

ABSTRACT

The Hotel Booking System project aims to develop a comprehensive and efficient platform for managing hotel reservations, enhancing guest experiences, and optimizing hotel operations. The project encompasses the design, development, and implementation of a web-based hotel booking system that caters to both guests and hotel staff. The system facilitates seamless booking processes, real-time room availability updates, secure payment transactions, and robust reservation management functionalities. In today's digital era, the hospitality industry relies heavily on efficient and user-friendly hotel booking systems to manage reservations and enhance guest experiences. This abstract outlines the key components, functionalities, and benefits of a modern hotel booking system.

The hotel booking system is Designed an intuitive and user-friendly interface for guests to search for rooms, view price rate, and make reservations.

TABLE OF CONTENTS

CHAPTER 1.

Introduction	1
1.1 Overview of system	2
1.2 problem statement	3
1.3 proposed systematic	
1.4 objectives	4

CHAPTER 2.

Requirement Specification

2.1 Tools and Technology used	5
2.2 Feasibility Study	6-7

CHAPTER 3.

System Design

3.1 Structure Chart	8
3.2 Module Description	9
3.3 Work breakdown structure	10
3.4 DFD	11
3.5 Context level diagram (0 level DFD)	12
3.6 1 st level DFD	13
3.7 ER Diagram	14-15
3.8 Data Dictionary	16
3.9 Table Structure	17-18

CHAPTER 4.

TESTING & IMPLEMENTATION	19-33
-------------------------------------	--------------

CHAPTER 5.

Document and User Manual	34-39
Future Scope	40
Conclusion	41
References Bibliography	42

LIST OF TABLES

- table 3.1 data dictionary
- table 3.2 administrator
- table 3.3 booking
- table 3.4 customers
- table 3.5 pricing
- table 3.6 reservation

LIST OF FIGURES

- figure 3.1 structure chart
- figure 3.2 level 0 DFD
- figure 3.3 level 1 DFD
- figures 3.4 E.R. Di

CHAPTER 1.

INTRODUCTION:

A hotel booking system, also known as a hotel reservation system, is an online platform that allows customers to search for, select, and book hotel rooms and other accommodations. These systems have become integral to the hospitality industry, facilitating seamless interactions between guests and hotels .

A hotel booking system is essential for modern hotels to remain competitive and meet the evolving needs of travelers. By providing a streamlined, efficient, and user-friendly booking process, hotels can enhance customer satisfaction, optimize operations, and drive revenue growth. As technology advances, these systems continue to evolve, incorporating features like mobile optimization, artificial intelligence, and personalized marketing to further improve the guest experience .

1.1 Overview.

The hotel booking system is a user-friendly website system, specially designed for the every individual for the bookings of the hotel rooms . the functions that are being managed by the hotel management can be done through the system in a much more easy and disciplined manner.

This Hotel booking System is developed in favour of the hotel management team which helps them to save the records of the individual or travel agents to reserve the rooms . It helps the people for booking the hotel rooms though they are away from the destination they can reserve the room .

Identification of the problems of the existing hostel management leads to the development of computerized solution that will be compatible to the existing hostel management with the solution which is more users friendly and more GUI oriented. We can improve the efficiency of the hostel management, thus overcome the drawbacks of the existing management

1.2 Problem statement

In today's digital age, efficient management of hotel bookings is essential for both hoteliers and guests. A hotel booking system aims to streamline the reservation process, providing a seamless experience for customers while optimizing hotel occupancy and revenue. However, challenges such as complex booking procedures, inventory management, and customer satisfaction need to be addressed to create a successful hotel booking system

To overcome these drawbacks we introduced the Hotel booking system project with more efficiency and with updated features such as no acceptance of duplicate values.

1.3 proposed system

This project is aimed at developing a system for keeping records and showing information about or in a hotel. This system will help the hotel managers to be able to manage the affairs of the hostel. This system will provide full information about a availability of the rooms in the hotel. It will show rooms available or not and number of guest in a particular room. This will also provide information on guest who have book the rooms This system will also provide a report on the Summary detail regarding bills . Also included is a User module for employees or the hotel managers. There will also be an administrator module which will accessed by the administrator and has the ability to delete, add and edit employee records. This system will be developed based on Software Development Life Cycle (SDLC) with Vb.net and MsSQL server. Visual Basic is good for the development and design of programs whiles MsSQL is good for databases because of its security and its advanced features and properties.

1.4 OBJECTIVES

1. The primary aim is to the booking process, making it quick and convenient for guests to reserve rooms. This involves providing an intuitive user interface where guests can easily search for available rooms, select their preferences, and complete the booking with minimal effort
2. By automating various tasks such as availability checks, reservation confirmations, and payment processing, the system reduces the need for manual intervention, saving time and effort for both guests and hotel staff.
3. The system would accurately reflect the real-time availability of rooms, ensuring that guests receive up-to-date information when making their bookings.
4. Maintaining an accurate record of available rooms and updating inventory in real-time helps prevent overbooking and ensures optimal utilization of hotel resources.

CHAPTER 2.

2.1 SYSTEM REQUIREMENTS

TOOLS & TECHNOLOGY USED

Hardware

- Ram: 512 MB
- Hard Disk: 2 GB
- Processor: Pentium III or higher

Software

- Visual Studio code
- MySQL
- Connection with front end

2.2- Feasibility Study:

A feasibility study is simply an assessment of the practicality of a proposed plan or project. A feasibility study includes an estimate of the level of expertise required for a project and who can provide it, quantitative and qualitative assessments of the other essential resources, identifications of critical points, a general timetable, and a general cost estimate whether a project is viable or not, i.e. whether it can generate an equal or a higher rate of return during its lifetime requires a thorough investigation of the investment per as well as the level of current expenditure. A feasibility study is an analysis that takes all of a project's relevant factors into account-including economic, technical, legal, and scheduling considerations-to ascertain the likelihood of completing the project successfully.

Most feasibility studies are distinguished for both user and analysts. First, the study often presupposed that when the feasibility document is being prepared, the analyst is in a position to evaluate solutions. Second, most studies tend to overlook confusion inherent in system development the constraints and the assumed attributes.

1. TECHNICAL FEASIBILITY:

- **SYSTEM REQUIREMENTS:** Determine the hardware and software requirements for developing and deploying the hotel booking system.
- **TECHNOLOGY STACKS:** Assess the availability and suitability of technologies and frameworks for building the system, such as web development frameworks, databases, and third-party APIs.
- **INTEGRATION CAPABILITIES:** Evaluate the feasibility of integrating the booking system with existing hotel management systems, payment gateways, and other external platforms.

2. ECONOMIC FEASIBILITY:

- **COST-BENEFIT ANALYSIS:** Estimate the costs associated with developing, implementing, and maintaining the hotel booking system compared to the expected benefits and returns.

- **ROI CALCULATION:** Determine the return on investment (ROI) based on projected revenue increases, cost savings, and operational efficiencies achieved through the system.
- **RISK ASSESMENT:** Identify potential risks and uncertainties that may affect the economic feasibility of the project, such as market competition, changing customer preferences, and technology obsolescence.

