunctional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately.

###### Use cases

This system will be used in two User Modules which are Administrator and Registered User. As all of these have different requirements the modules are designed to meet their needs and avoid any type of confusion. The Uses of all two User Modules have been described below.

1. User can do the following functions in the Administrator Module

* Add/Delete reservations
* Edit Schedule
* Edit/View no. of Tables.



1. User can do the following functions in the Registered User Module
   * Can visit the site.
   * Add a new reservation.
   * View reservations.
     1. ***User characteristics***

The user should be familiar with the Internet.

The user should be familiar with the property related terminologies.

#### Constraints

Limited to HTTP/HTTPS.

Real-life payment options not available.  
No multilingual support



# CHAPTER – II

***SOFTWARE REQUIREMENT SPECIFICATION***



### Software Requirement Specification

##### Purpose

###### Introduction

This Software Requirements Specification provides a complete description of all the functions and specifications of the website.

###### Scope

* + - * This can be used in Hospitality Industry.
      * Can be used anywhere any time as it is a web based application.
      * No restriction that the manager of the restaurant has to be present.

###### Glossary

**Table 2.1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Term** | **Definition** | | | | |
| Admin | The only user who has the permission to manage reservations, manage tables, manage schedule. | | | | |
| Entry | User stored in the Database | | | | |
| Html | Hyper text markup language | | | | |
|  |  |  |  |  |  |



|  |  |
| --- | --- |
| QA | Quality assurance |
| SCMP | Software Configuration Management Plan |
| SDD | Software Design Document |
| SQAP | Software Quality Assurance Plan |
| SRS | Software Requirements Specification |
| Web Site | A place on the world wide web |

* + 1. ***References***

###### Document overview

The remainder of this document is two chapters, the first providing a full description of the project for the owners of the Restaurant Reservation System. It lists all the functions performed by the system. The final chapter concerns details of each of the system functions and actions in full for the software developers’ assistance. These two sections are cross-referenced by topic; to increase understanding by both groups involved.



##### Overall description

* We have developed a website for Restaurant Reservations operations in order to make it more accessible and much organised which solves a number of problems faced by consumers.
* This website provides easy way to find tables and make reservations without much hustle.

###### Functional requirements definitions

Functional Requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Nonfunctional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately.

###### Use cases

This system will be used in four User Modules which are Administrator, User. As all of these have different requirements the modules are designed to meet their needs and avoid any type of confusion. The Uses of all four User Modules have been described below.

This system will be used in two User Modules which are Administrator and Registered User. As all of these have different requirements the modules are designed to meet their needs and avoid any type of confusion. The Uses of all two User Modules have been described below.

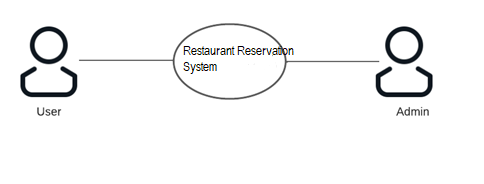
1. User can do the following functions in the Administrator Module

* Add/Delete reservations
* Edit Schedule
* Edit/View no. of Tables.



1. User can do the following functions in the Registered User Module
   1. Can visit the site.
   2. Add a new reservation.
   3. View reservations.

2.2.2a Use Case: Access Home Page



##### Fig. 2.2.2a Access Home Page

Brief Description:

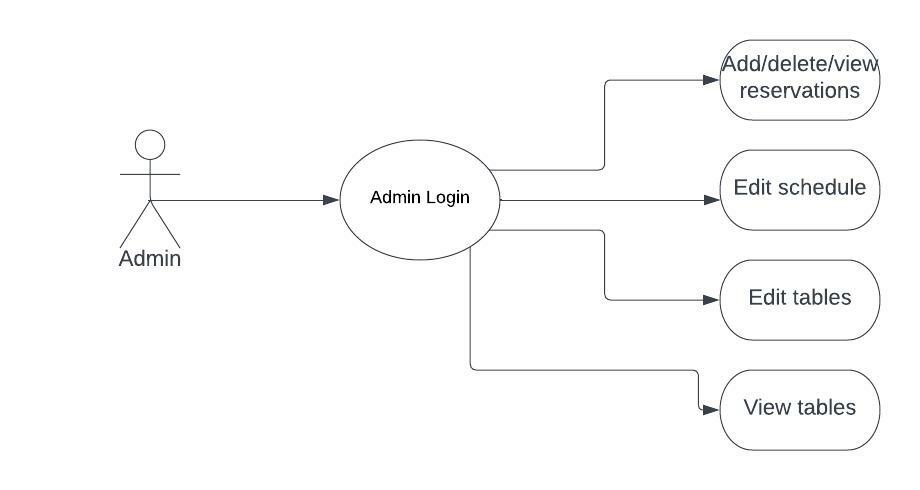
User uses the website to access the home page

Initial step-by-step description:

For this use case to be initiated, the user (user or admin) can use the website.

1. The user connects to the website using a web browser.
2. The user clicks on the logo and gets redirected to the home page.

2.2.2b. Use Case: manage reservations, schedules, tables



##### Fig. 2.2.2b Manage reservations, schedules, tables

Brief Description:

Admin should be on the admin site by login with administrator credentials.

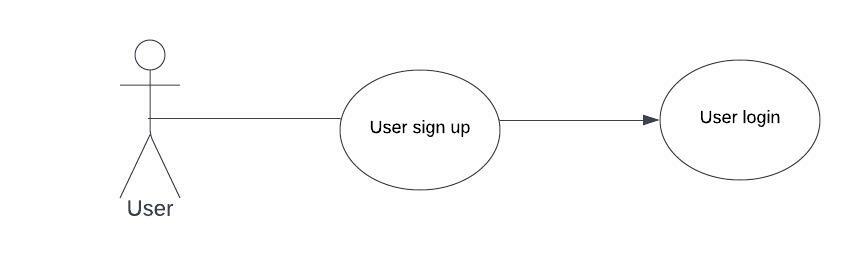
The admin can view, delete, add reservations.

The admin can manage the schedule.

The admin can manage tables.



2.2.2c. Use Case: User Login



##### Fig. 2.2.2.c User Login

Brief Description:

The user can sign up and then login to the system to make a reservation.

Initial step-by-step description:

1. The User selects the “Signup” link.
2. The user enters the UserID, email and Password to sign up.
3. Then the user can login using the email and password.
4. The user can make reservations



###### Non-functional requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within.

The web site must be compatible with Internet Explorer web browser.



##### Requirement specifications

###### External interface specifications

None

###### Functional Requirements

**Table 2.2 Access Home Page**

|  |  |
| --- | --- |
| **Use Case Name:** | Access Home Page |
| **Priority** | Essential |
| **Trigger** | Menu selection |
| **Precondition** | User is on the home page. |
| **Basic Path** | 1. The user connects to the system using a web browser. 2. The user selects the Home link on the website home page. 3. The system passes the user to the   website Home Page. |
| **Alternate Path** | N/A |
| **Postcondition** | The User is on the Home Page |
| **Exception Path** | If there is a connection failure the website returns to the wait state |
| **Other** |  |

**Table 2.3 User Login or Signup**

|  |  |
| --- | --- |
| **Use Case Name:** | User Login or Signup |
| **Priority** | Essential |
| **Trigger** | Selects |
| **Precondition** | The User is on the Home Page |



|  |  |
| --- | --- |
| **Basic Path** | 1. The User selects the “Login” link. 2. The user enters the ID and Password to login. 3. If the user is already registered and credentials are correct, he successfully gets logged in. 4. Else it displays an error message to signup. |
| **Alternate Path** |  |
| **Postcondition** | The user is on the home page. |
| **Exception Path** | If the connection is terminated before the form is submitted, the fields are all cleared and the website is returned to the wait state. |
| **Other** |  |

