**Project Report On**

**<<ChatBot and Secure Minor>>**



Prepared By

**<<Elina Singh>><< RA2011042010041>>**

**<<Reny Patel>><< RA2011042010045>>**

**<<Aditi Balaji>><< RA2011042010053>>**

**Software Engineering Course**

**(2022)**

**School of Computing**

**Department of Data Science and Business Systems**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

## KATTANKULATHUR- 603 203.



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

## KATTANKULATHUR- 603 203.

**School of Computing**

**Department of Data Science and Business Systems**

**Certificate**

This is to certify that Student Name (Roll Number), student of Bachelor of Technology, III Semester, Department of <<Computer Science>> of SRM Institute of Science and Technology, has pursued the Project titled <<Secure Minor>>under the supervision and Internal guidance of Dr. K.Arthi and the report has been submitted in the Year 2022.

Inter Examiner

Signature

**ACKNOWLEDGEMENT**

We/I express my sincere regard and indebtedness to my project guide Dr. K.Arthi, Associate Professor, Department of << >> of SRM Institute of Science and Technology for her valuable time, guidance, encouragement, support and cooperation throughout the duration of our project. This project helped in understanding the various parameters which are involved in the development of a software project and the integration of front end along with the back end to create a fully functional web application.

Student Name

Roll Number

**ABSTRACT**

**Sample is given here**

This project aims at creating on Hotel Management System which can be used by Admin and Customers. The admin to advise/publish the availability of rooms in different hotels and customers are checking the availability of room in required hotel. Customers should be able to know the availability of the rooms on a particular date to reserve in hotel. They should be able to reserve the available rooms according to their need in advance to make their stay comfortable. The Admin hands the booking information of customers. The users can register and log into the system. The administrator will know the details of reservation and daily income. The hotel department maintain the seat availability and booking details in certain database. This project provides high security to Admin and user information.

The main objective of this project is to design a hotel management system for running a hotel business. The system should be as flexible as possible so that it can be used for different hotels. You have to find out which procedures hotels have used for different hotels. You have to find out which procedures hotels have and based on that information, you should create a system which makes it efficient. You need to find out how a hotel system works on the internet, use your own experience or directly talk to people in the hotel business. The more diverse the sources of your information are, the better will be the resulting system and, possibly, your grade.

**INDEX**

**Content Page No.**

**CHAPTER 1 : INTRODUCTION**

**1.1 INTRODUCTION**

**1.2 AIM**

**1.3 EXISTING SYSTEM**

**1.4 PROPOSED SYSTEM**

**1.4.1 BUSINESS CASE**

**1.5 FEASIBILITY STUDY**

**1.5.1 OPERATIONAL FEASIBILITY**

**1.5.2 TECHNICAL FEASIBILITY**

**1.5.3 ECONOMICAL FEASIBILITY**

**1.6 GIANT CHART**

**1.7 ORGANISATION OF THE REPORT**

**1.7.1 INTRODUCTION**

**1.7.2 SOFTWARE REQUIREMENTS SPECIFICATION**

**1.7.3 DESIGN & PLANNING**

**1.7.4 IMPLEMENTATION DETAILS**

**1.7.5 RESULTS AND DISCUSSION**

**1.8 STAKE HOLDER ANALYSIS**

**CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION**

**2.1 Software Requirements**

**CHAPTER 3 : DESIGN & PLANNING**

**3.1 Software Development Life Cycle Model**

**3.1.1 << >> MODEL**

**3.2 Architecture Diagram**

**3.3 State Diagram**

**3.4 Collaboration Diagram**

**3.5 DFD Diagram**

**3.7 Risc Management Plan**

**CHAPTER 4 : IMPLEMENTATION DETAILS**

**4.1 : FRONT END**

**4.1.1 HTML**

**4.1.2 Css**

**4.1.3 JavaScript**

**4.2 : BACK END**

**4.2.1 PHP**

**4.2.2 MySQL**

**CHAPTER 5 : TESTING AND IMPLEMENTATION**

**5.1 : UNIT TESTING**

**5.2 : INTEGRATION TESTING**

**5.3.2 Test Cases**

**5.3 : SOFTWARE VERIFICATION AND VALIDATION**

**5.3.3 Test Cases**

**5.4 : Black-Box Testing**

**5.3.4 Test Cases**

**5.5 : White-Box Testing**

**5.3.5 Test Cases**

**5.6 : SYSTEM TESTING**

**5.3.6 Test Cases**

**CHAPTER 6 : Conclusion And Future Work**

**CHAPTER 7 : Reference**

**CHAPTER 1 : INTRODUCTION**

**1.1 INTRODUCTION**

The project, Hotel Management System is a web-based application that allows the hotel manager to handle all hotel activities online. Interactive GUI and the ability to manage various hotel bookings and rooms make this system very flexible and convenient. The hotel manager is a very busy person and does not have the time to sit and manage the entire activities manually on paper. This application gives him the power and flexibility to manage the entire system from a single online system. Hotel management project provides room booking, staff management and other necessary hotel management features. The system allows the manager to post available rooms in the system. Customers can view and book room online. Admin has the power of either approving or disapproving the customer’s booking request. Other hotel services can also be viewed by the customers and can book them too. The system is hence useful for both customers and managers to portable manage the hotel activities.