3. OPERATIONAL FEASIBILITY:

- **USER ACCEPTANCE:** Assess the willingness and ability of hotel staff to adopt and use the booking system effectively.
- **TRAINING REQUIREMENTS:** Determine the training needs for hotel staff to familiarize themselves with the system's functionalities and operations.
- **CHANGE MANAGEMENT:** Evaluate the impact of implementing the booking system on existing hotel processes and workflows, identifying areas for improvement and adjustment.

4. SCHEDULING FEASIBILITY:

- **TIMELINE & MILESTONES:** Develop a project timeline with specific milestones for each phase of development, testing, deployment, and training.
- **RESOURCE ALLOCATION:** Allocate human and financial resources required for the project, ensuring sufficient staffing, budget, and infrastructure.
- **RISK MITIGATION PLAN:** Develop contingency plans to address potential delays, setbacks, or unforeseen challenges that may arise during the project lifecycle.

SYSTEM DESIGN

3.1 structure chart:

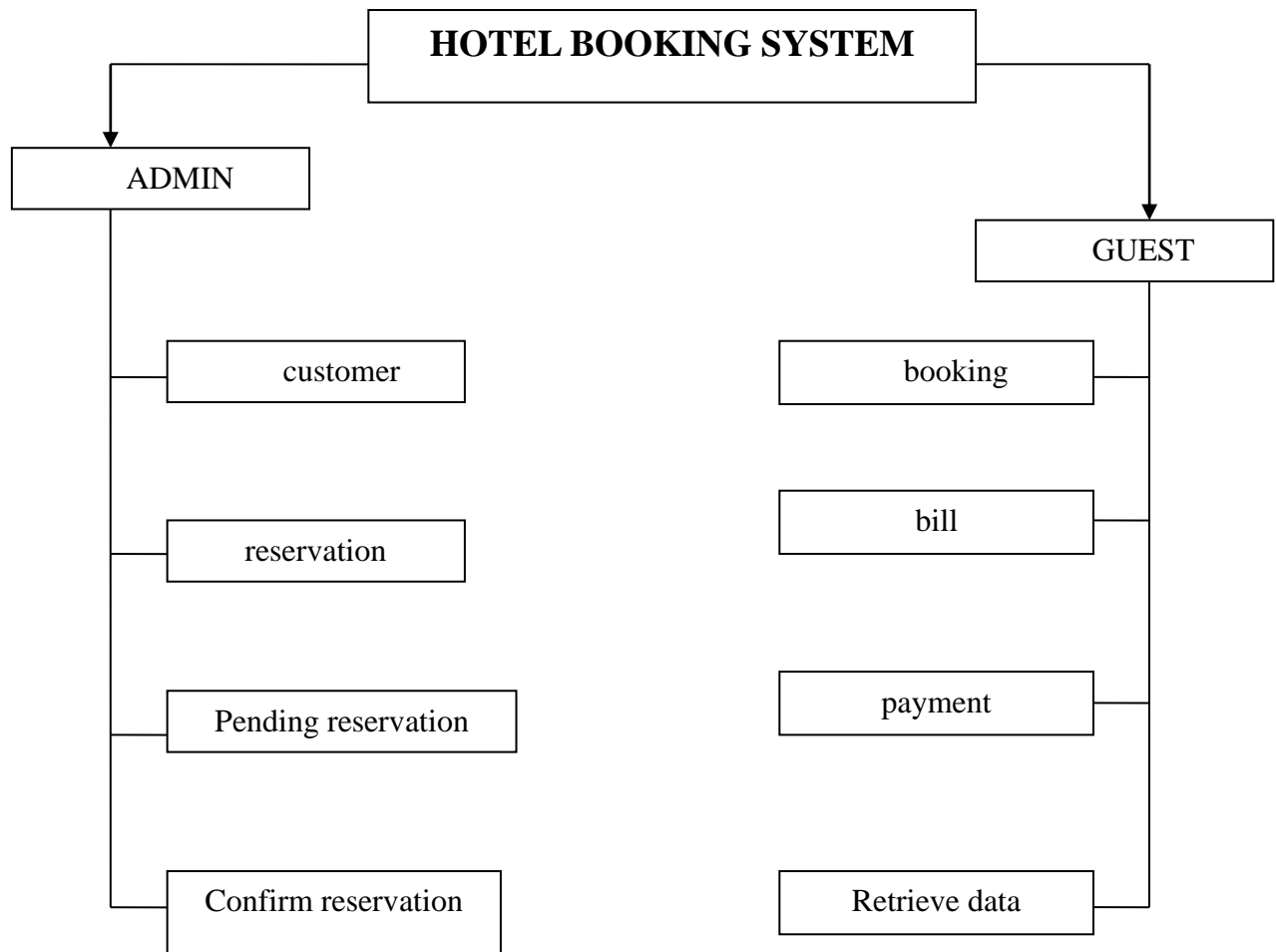


Figure. 3.1

3.2. Module Description

In the proposed system we have divided the information into two modules:

1. ADMIN:

The Administrator can see the whole system and can see the room has available or not. Also the admin can confirm the reservation of the room by the guest as well as cancel the reservation too. **2GUEST:**

The guest can book the room based on the category of the rooms available at the hotel.

3.3. Gantt Chart:

A Gantt chart is a graphical depiction of a project schedule. It's is a type of bar chart that shows the start and finish dates of several elements of a project that include resources, milestones, tasks, and dependencies. Henry Gantt, an American mechanical engineer, designed the Gantt chart.

Gantt chart is a visual view of tasks scheduled over time. Gantt charts are used for planning projects of all sizes and they are a useful way of showing what work is scheduled to be done on a specific day. They also help us view the start and end dates of a project in one simple view.

3.3 WORK BREAKDOWN STRUCTURE

A work breakdown structure in project hotel booking system is a deliverable- oriented breakdown of a project into smaller components. A work breakdown structure is a key project deliverable that organized the team or individuals work into manageable section.

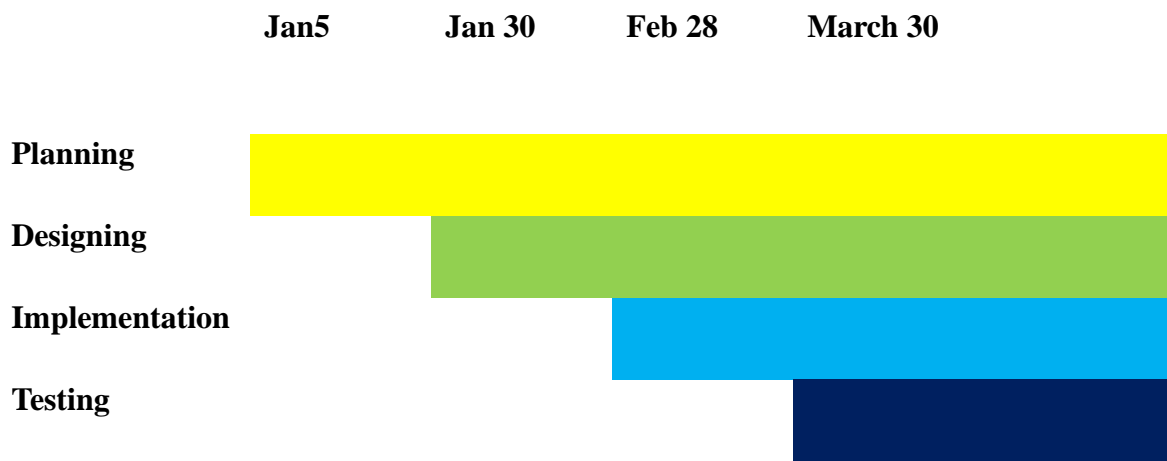
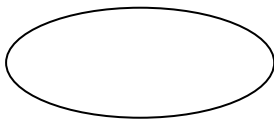


Figure. 3.2

3.4 Data Flow Diagram (DFD)

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. A DFD shows what kind of information will be input to and output from the system. A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. We can perform many levels of DFD.