**Table 2.4 Manage Reservations**

|  |  |
| --- | --- |
| **Use Case Name:** | Manage **Reservations** |
| **Priority** | Essential |



|  |  |
| --- | --- |
| **Trigger** | Menu selection |
| **Precondition** | The user must be logged in and on any of the pages . |
| **Basic Path** | 1. The user selects the New reservation button and fill the form to make a reservation. 2. The user then can view their reservation under ‘View reservation’ page. |
| **Alternate Path** | N/A |
| **Postcondition** | A record is created or updated in the related Table of the Database. |
| **Exception Path** | 1. If the connection is terminated before the form is submitted, the fields are cleared and the website is returned to the wait state. 2. If the connection is terminated after the form is submitted, but before the Admin   is returned to the Admin Home Page, the |
|  | record is created in the Table of the  Database. |
| **Other** |  |



**Table 2.5 Admin can manage schedule, tables, and reservations.**

|  |  |
| --- | --- |
| **Use Case Name:** | **Admin can manage schedule, tables, and reservations.** |
| **Priority** | Essential |
| **Trigger** | Menu selection |
| **Precondition** | Admin is on the home page. |
| **Basic Path** | * + - 1. The admin can make a new reservation by clicking on ‘New reservation’       2. The admin can view the reservations under ‘View reservation’       3. The admin can delete the reservations       4. The admin can edit the schedule.       5. The admin can view and edit the tables. |



|  |  |
| --- | --- |
| **Alternate Path** |  |
| **Postcondition** | The related data is updated in the database |
| **Exception Path** | 1. If the connection is terminated before the form is submitted, the fields are cleared and the website is returned to the wait  state. |
| **Other** |  |

* 1. **Hardware Specification Client Side:**
* Internet explorer or Google chrome or Fire fox or safari
* Processor: Intel i3 or above.
* RAM : 2 GB
* Hard Disk : 512GB

##### Server Side:

* Processor: Intel i3 or above.
* RAM : 1 GB
* Disk space : 4GB

##### Software Specification



* Languages
* PHP
* Javascript
* HTML
* CSS
* GITHUB



* Visual Studio Code

* Xampp

##### Data Base Server:

* MySql

##### Hardware and Software Requirements in detail Hardware Requirements:

* Processor: Intel i3 or above.
* Internet explorer or Google chrome or Fire fox or safari
* RAM : 1 GB
* Hard Disk : 80GB
* Disk space : 4GB



# CHAPTER – III

***SYSTEM ANALYSIS AND DESIGN***



### System Analysis and Design

##### Study & Weaknesses of Current System Current System

The current system of restaurant reservation is highly unorganized and very difficult in terms of reserving. In the current system operations and accessibility of reservations are in very poor condition. It is not much user-friendly.

##### Weaknesses in Current System

The current system is as mentioned earlier very complicated and expensive as compared to the new system. It also wastes the precious time of the user/customer.

There is a need of lot of improvement in the current system. As this system is very much complex and not at all transparent. Many errors can be made in this system because of less security and less authorization. People make wrongs reservations and have to suffer with inconvenience when they reach the restaurant.



##### Requirements of New System

* + 1. **User Requirements**

The User requirements for the new system are to make the system fast, flexible, less prone to errors and reduce expenses and save time.

* + - * Time can be saved in finding the right table as before the user have to go find a suitable table in person.
      * Admin can easily manage the schedule of the restaurant for the bookings.

##### Feasibility Study

A key part of the preliminary investigation that reviews anticipated costs and benefits and recommends a course of action based on operational, technical, economic, and time factors. The purpose of the study is to determine if the systems request should proceed further.



##### Data Flow Diagram (DFD)

The DFD (also known as *bubble chart*) is a simple graphical formalism that can be used to represent a system in terms of the input data into the system, various processes carried on these data, and the output data generated by the system.

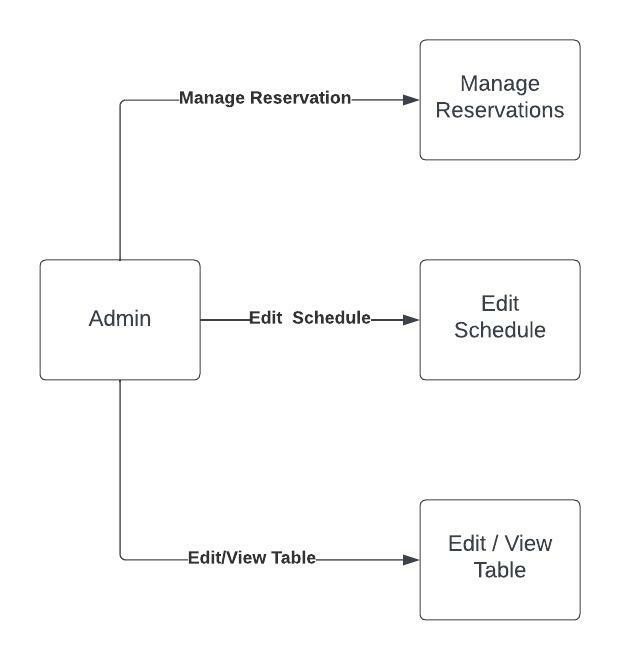
The main reason why the DFD technique is so popular is because the fact that the DFD is a very simple formalism – it is simple to understand and use. A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and the data flow among the functions. Starting with a set of high-level functions that a system performs, a DFD model hierarchy represents various sub-functions.