**1.2 AIM**

The mission is to facilitate easy management and administration of a hotel with capabilities to do Booking or reservations of the rooms, Cancellation of the rooms, Cash billing, Room service, Restaurant service, Restaurant Billing, Total Billing, Travels arrangement etc. using the automated hotel management software. One can Keep detailed records or info on an unlimited amount of customers. The system lets the user Know which all rooms are available for occupancy at any point of time. This makes the Booking considerably faster. And thus helps the hotel in better management and reduce a lot of paper work as well as manpower.

**1.3 EXISTING SYSTEM**

In the existing system, only provides the information about particular hotel and only some of the hotels has possibility to reserve the rooms. The previous system was failure to publish the room’s availability of multiple hotels. And, it is failure to provide defence to admin and user information.

**1.4 PROPOSED SYSTEM**

The hotel management system provides the quality service to the end user. This project aims at creating on Hotel Management System which can be used by Admin and Customers. The admin to advise/publish the availability of rooms in different hotels and customers are checking the availability of room in required hotel. Customers should be able to know the availability of the rooms on a particular date to reserve in hotel. They should be able to reserve the available rooms according to their need in advance. To make their stay comfortable. The Admin hands the booking information of customers. The users can register and log into the system. The administrator will know the details of reservation and daily income.

**1.5 FEASIBILITY STUDY**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1) Operational Feasibility

2) Technical Feasibility

3) Economical Feasibility

**1.5.1 OPERATIONAL FEASIBILITY**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters.

**1.5.2 TECHNICAL FEASIBILITY**

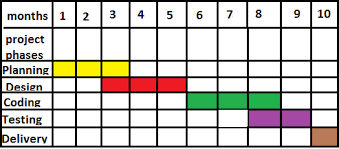
This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**1.5.3 ECONOMICAL FEASIBILITY**

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

**1.6 GIANT CHART**

Gantt charts are a great way to visualize the development process and track the progress after kickoff. The tool allows breaking a single large project into smaller manageable chunks of work, setting start and end dates.

****

**1.7 ORGANISATION OF THE REPORT**

1.7.1 INTRODUCTION

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

1.7.2 SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

1.7.3 DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

1.7.4 IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

1.7.5 RESULTS AND DISCUSSION

This section has screenshots of all the implementation i.e. user interface and their description.

1.7.6 SUMMARY AND CONCLUSION

This section has screenshots of all the implementation i.e. user interface and their description.

**CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION**

**2.1 General description :**

In this, general functions of product which includes objective of user, a user characteristic, features, benefits, about why its importance is mentioned. It also describes features of user community.

**2.2 Functional Requirements :**

In this, possible outcome of software system which includes effects due to operation of program is fully explained. All functional requirements which may include calculations, data processing, etc. are placed in a ranked order.

**2.3 Interface Requirements :**

In this, software interfaces which mean how software program communicates with each other or users either in form of any language, code, or message are fully described and explained. Examples can be shared memory, data streams, etc.

**2.4 Non-Functional Attributes :**

In this, non-functional attributes are explained that are required by software system for better performance. An example may include Security, Portability, Reliability, Reusability, Application compatibility, Data integrity, Scalability capacity, etc.

**2.5 Software Requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **Type** |
| 1 | Operating System | Windows XP / Windows |
| 2 | Language | PHP |
| 3 | Database | MySQL |
| 4 | IDE | Visual Code |
| 5 | Browser | Google Chrome |