There are various symbols used to draw a DFD, like



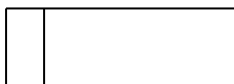
A circle or a bubble represents a process that transforms Incoming data flow(s) into outgoing Flow(s).



A square defined a source (originator) or destination (Sink) of system data.



An arrow identifies data flow – data in motion.
It is a pipeline through information flows.



An open rectangle is a data store—data at rest, or a Temporary repository of data.

CONTEXT DIAGRAM (0 LEVEL DFD)

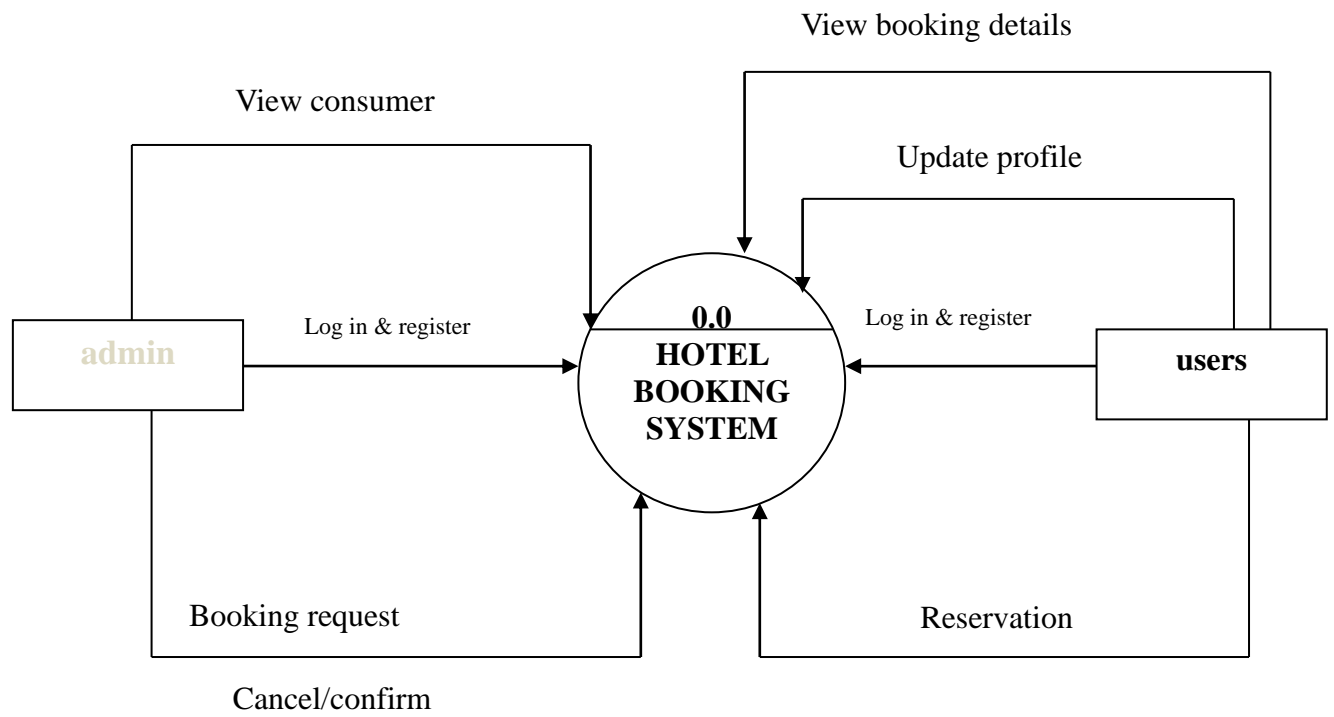


Figure. 3.3

1st LEVEL DFD

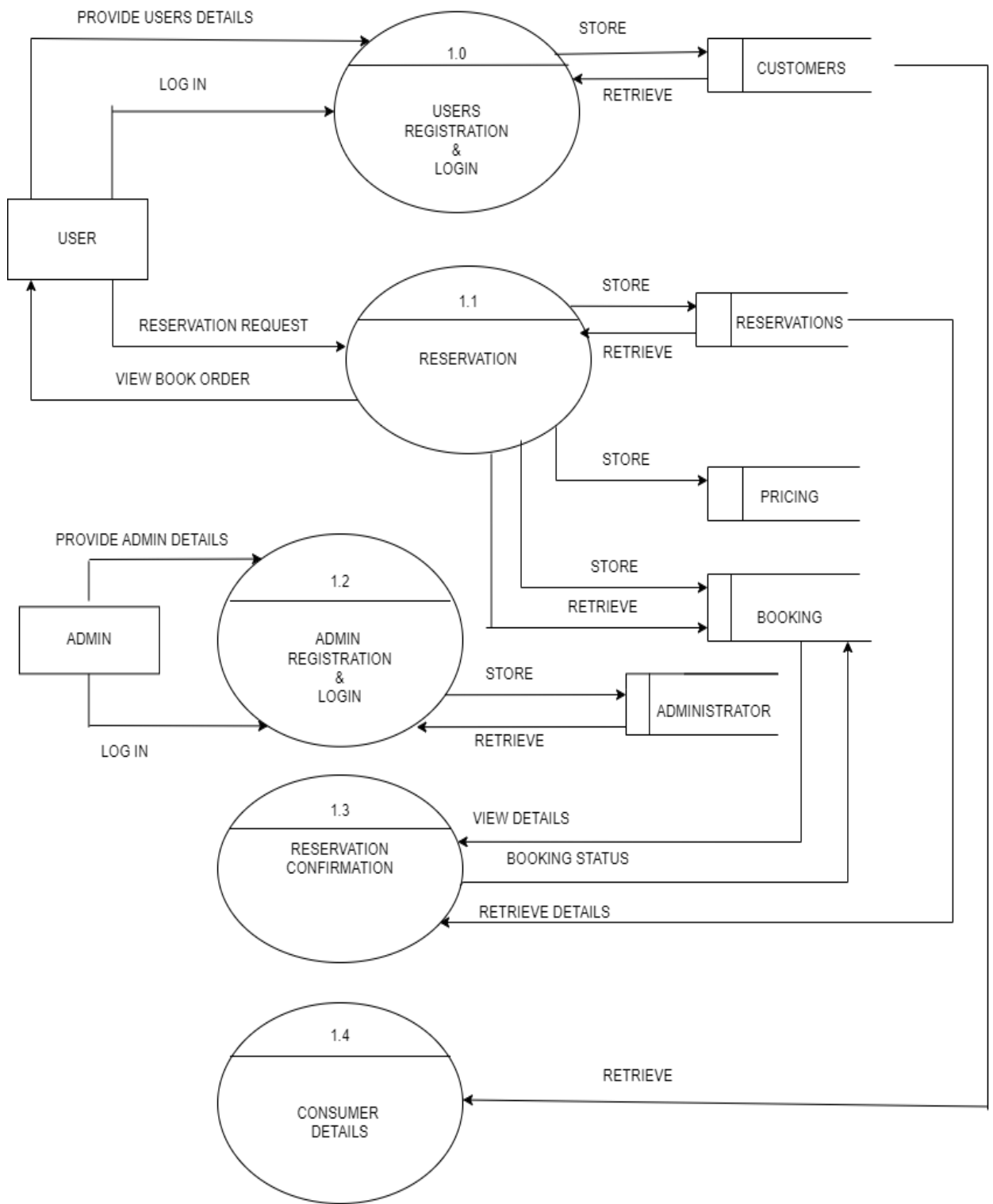


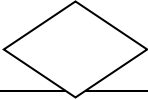

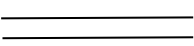

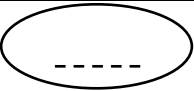


Figure. 3.4

ER DIAGRAM

The E-R model was introduced by P.P. Chen. Entity-relationship modeling is a detailed, logical representation of the entities, associations and data elements for an organization or business area. This technique is used in database design that helps to describe how entities in an enterprise are related to one another. The entity relationship model for data uses there features to describe data. These are the following:

The basic Symbols used in E-R Diagram are:

<i>SYMBOLS</i>	<i>REPRESENTS</i>
<i>Rectangle</i> 	<i>Entity set</i>
<i>Ellipse</i> 	<i>Attributes</i>
<i>Diamond</i> 	<i>Relationship</i>
<i>Lines</i> 	<i>Links attributes to entity sets & Entity sets to relationship sets.</i>