##### Fig. 3.1 USER DFD Level 2

##### 



**Fig. 3.2 Admin (DFD Level 2)**

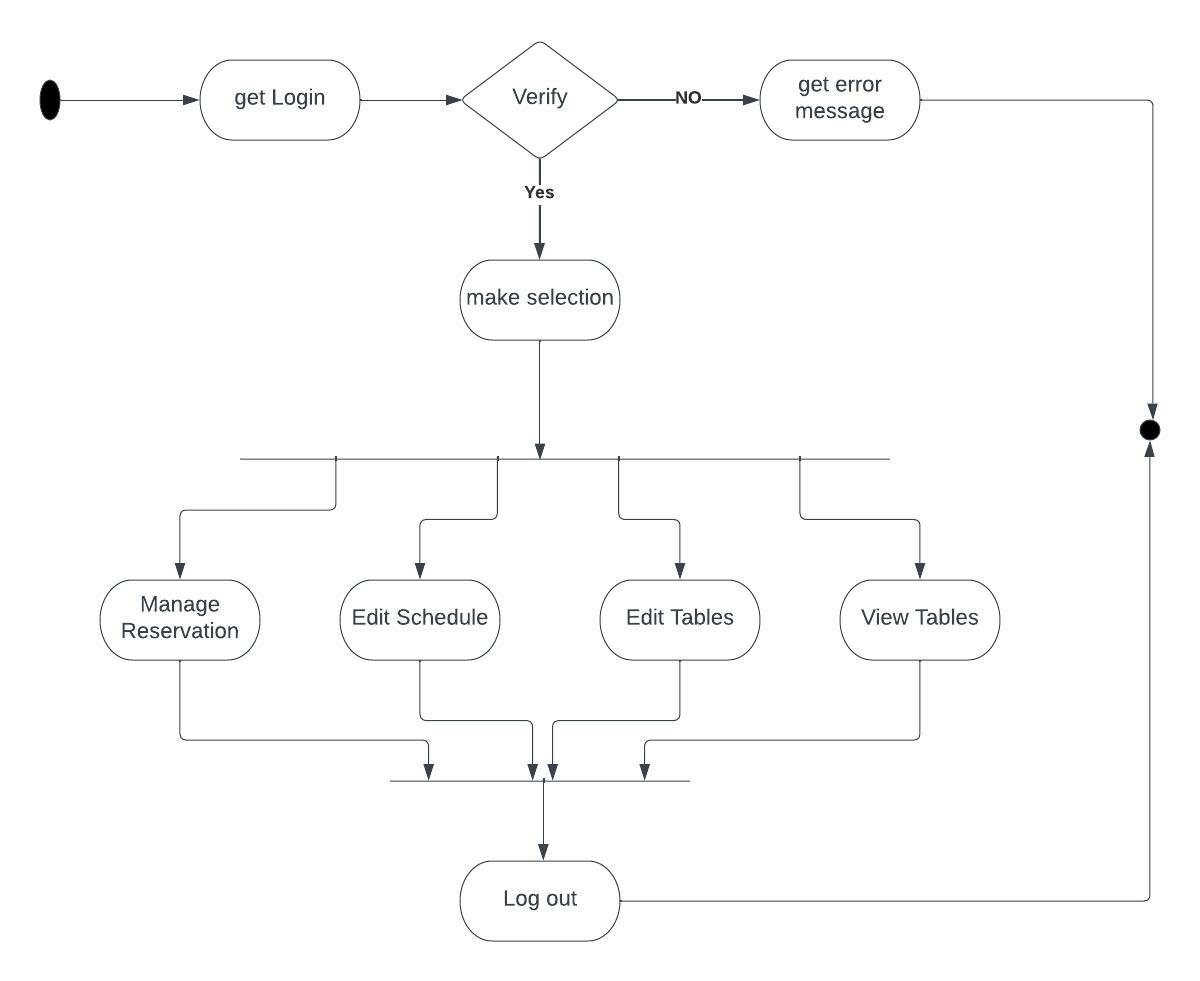




* 1. **UML Modelling**

**3.6.1 Activity Diagram**

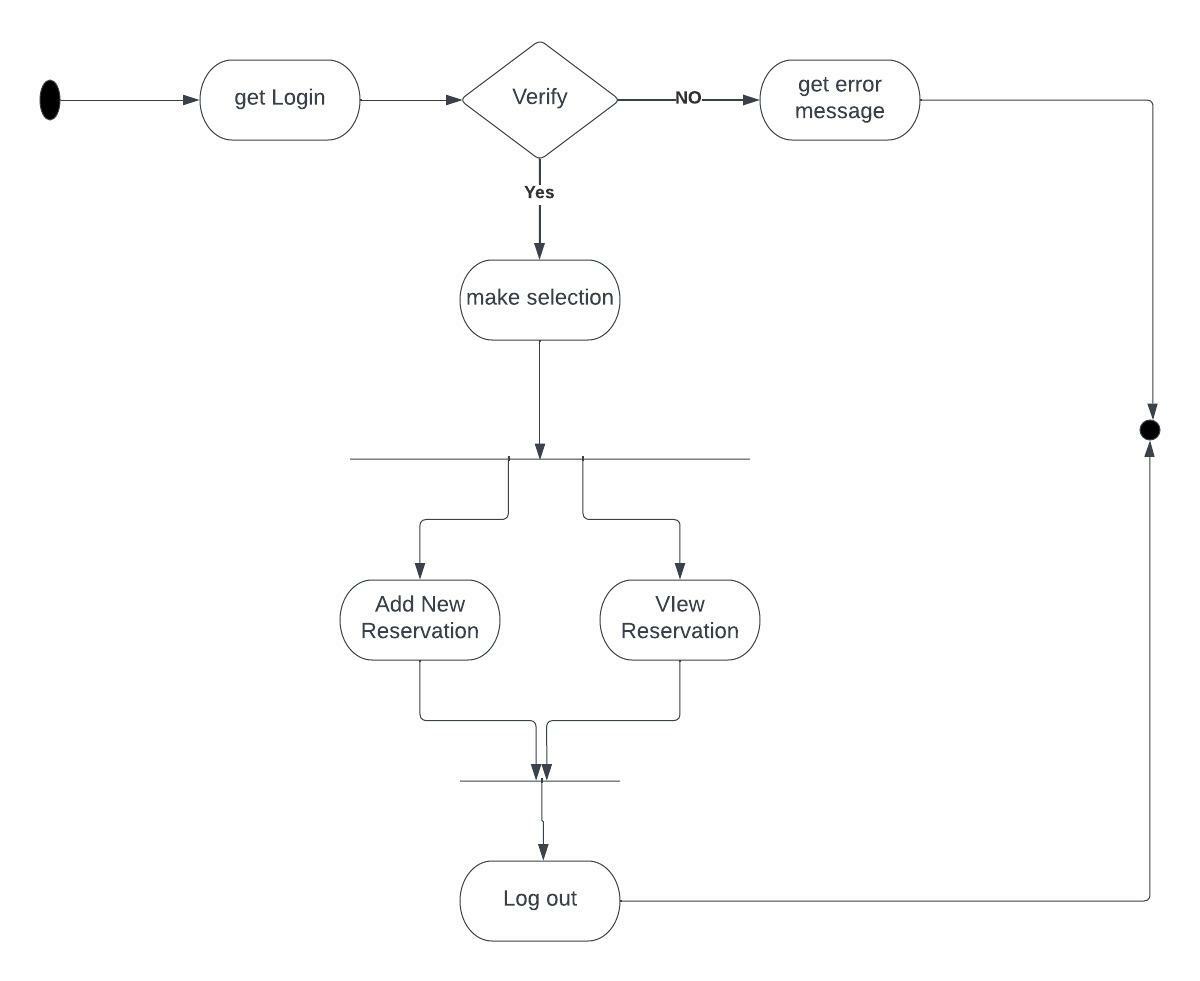
**3.6.1a. Activity Diagram for Admin**

****

**Fig. 3.10 Activity Diagram for Admin**

**3.6.1b. Activity Diagram for User**



****

**Fig. 3.11 Activity Diagram for User**



***CHAPTER – IV DATA DICTIONARY***



**Data Dictionary**

A data dictionary is a catalog-a-repository of the elements in a system. As the name suggests, their elements center on data and the way they are structured to meet user requirements and organization needs. In a data dictionary you will find a list of all the elements composing the data flowing through a system. The major elements are data flows, data stores and processes. The data dictionary stores details and descriptions of these elements.

If analysis want to know characters are in a data item by what other names it is referenced in the system, or where it is referenced in the system, or where it is issued in the system, they should be able to find the answers in issued in the system, they should be able to find the answer in properly developed data dictionary.

The Dictionary contains two types of description for the data following through the system.

##### Data Elements

The most fundamental data is the elements. They are building blocks for all other data in the system. Data elements are also alternatively known as fields, data item or elementary item.

##### Data Structure

A data structure is a set if items that are related to one another and described a components in the system.