<i>Doubled lines</i> 	<i>Total participations of an entity in a relationship sets.</i>
<i>Oval with solid Line</i> 	<i>Primary key</i>
<i>Oval with dot Line</i> 	<i>Foreign key</i>

ENTITY RELATIONSHIP DIAGRAM

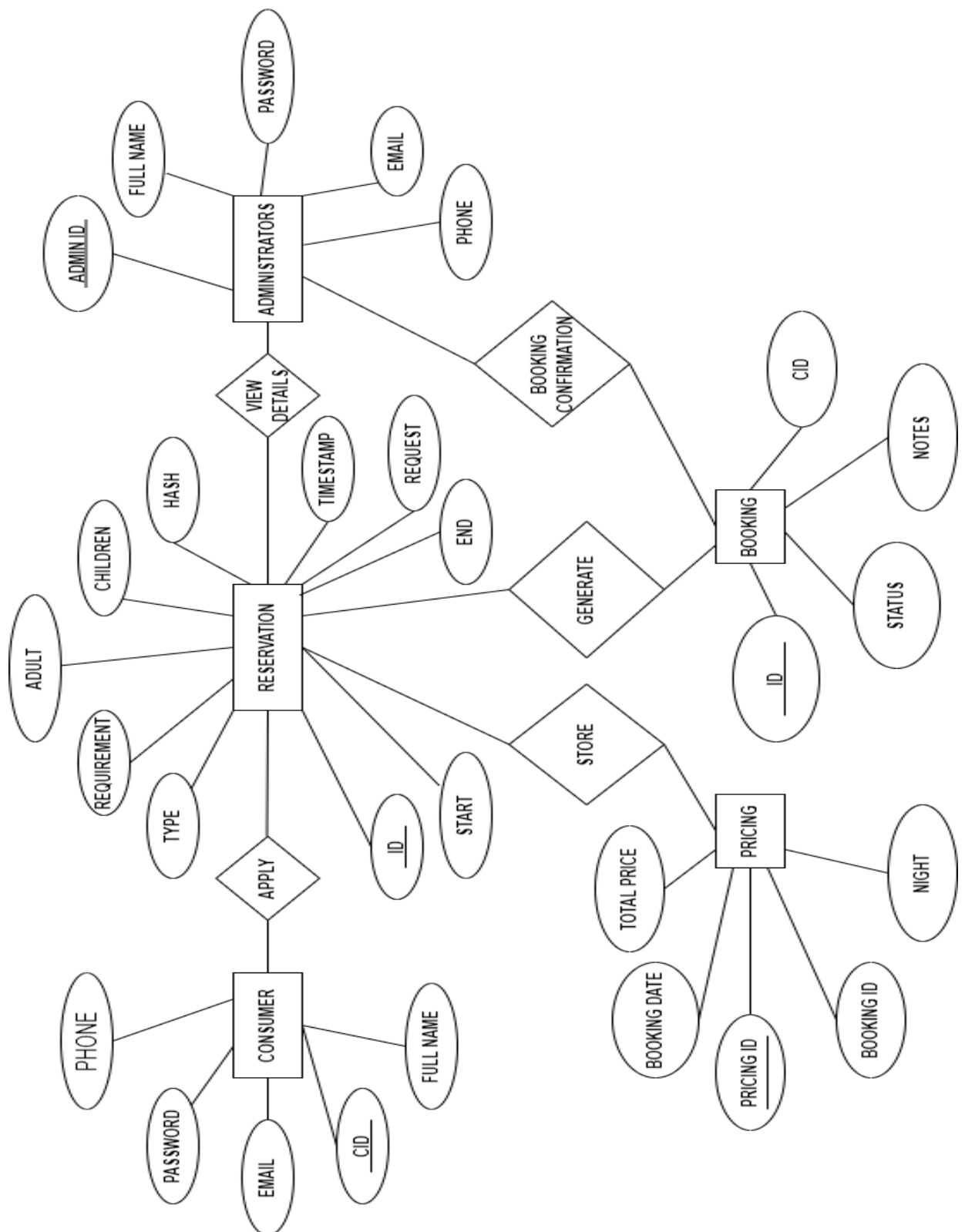


Figure. 3.5

DATA DICRIONARY

S.NO	Field name	Source table	description	constraints
1	adults	reservation	Reserve room	null
2	admin	administrator	Admin	primary
3	Booking id	pricing	Id of guest	candidate
4	Booking date	pricing	Date of booking	Not null
5	children	reservation	For children's	null
6	cid	booking	Customer id	candidate
7	cid	customer	Customer id	primary
8	emails	customer	Unique emails	Not null
9	emails	administrator	Admin emails	candidate
10	end	reservation	Upto date	Not null
11	Full name	customer	Guest name	Not null
12	Full name	administrator	Admin name	Not null
13	Full name	customer	Customer name	Not null
14	hash	reservation	To stored value	Not null
15	id	reservation	Unique id	primary
16	id	booking	Unique booking id	Primary
17	nights	pricing	Prices for nights	Not null
18	password	administrator	Admin log in	Not null
19	Password	customer	Customer sign in	Not null
20	phone	Customer	Contact no	Not null
21	Pricing id	Pricing	Pricing of rooms	primary
22	request	reservation	Requesting for reserve	Not null
23	requirements	reservation	Requirements needed	Not null
24	start	booking	Start date	Not null
25	status	reservation	Pending or cancelled	Not null
26	Time stamp	reservation	Time start at	Not null

TABLE STRUCTURE

ADMINISTRATOR

field_name	data_type	size	constraints
adminId	int	20	primary
fullname	varchar	100	Not null
password	varchar	100	Not null
email	varchar	30	candidate

BOOKING

field_name	data_type	size	constraints
id	int	11	primary
cid	int	11	candidate
status	enum		Not null
notes	varchar	500	Null

CUSTOMER

field_name	data_type	size	constraints
cid	int	10	primary
fullname	varchar	100	Not null
email	varchar	50	Not null
password	varchar	150	Not null
phone	varchar	25	Not null

PRICING

field_name	data_type	size	constraints
pricing_id	int	11	primary
booking_id	int	11	candidate
nights	int	11	Not null
total_price	double	10	Not null
booked_date	date		Not null

RESERVATION

field_name	data_type	size	constraints
id	int	11	primary
start	int	30	Not null
end	varchar	30	Not null
type	enum		Not null
requirement	enum		Null
adults	int	2	Null
children	int	2	Null
requests	varchar	500	Null
timestamp	timestamp		Not null
hash	varchar	100	Not null

4.1 TESTING AND IMPLEMENTATION

TESTING and TYPES:

Testing makes a logical assumption that identify (a) whether all the parts of the system are correct and (b) the goal of the candidate system is successfully achieved or not. The first test is to see whether it provides correct output result or not as per user requirements. Testing can be set to the process of “finding the errors we don’t know is there”. The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program or application. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding errors or other defects, and verifying that the software product is fit for use.

1.Verification: it refers to the set of tasks that ensure that software correctly implements a specific function.

2.Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements

SOFTWARE TESTING:-

Software Testing can be broadly classified into two types:

1. **Manual Testing:** Manual testing includes testing software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behaviour or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing

Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

1.Automation Testing: Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses software to test the product.

This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly and repeatedly.

Testing makes a logical assumption that identify (1) whether all the parts of the system are correct and (2) the goal of the candidate system is successfully achieved or not. The first test is to see whether it provides correct output result or not as per user requirements. Testing can be said to be the process of “finding the error we don’t know is there”.

Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

Target of the test are -

- **Errors** - These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.
- **Fault** - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.
- **Failure** - failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

A system should always be tested thoroughly before implementing it, as regards its individual programs, the system as a whole, user’s acceptance etc. Effective testing early in the process translates directly into a long term testing involves:

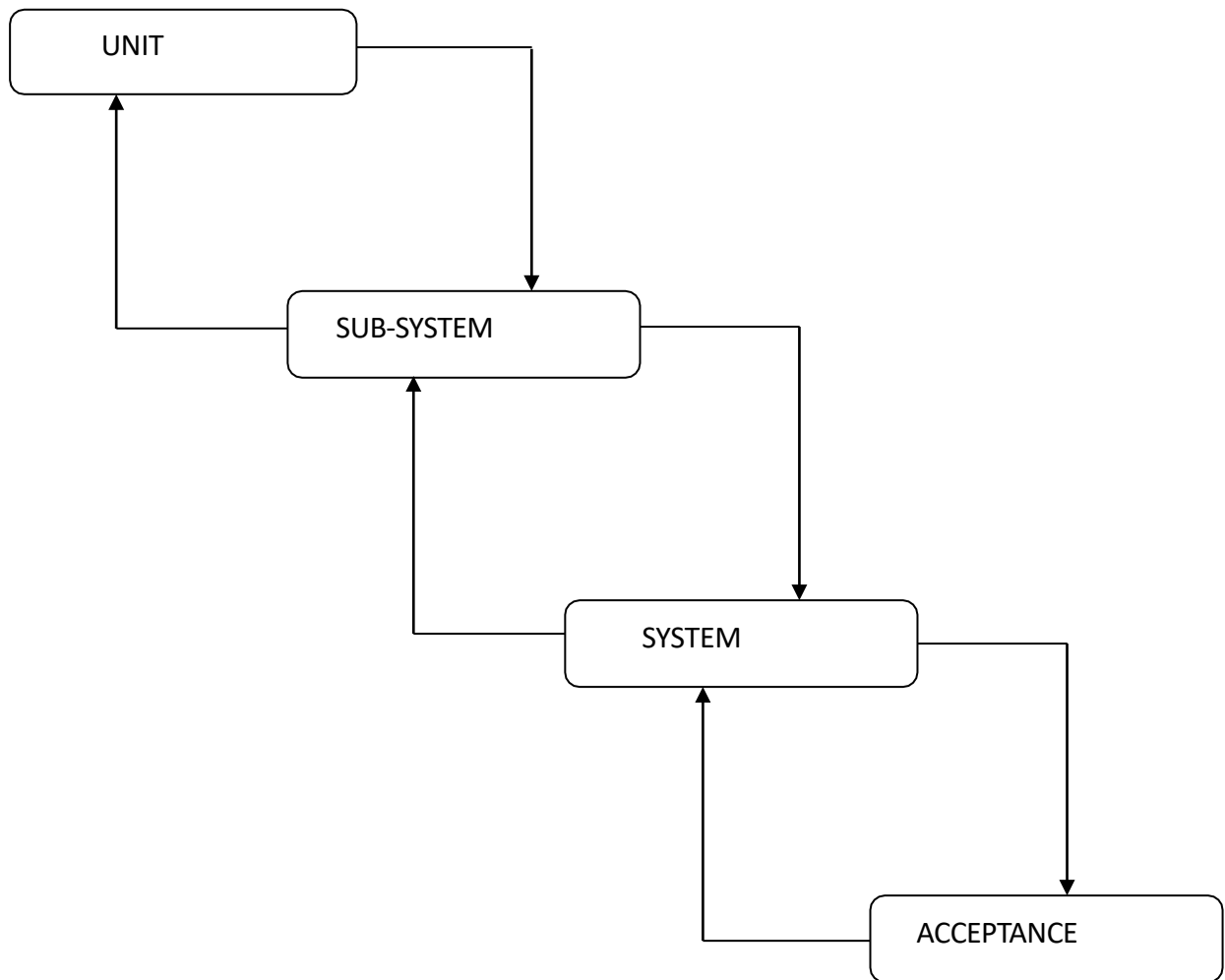
1.Module Testing

2.Sub-System Testing

3.System Testing and

4.Acceptance Testing

With the help of the following diagram we can illustrate the testing process:



1. Module Testing or Unit Testing:-

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. Module testing is a white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. It checks for errors in individual modules from isolating them from the rest of the software product and the errors are corrected for each individual unit. Unit test are a collection of tests written by a developer during the software development process. Module tests are a collection of tests written by a tester after some code has been

written by a developer. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

Here the program unit or functions are tested with:-

- 1) Legal data values i.e., simple realistic value,
- 2) Null values,
- 3) Illegal data values and
- 4) Program's behaviour is tested under stress of extreme conditions (either the quality of data we provide or its operating environment).