### Table Details

##### Table 4.1. User Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Constraints** | **Size** | **Data Type** |
| Username | username  of the user |  |  | tinytext |
| Password | User password |  |  | tinytext |
| User\_id | Unique ID of the user | Primary Key | 11 | integer |
|  |  |  |  |  |
| email | Email of the user |  |  | tinytext |
| Registration date | Date of registration of the user |  |  | datetime |
| Role id | Role of the user |  | 11 | integer |



**Table 4.2. Tables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Constraints** | **Size** | **Data Type** |
| Tables\_id | Unique id for  tables | Primary key | 11 | integer |
| T\_date | date |  |  | date |
| T\_tables | Total tables per date |  | 11 | integer |

**Table 4.3. Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Constraints** | **Size** | **Data Type** |
| Schedule\_id | Id of the schedule | Primary key | 11 | integer |
| date | date |  |  | date |
| Open\_Time | Opening time |  |  | time |
| Close\_Time | Closing time |  |  | time |

**Table 4.4. Role**

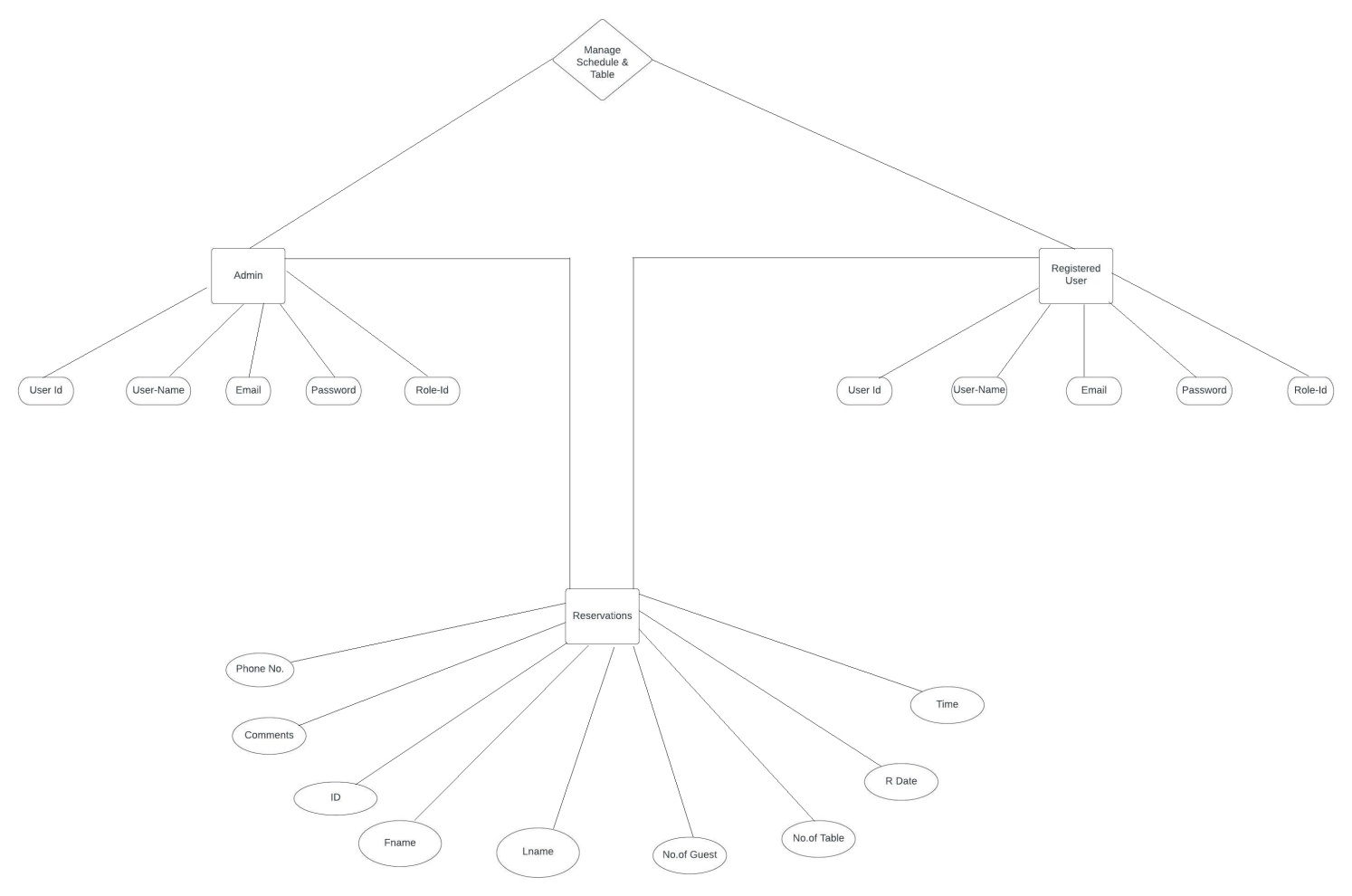
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Constraints** | **Size** | **Data Type** |
| Role\_id | Id of the various roles | Primary Key | 11 | Integer |

**Table 4.5. Property details**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Constraints** | **Size** | **Data Type** |
| Reserve\_id | Unique Id for reservation | Primary Key | 11 | Integer |
| F\_Name | First Name of the user |  | 50 | Varchar |
| L\_name | Last Name of the user |  | 50 | Varchar |
| Num\_guests | No. of guests |  | 11 | Integer |
| Num\_tables | No. of tables |  | 11 | Integer |
| rdate | Date of reservation |  |  | date |
| Time\_zone | Time of reservation |  |  | text |
| telephone | Phone number of the user |  |  | text |
| comment | Comment from the user |  |  | mediumtext |
| Reg\_date | Current date |  |  | timestamp |
| User\_fk | User id of the user |  | 11 | integer |