Module Testing is recommended because

- Probability of identifying errors or bugs on smaller chunks of program becomes higher.
- Multiple modules can be tested simultaneously and hence supports parallel testing.
- Complexity of testing can be easily managed.
- For Module Testing, designing an important segment. While designing test cases for a module test, a tester has to take two things into consideration.
 - Specification for the module
 - The module's source code
- Analyze the module's logic by using one or more of the white box methods, and then supplement these test cases by applying black box methods to the modules specification
- Once the test case is designed, the next step is to combine the module for testing. For this, the method used is either an Incremental or non-Incremental method.
- Non-incremental method- all modules are tested independently. First, it

combines all modules and then test the whole program

- Incremental method- each module is tested first and then gradually incremented to the tested collection. It does a step wise retesting
- Incremental Testing, there are two approaches – Top down and Bottom Up testing
- To execute the module with the selected data, it requires a driver for supplying the test data, monitoring the execution and capturing the results

Challenges in Module Testing

- Non-incremental testing requires more work
- Misunderstanding test doubles
- Debugging test often
- Need to understand the code

Here are some of the best practices which are recommended to be considered before conducting Module Testing.

- **Test Cases Review:** We should be getting our test cases reviewed with the required team or peers in order to make sure that we have 100% test coverage.
- **Automated test execution:** We should identify the test scenarios or functionalities which could be automated through the testing tool. This will save us time in executing such test cases over and over.
- **Avoid confusion:** There should not be any confusion related to the data input and the expected output to be tested for any module.
- **Examine variables:** The variables which are supposed to be changed or to be kept constant all the time should be monitored thoroughly and examined at the time of recording

test results.

- Re-use the test cases and avoid any test case duplication.

Advantages:-

- Higher level of Automation is achieved.
- Exhaustive testing is possible.
- Changes to the model can be easily tested.

Disadvantages:-

- Requires a formal specification or model to carry out testing.
- Changes to the model might result in a different set of tests altogether.
- Test Cases are tightly coupled to the model.

Unit Testing Tasks:-

- Unit Test Plan
 - Prepare
 - Review
 - Rework
 - Baseline
- Unit Test Cases/Scripts
 - Prepare
 - Review
 - Rework
 - Baseline
- Unit Test
 - Perform

1. Sub-System Testing

Sub-System involves testing a collection of modules that have been integrated into sub-system. The sub-system may be independently designed and implemented. This phase involves testing collections of modules which have been integrated into sub-systems. Sub-systems may be independently designed. The most common problems which arise in large software systems are sub-system interface mismatches. The sub-system test process should

therefore concentrate on the detection of interface errors by rigorously exercising the interfaces. System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together. System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behaviour produced or observed when a component or system is tested.

System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specifications.

Sub systems are integrated to make up the entire system. The testing process is concerned with finding errors that result from unanticipated interactions between sub- systems and system components. It is also concerned with validating that the system meets its functional and non-functional requirements. The errors that are not discovered into modules of unit testing are not discovered into modules of unit testing but discovered into sub-system testing may be:

1. Errors in the interface between two sub-systems.
2. Errors or misunderstanding in interpreting the module specification.
3. Errors due to side-effects of a module (i.e. some module may change the global data variables).

1. System Testing:-

System testing is a method of monitoring and assessing the behaviour of the complete and fully- integrated software product or system, on the basis of pre-decided specification and functional requirements. In System Testing interface between two sub-systems are tested because errors existed in the interface between sub-systems may not be visible until they are combined tested as a complete unit. System testing means testing the

system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not. System testing is done after integration testing. This plays an important role in delivering a high-quality product. In System testing, the functionalities of the system are tested from an end-to-end perspective. System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

Key areas of system Testing:-

Some of the aspects, on which system testing focuses are:

- **Performance:** It makes sure that the software system performs as per requirements of the user, without depicting any defects or issues.
- **Security:** Protects the product from any security breaches, data theft, etc., which can sacrifice critical data & information of the organization.
- **Recovery:** Ensures that the recovery of the system is as per expectations and in an accurate condition.
- **Interface:** System testing also focuses on the interface of the product and ensures that all requirements are met accurately and no issues occur when two components of the system are integrated together.

Reasons for Module testing:-

As modules are an integral part of a software program, testing them becomes extremely important for defining the qualities and effectiveness of the software. Apart from testing modules, module testing is also implemented by the team to test different components of the software separately. Other reasons for module testing are:

- It is a way of managing combined elements of the software.

System testing is divided into two types: -

- White-Box Testing or Structural Testing
- Black-Box Testing or Functional Testing

1. White Box Testing

White-Box Testing is concerned with the implementation of the programs. In this type of testing different programming structures and data structures used in the programs are tested for proper operations. This test concentrates on the examinations of coding methods. The system software engineers and programmers formulate test-cases and select test data. The system designers create test cases which have the possibility of locating errors. White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised. White box testing technique is used by both the developers as well as testers. It helps them to understand which line of code is actually executed and which is not. This may indicate that there is either a missing logic or a typo, which eventually can lead to some negative consequences.

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clears box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc.

In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.

Types of White Box testing

Unit Testing:

The developer carries out unit testing in order to check if the particular module or unit of code is working fine. The Unit Testing comes at the very basic level as it is carried out as and when the unit of the code is developed or a particular functionality is built.

Static and dynamic Analysis:

Static analysis involves going through the code in order to find out any possible defect in the code. Dynamic analysis involves executing the code and analysing the output.

Statement Coverage:

In this type of testing the code is executed in such a manner that every statement of the application is executed at least once. It helps in assuring that all the statements execute without any side effect.

Branch Coverage:

No software application can be written in a continuous mode of coding, at some point we need to branch out the code in order to perform a particular functionality. Branch coverage testing helps in validating of all the branches in the code and making sure that no branching leads to abnormal behaviour of the application.

Security Testing:

Security Testing is carried out in order to find out how well the system can protect itself from unauthorized access, hacking – cracking, any code damage etc. which deals with the code of application. This type of testing needs sophisticated testing techniques.

Working process of white box testing:

- Input: Requirements, Functional specifications, design documents, source code.
- Processing: Performing risk analysis for guiding through the entire process.

- Proper test planning: Designing test cases so as to cover entire code. Execute rinse- repeat until error-free software is reached. Also, the results are communicated.
- Output: Preparing final report of the entire testing process.

White-Box Testing is done to ensure:

- That all independent paths within a module have been exercised at least once.
- All logical decisions verified on their true and false values.
- All loops executed at their boundaries and within their operational bounds internal data structures validity.

White-Box Testing discovers the following types of bugs:

- Logical error tends to creep into our work when we design and implement functions, conditions or controls that are out of the program.
- The design errors due to difference between logical flow of the program and the actual implementation.
- Typographical errors and syntax checking.

i) Black-Box Testing:

Black box testing, also known as Behavioural Testing is a software testing method in which the internal structure/design/implementation of the system being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. Black-Box testing is concerned with the proper examination of the programs specifications. In this testing, each functions as well as sub-programs used in the main programs are first identified.

Hereafter, test cases are devised to test each function or sub-program separately. Test cases are decided solely on the basis of the requirement or specifications of the programs are not on the basis of the coding of the modules or the type of the data structures used.

Because black box testing is not concerned with the underlying code, then the techniques can be derived from the requirement documents or design

specifications and hence testing can start as soon as the requirements are written.

Some Black Box Testing Techniques are:

- Cause / Effect Graphing
- Classification Tree Method

This method attempts to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external database access.
- Behaviour or performance errors.
- Initialization and termination errors.

Following black box testing techniques are used for testing the software application.

- Boundary Value Analysis (BVA)
- Equivalence Class Partitioning
- Decision Table based testing
- Cause-Effect Graphing Technique
- Error Guessing

In Black Box Testing, the tester tests an application without knowledge of the internal workings of the application being tested. Data are entered into

the application and the outcome is compared with the expected results; what the program does with the input data or how the program arrives at the output data is not a concern for the tester performing black box testing. All that is tested is the behaviour of the functions being tested.

This is why black box testing is also known as functional testing which tests the functionality of a program. Note we can also have non-functional black box testing, such as performance testing which is a type of black

4) Acceptance Testing:

Acceptance Testing is the system testing performed by the customer to determine whether or not to accept the delivery of the system. User Acceptance is defined as a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to the Market or Production environment. The main purpose of this testing is to validate the end to end business flow. It does not focus on cosmetic errors, Spelling mistakes or System testing. This testing is carried out in a separate testing

Environment with production like data setup. It is a kind of black box testing where two or more end users will be involved. Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system. This last phase of testing determines whether or not the program is considered as finished product and can be released for general use. It differs from module testing or system testing in the two following ways.

In acceptance testing we check the program's user-friendliness, robustness and on-line help assistance. Acceptance testing often demonstrates errors in the system requirements definitions.

It is the process of setting with real data in the information, which the

system in intended to manipulate.

Acceptance criteria are defined on the basis of the following attributes:

- Functional Correctness and Completeness
- Data Integrity
- Data Conversion
- Usability
- Performance
- Timeliness
- Confidentiality and Availability
- Install ability and Upgradability
- Scalability

Acceptance Testing is the system testing performed by the customer to determine whether or not to accept the delivered of the system. This last phase of testing determines whether or not the program is considered as finished product and can be released for general use. It differs from module testing or system testing in two following ways:

- 1) It is typically done by the end-user nut not the programmer,
- 2) It is carried out without the knowledge of the internal structure and organisation of the program.

In acceptance testing we check for the program's user-friendliness, robustness and online (help) assistance. Acceptance testing often demonstrates errors in the system requirement's definition. It is the process of setting the system with real data the information, which the system is intended to manipulate.

A system's acceptance testing is a final system test performed by end-user using real data over an extended time period. It is an extensive test that addresses three levels of acceptance testing:

- 1) Verification by Alpha Testing

2) Validation or Beta Testing and

Alpha Testing:

Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or the public. The focus of this testing is to simulate real users by using a black box and white box techniques. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually, the testers are internal employees of the organization. As such alpha testing is done on a prototype, in-depth reliability testing, installation testing, and documentation testing can be ignored. A good alpha test must have a well-defined Test plan with comprehensive test cases. Various activities involved in alpha testing are logging defects, fixing defects, retesting, several iterations, etc. Although Alpha testing is not completely functional, QA team must ensure that whatever is on hand should be thoroughly tested, especially those which has to be sent to the customer. In the first phase of alpha testing, the software is tested by in-house developers during which the goal is to catch bugs quickly. In the second phase of alpha testing, the software is given to the software QA team for additional testing. Alpha testing is often performed for Commercial off-the-shelf software (COTS) as a form of internal acceptance testing, before the beta testing is performed. This is a form of internal acceptance testing performed mainly by the in-house software QA and testing teams. Alpha testing is the last testing done by the test teams at the development site after the acceptance testing and before releasing the software for beta test.

Alpha testing can also be done by the potential users or customers of the application. But still, this is a form of in-house acceptance testing.

To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing. Alpha testing has two phases:

The first phase of testing is done by in-house developers. They

either use hardware- assisted debuggers or debugger software.

The aim to catch bugs quickly. Usually while alpha testing; a tester will come across to plenty of bugs, crashes, missing features, and docs.

1. While the second phase of alpha testing is done by software QA staff, for additional testing in an environment. It involves both black box and White Box Testing.

Beta Testing:

Beta Testing is one of the Acceptance Testing types, which adds value to the product as the end- user (intended real user) validates the product for functionality, usability, reliability, and compatibility. Inputs provided by the end-users helps in enhancing the quality of the product further and leads to its success. This also helps in decision making to invest further in the future products or the same product for improvisation. Since Beta Testing happens at the end user's side, it cannot be the controlled activity. Beta testing also known as user testing takes place at the end users site by the end users to validate the usability, functionality, compatibility, and reliability testing.

Beta testing adds value to the software development life cycle as it allows the "real" customer an opportunity to provide inputs into the design, functionality, and usability of a product. These inputs are not only critical to the success of the product but also an investment into future products when the gathered data is managed effectively.

There are number of factors that depend on the success of beta testing:

- Test Cost
- Number of Test Participants
- Shipping

Real world compatibility for a product can be ensured to a greater extent through this testing, as a great combination of real platforms is used here

for testing on a wide range of devices, OS, Browsers, etc.

As a wide range of platforms which the end users are actually using, might not be available to the internal testing team during the QA, this testing also helps to uncover the hidden bugs and gaps in the final product.

Few specific platforms will cause the product to fail with showstopper bug which was not covered during QA. And this helps in improvising/fixing the product to be a compatible one with all possible platforms.

Known Issues, which are accepted by the Product Management team, may take a great turn when the end user faces the same issue and may not be comfortable while using the product. In such cases, this testing helps to analyse the impact of known issues on the entire product as the user experience gets hampered and is not acceptable for any successful business.

Purpose of Beta Testing:

1) Beta Test provides a complete overview of the true experience gained by the end users while experiencing the product.