* 1. **E-R Diagram**

****

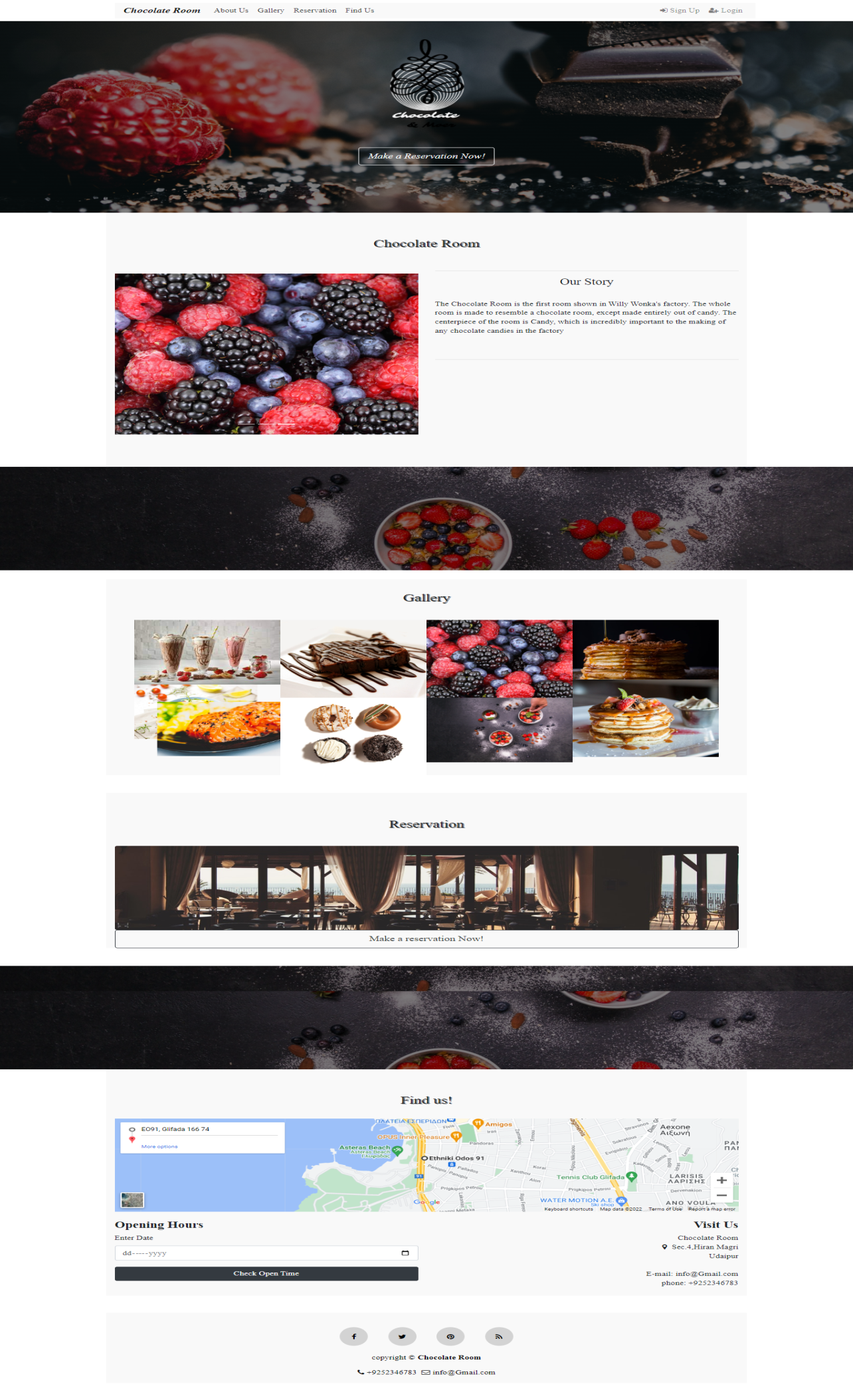
**Fig. 4.1 E-R Diagram**



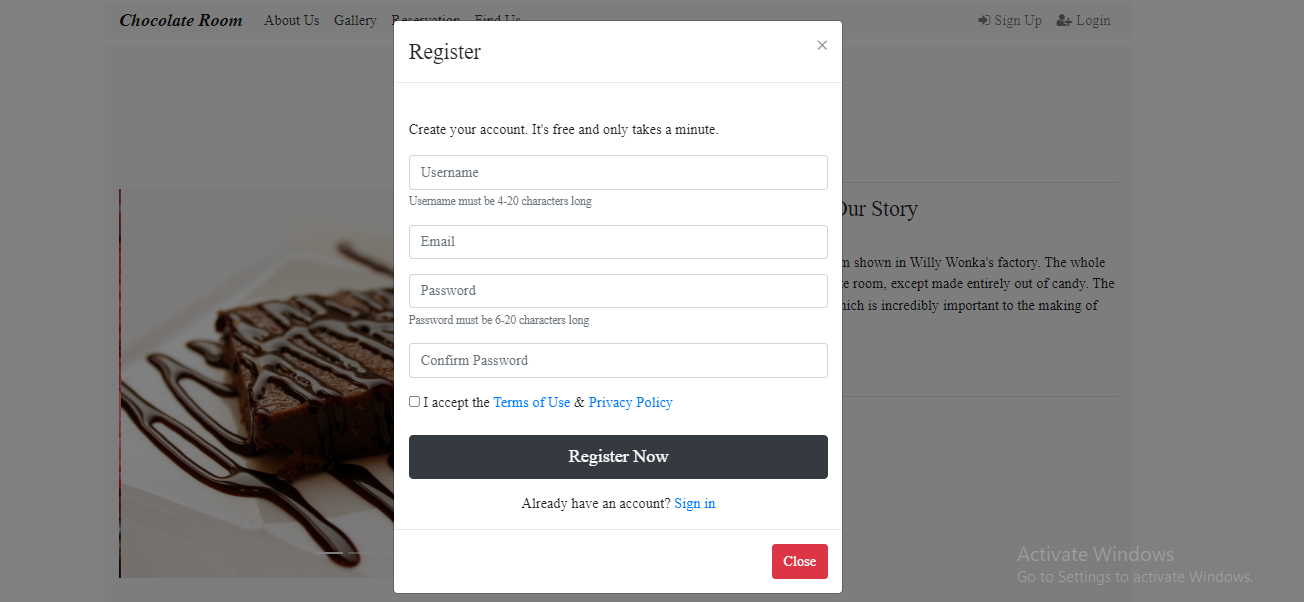
***CHAPTER – V SCREEN SHOTS***



* 1. **User Side**
     1. **Home Page**

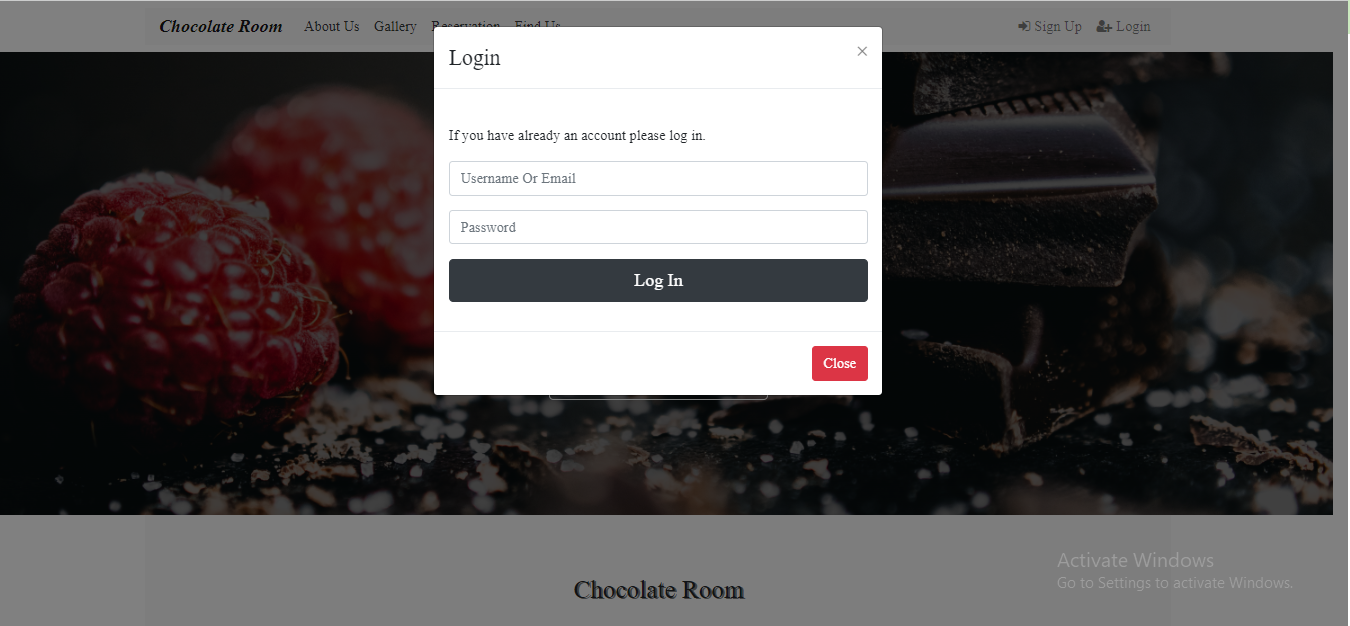
**s**

* + 1. **Registration page**

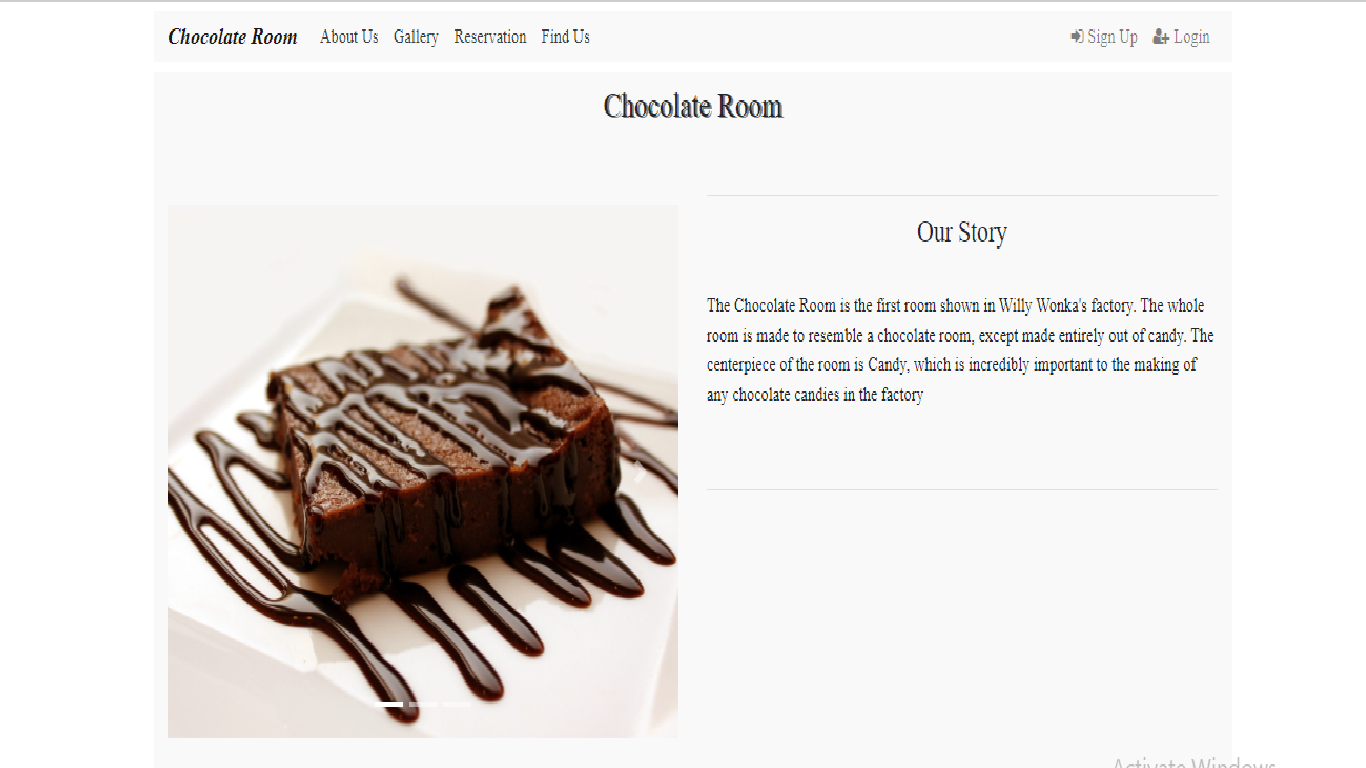




* + 1. **Login page**