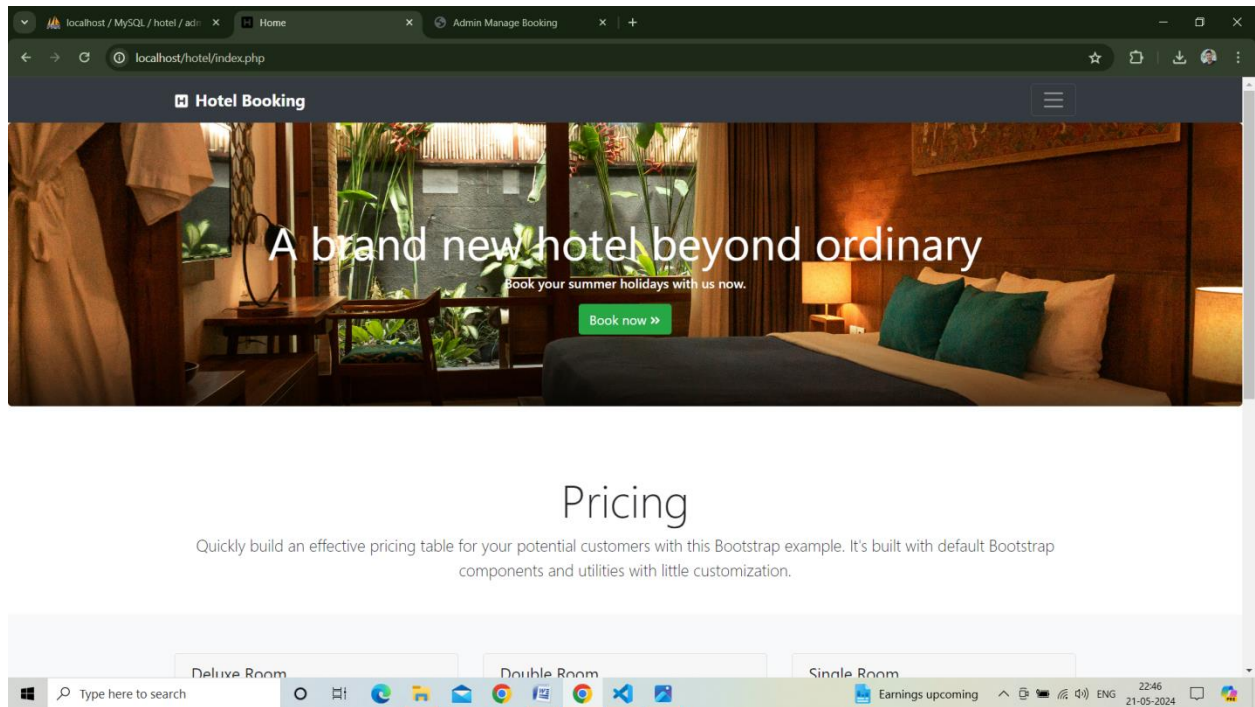
2) It is performed by a wide range of users and the reason for which the product is being used varies highly. Marketing managers focus on target market's opinion on each and every feature, while a usability engineer / common real users focus on product usage and easiness, technical users focus on installation and uninstallation experience etc.

Testing Reports:

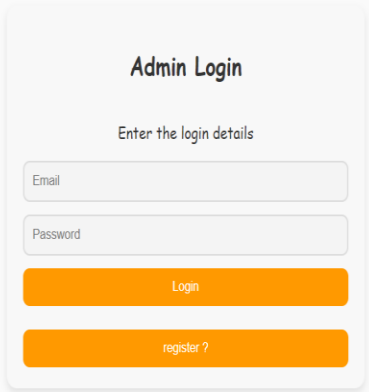
- Only Alphabets are accepted in Name.
- Only Numeric is accepted in Account and Basic
- When entering data if any fields remains blank then it will show error

DOCUMENTS & USER MANUAL

HOTEL BOOKING SYSTEM HOME INTERFACE



ADMIN LOGIN HOME PAGE



The image shows a web form for admin login. It is centered on a light gray background. The form itself is a white rounded rectangle with a subtle drop shadow. At the top of the form is the title 'Admin Login' in a bold, black, sans-serif font. Below the title is the instruction 'Enter the login details' in a smaller, regular font. There are two input fields: the first is labeled 'Email' and the second is labeled 'Password', both in a small gray font. Below these fields are two orange buttons. The first button is labeled 'Login' and the second button is labeled 'register ?'.

Admin Login

Enter the login details

Email

Password

Login

register ?

SYSTEM LOGIN FOR USERS

Hotel Booking System Login

Login

Email *

email address

Password *

password

Not registered? [Register here.](#)

Home

Sign in

HOTEL MANAGEMENT SYSTEM REGISTRATION PAGE

Hotel Management System Registration

Sign Up

Name

Phone Number

Email *

Password *

Confirm Password *

Already registered? [Sign in here.](#)

[Home](#)[Submit](#)

HOTEL BOOKING FOR CANCELLATION & CONFIRMATION

Hotel Booking

Reservations

Total 6

Customers

Total 4

Confirmed Reservations

Total 2

Pending Reservations

Total 1

Reservation

Customers

#	Email	Start	End	Room type	Timestamp	Status	Notes
1	ud@gmail.com	2024-05-18	2024-05-20	Deluxe	2024-05-17 13:30:54	CONFIRMED	NILL
2	ud@gmail.com	2024-05-17	2024-05-24	Single	2024-05-17 17:42:58	CANCELLED	NILL
3	pd@gmail.com	2024-05-17	2024-05-20	Double	2024-05-17 18:32:47	CONFIRMED	NILL
4	chukameyampang@gmail.com	2024-05-18	2024-05-19	Single	2024-05-17 19:14:26	CANCELLED	NILL
5	mimi@gmail.com	2024-05-21	2024-05-24	Double	2024-05-20 22:28:45	CANCELLED	NILL
6	mimi@gmail.com	2024-05-22	2024-05-25	Deluxe	2024-05-21 21:58:10	PENDING	NILL

To confirm and cangle select a row

With selected: Confirm Cancel

© MCA FINAL PROJECT

FUTURE SCOPE

It is easy to extend the system that we have proposed. A Guest can see the available rooms in the websites by signing it . In future we can implement some features for “HOSTEL BOOKING SYSTEM” project. Hotel booking Software System is offering a maximum of stability, cost-effectiveness and usability. It provides the most flexible and adaptable standards management system solutions for hotel. It offers numerous advantages for both hoteliers and guests alike. The system will be very useful for the future as you can book a hotel room by signing it on the websites.

The system will support efficient inventory management, flexible pricing strategies and integration with other booking channels. security measures such as data encryption and secure payment will provide for the better version and for the better future.

CONCLUSION

The Hotel Booking System project represents a significant advancement in the hospitality industry, offering a comprehensive solution to manage hotel reservations, improve guest experiences, and optimize hotel operations. This is a web-based platform with intuitive and user-friendly interfaces, the system ensures seamless booking processes for guests, empowering them to search for rooms, view price rates, and make reservations with ease . Furthermore, the system's incorporation of real-time room availability updates and secure payment transactions enhances the reliability and security of the booking process, instilling trust and confidence in guests. Meanwhile, the robust reservation management functionalities provided cater to the needs of hotel staff, enabling efficient handling of bookings, cancellations, and modifications.

REFERENCES BIBLIOGRAPHY

<https://sirvoy.com/>

<https://hoteltechreport.com/revenue-management/hotels-reservation-system>

<https://www.mews.com/en/products/booking-engine>