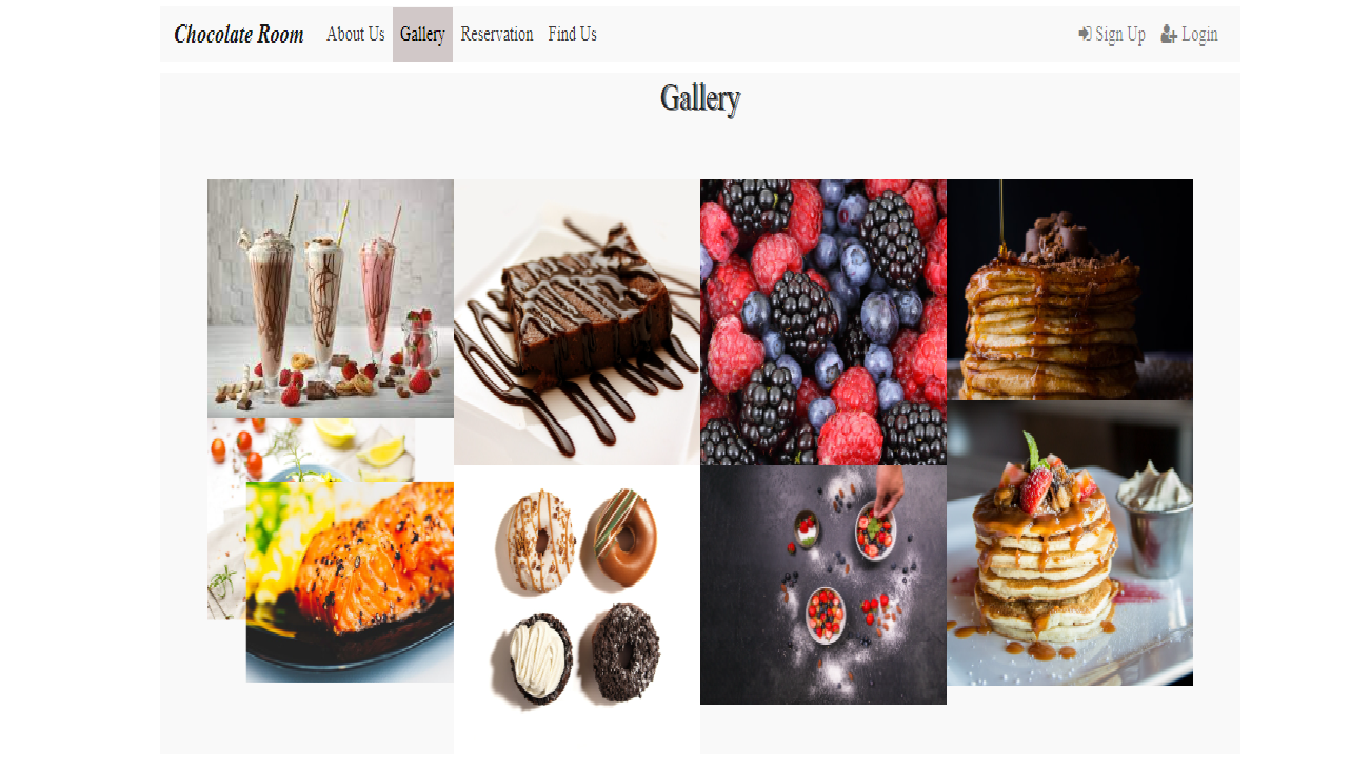
****

* + 1. **About us**

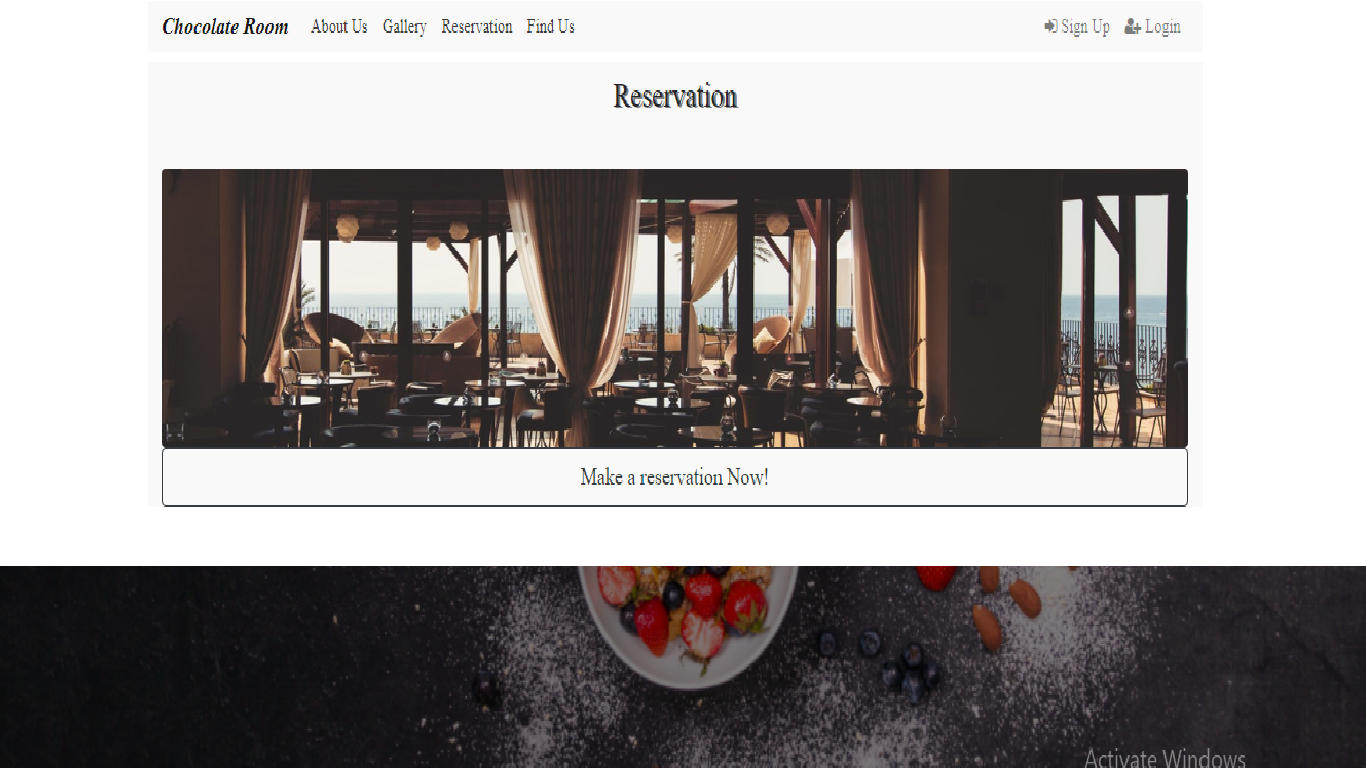
****



**5.1.5 Gallery**

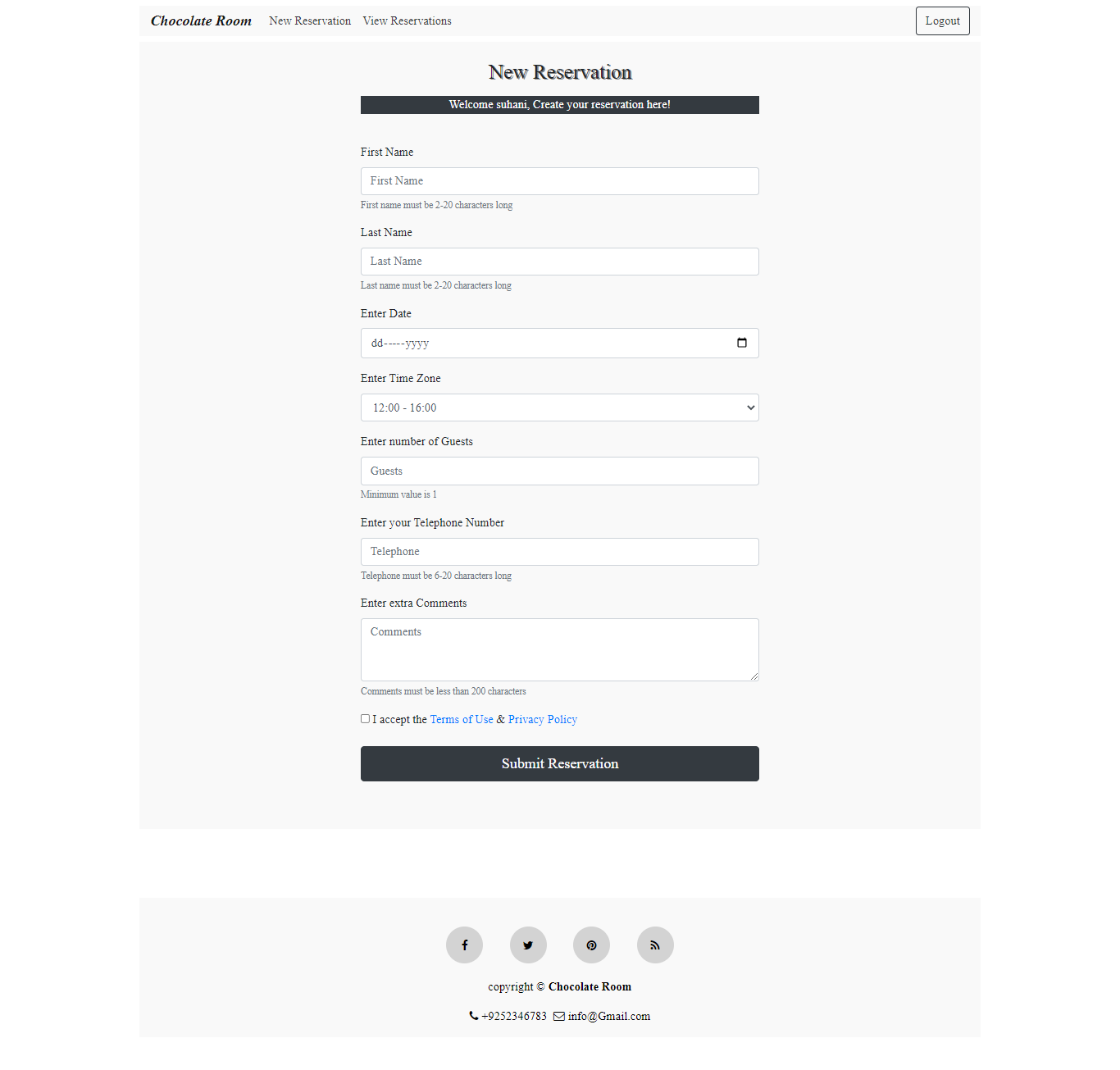
****

**5.1.6 Reservation**

****

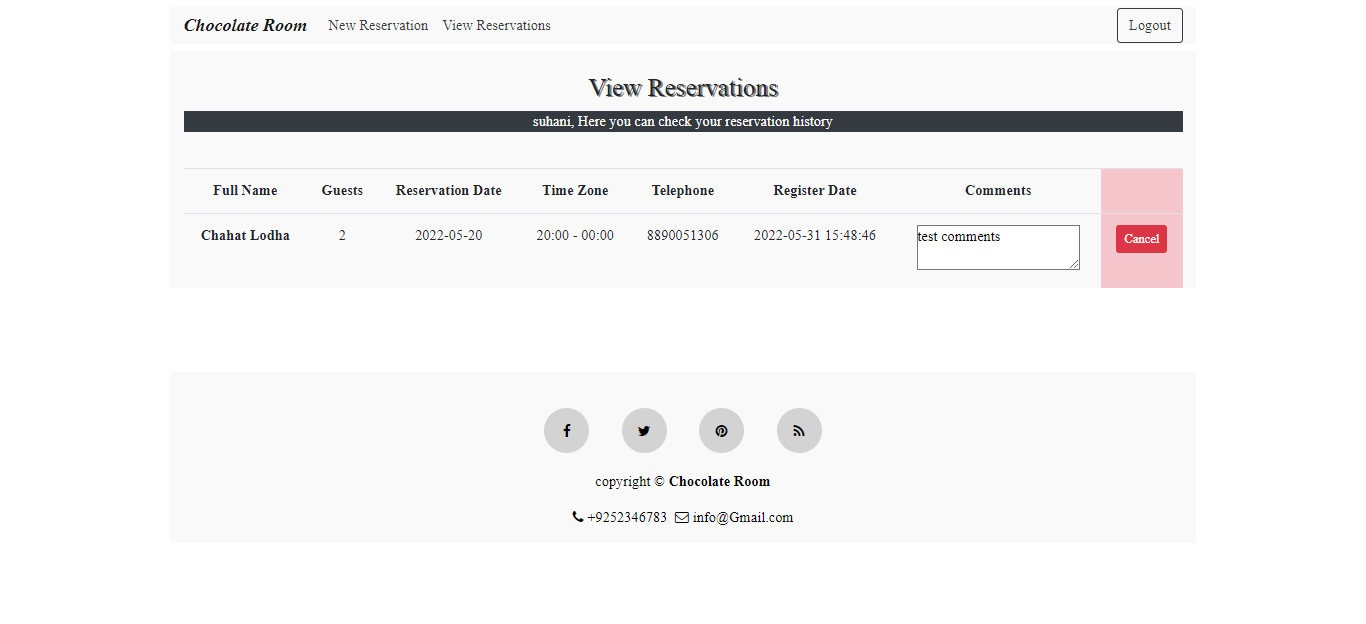


5.1.6.1 New Reservation





**5.6.1.2 View Reservation**

****



***CHAPTER – VI***

***TESTING***



**Testing**

**Testing Methodology**

Companies rely on software more than ever to provide and manage information with strategic and operational importance and to provide key decision support. Rising customer expectations for fault-free, requirements-exact software have increased awareness of the importance of software testing as a critical activity.

We begin the testing process by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. At the end of each testing day, we prepare a summary of completed and failed tests. Applications are not allowed to launch until all identified problems are fixed. A report is prepared at the end of testing to show exactly what was tested and to list the final outcomes.

Our software testing methodology is applied in three distinct phases: unit testing, system testing, and acceptance were testing.

**Unit Testing**: The programmers conduct unit testing during the development phase. Programmers can test their specific functionality individually or with other units. However, unit testing is designed to test small pieces of functionality rather than the system as a whole. This allows the programmers to conduct the first round of testing to eliminate bugs before they reach the testing staff. In unit testing the analyst tests the programs making up a system.

For this reason, unit testing is sometimes called program testing. Unit testing gives stress on the modules independently of one another, to find errors. This helps the tester in detecting errors in coding and logic that are contained within that module alone. The errors resulting from the interaction between modules are initially avoided.

For example, a hotel information system consists of modules to handle reservations; guest checking and checkout; restaurant, room service and miscellaneous charges; convention activities; and accounts receivable billing. For each, it provides the ability to enter, modify or retrieve data and respond to different types of inquiries or print reports. The test cases needed for unit testing should exercise each condition and option.



Unit testing can be performed from the bottom up, starting with smallest and lowest-level modules and proceeding one at a time. For each module in bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embedded within the larger system.

**System Testing*:*** The objective of system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified and to ensure that the computer system and the associated clerical and other procedures work together.

The initial phase of system testing is the responsibility of the analyst who determines what conditions are to be tested, generates test data, produced a schedule of expected results, runs the tests and compares the computer produced results with the expected results with the expected results.

The analyst may also be involved in procedures testing. When the analyst is satisfied that the system is working properly, he hands it over to the users for testing. The importance of system testing by the user must be stressed. Ultimately it is the user must verify the system and give the go-ahead.

During testing, the system is used experimentally to ensure that the software does not fail, i.e., that it will run according to its specifications and in the way users expect it to. Special test data is input for processing (test plan) and the results are examined to locate unexpected results.

A limited number of users may also be allowed to use the system so analysts can see whether they try to use it in unexpected ways. It is preferably to find these surprises before the organization implements the system and depends on it. In many organizations, testing is performed by persons other than those who write the original programs. Using persons who do not know how certain parts were designed or programmed ensures more complete and unbiased testing and more reliable software.

The system is tested as a complete, integrated system. System testing first occurs in the development environment but eventually is conducted in the production environment. Functionality and performance testing are designed to catch bugs in the system, unexpected results, or other ways in which the system does not meet the stated requirements.

The testers create detailed scenarios to test the strength and limits of the system, trying to break it if possible. Editorial reviews not only correct typographical and grammatical errors, but also



improve the system’s overall usability by ensuring that on-screen language is clear and helpful to users. Accessibility reviews ensure that the system is accessible to users with disabilities.

System testing consists of the following five steps:

1. Program testing
2. String testing
3. System testing
4. System documentation
5. User acceptance testing

##### Program Testing

A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. It is the responsibility of a programmer to have an error free program. At

The time of testing the system, there exists two types of errors that should be checked. These errors are syntax and logic.

A syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted key words are common syntax errors. These errors are shown through error messages generated by the computer. A logic error, on the other hand, deals with incorrect data fields out of range items, and invalid combinations.

Since the logical errors are not detected by compiler, the programmer must examine the output carefully to detect them. When a program is tested, the actual output is compared with the expected output. When there is a discrepancy, the sequence of the instructions, must be traced to determine the problem. The process is facilitated by breaking the program down into self- contained portions, each of which can be checked at certain key points.

##### String Testing

Programs are invariably related to one another and interact in a total system. Each program is tested to see whether it conforms to related programs in the system. Each part of the system is tested against the entire module with both test and live data before the whole system is ready to be tested.



##### System Testing

System testing is designed to uncover weaknesses that were not found in earlier tests. This includes forced system failure and validation of total system as it will be implemented by its user in the operational environment. Under this testing, generally we

Take low volumes of transactions based on live data. This volume is increased until the maximum level for each transaction type is reached.

The total system is also tested for recovery and fallback after various major failures to ensure that no data are lost during the emergency.

All this is done with the old system still in operation. When we see that the proposed system is successful in the test, the old system is discontinued.

##### System Documentation

All design and test documentation should be well prepared and kept in the library for future reference. The library is the central location for maintenance of the new system.

##### User Acceptance Testing

An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and that the integrity of important data is maintained. Performance of an acceptance test is actually the user's show. User motivation is very important for the successful performance of the system. After that a comprehensive test report is prepared. This report shows the system's tolerance, performance range, error rate and accuracy.



##### Table 6.1 Test Report with test data

|  |  |  |
| --- | --- | --- |
| **TEST REPORT WITH TEST DATA**  (To be filled by System Analyst/Programmer) | | |
| **Project Name : Restaurant Reservation System** | | |
| **S No.** | **Testing Parameter** | **Observations** |
| A. | INTERFACE TESTING   1. User-friendliness 2. Consistent menus | OK NA |
| B. | CONTROL FLOW TESTING   1. IF-THEN-ELSE 2. DO WHILE 3. CASE-SWITCH | OK  OK OK |
| C. | VALIDATION TESTING   1. Check for improper or inconsistent typing 2. Check for erroneous initialization or default values 3. Check for incorrect variable names 4. Check for inconsistent Data Types 5. Check for relational/arithmetic operators | OK OK OK OK  OK |
| D. | DATA INTEGRITY/SECURITY TESTING   1. Data Insertion/ Deletion/ Updating 2. Boundary condition (Underflow, Overflow Exception) 3. Check for unauthorized access of data 4. Check for data availability | OK OK OK OK |



|  |  |  |
| --- | --- | --- |
| E. | EFFICIENCY TESTING   1. Throughput of the system 2. Response time of the system 3. Online disk storage required by the system 4. Primary memory required by the system | OK OK OK  OK |
| F. | ERROR HANDLING ROUTINES   1. Error description are intelligent/ understandable 2. Error recovery is smooth 3. All error handling routines are tested and executed at least once | OK OK OK |



***CHAPTER – VII***

***CONCLUSION AND FUTURE ENHANCEMENTS***



* 1. **Limitations**

The new system has been designed to meet almost all of the user requirements but it too has certain limitations some of which can be enhanced in the future enhancements or updates

### 7.2 Future Enhancements

Enhancements are the perquisite for development of a system. Every existing system has proposed enhancements which make it better and easier to use and more secure. The enhancements that have been proposed for this system are listed here.

### Conclusion

The development of software includes so many people like user system developer, user of system and the management, It is important to identify the system requirements by properly collecting required data to interact with supplier and customer of the system.

Proper design builds upon this foundation to give a blue print, which is actually implemented by the developers.



# CHAPTER – VIII BIBLIOGRAPHY



### Bibliography

* List of useful Websites
  + <https://www.w3schools.com/html>
  + <https://www.javatpoint.com/javascript-tutorial>
  + <https://github.com>