**CHAPTER 3 : DESIGN & PLANNING**

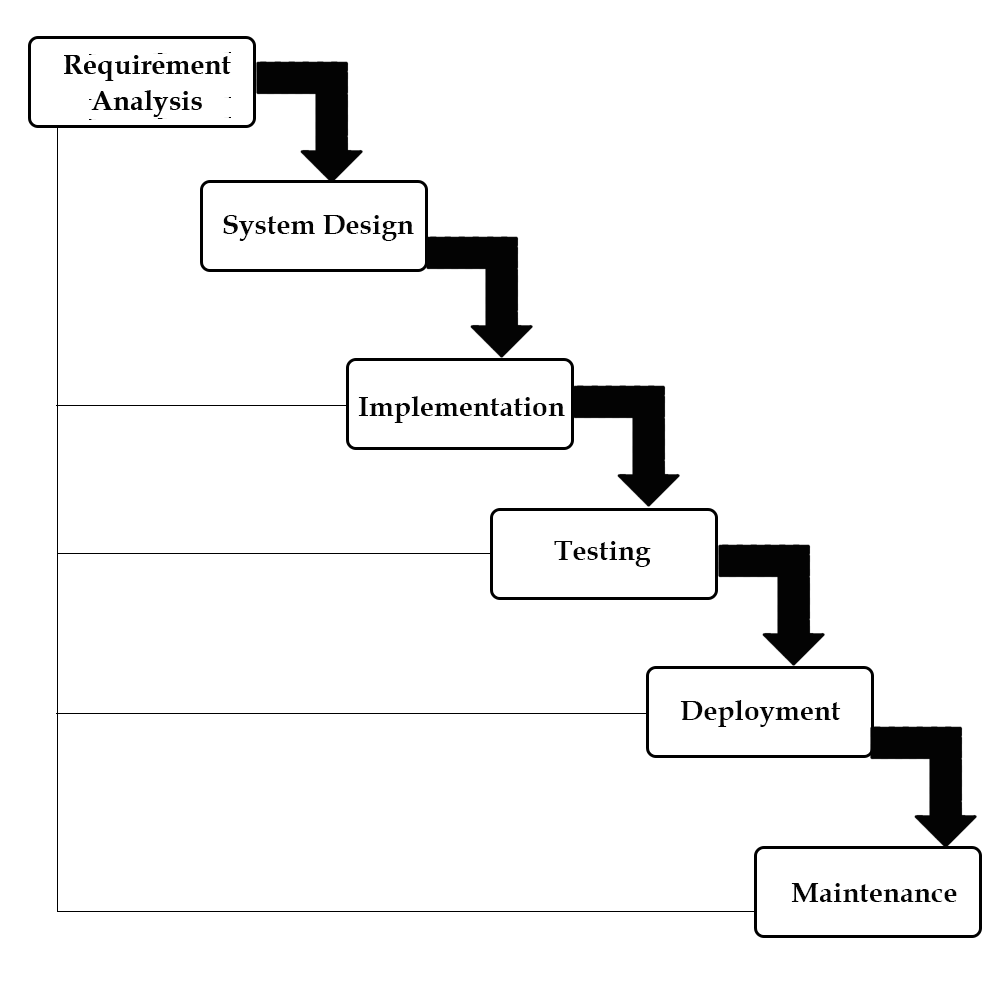
**3.1 Software Development Life Cycle Model**

A software development life cycle (SDLC) model is a conceptual framework describing all activities in a software development project from planning to maintenance. This process is associated with several models, each including a variety of tasks and activities.

**3.1.1 WATERFALL MODEL <<Sample>>**

The waterfall model was selected as the SDLC model due to the following reasons:

* Requirements were very well documented, clear and fixed.
* Technology was adequately understood.
* Simple and easy to understand and use.
* There were no ambiguous requirements.
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Clearly defined stages.
* Well understood milestones.Easy to arrange tasks.



3.2 Architecture Diagram

An architecture diagram describes what you're building, how stakeholders interact with it, and where constraints lie. A design diagram explains how to build it. ... An architecture diagram will tell you what the architect wants, plus details about the investors, the building contractors, and local laws.

**3.3 State Diagram**

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

**3.4 Collaboration Diagram**

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

**3.5 DFD Diagram**

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.

**3.6 Risk Management Plan**

A risk management plan is a document that a project manager prepares to foresee risks, estimate impacts, and define responses to risks. ... Risk management plans should be periodically reviewed by the project team to avoid having the analysis become stale and not reflective of actual potential project risks.

**CHAPTER 4 : IMPLEMENTATION DETAILS**

In this Section we will do Analysis of Technologies to use for implementing the project.

**4.1 : FRONT END**

**4.1.1 HTML    **

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**4.1.2 Css    **

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.

One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet. Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

**4.1.3 JavaScript    **

JavaScript s a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it,and major web browsers have a dedicated JavaScript engine to execute it.As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms Vanilla JavaScript and Vanilla JS refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.Google's Chrome extensions, Opera's extensions, Apple's Safari 5 extensions, Apple's Dashboard Widgets, Microsoft's Gadgets, Yahoo! Widgets, Google Desktop Gadgets, and Serence Klipfolio are implemented using JavaScript.

**4.2 : BACK END**

**4.2.1 PHP    **

PHP is a server side scripting language that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only. A PHP file contains PHP tags and ends with the extension ".php".

The term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file. The thing that differentiates PHP with client-side language like HTML is, PHP codes are executed on the server whereas HTML codes are directly rendered on the browser.

PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994.PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

**4.2.2 MySQL    **

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). It is one part of the very popular LAMP platform consisting of Linux, Apache, My SQL, and PHP. Currently My SQL is owned by Oracle. My SQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows, or Mac OS. Wikipedia and YouTube use My SQL. These sites manage millions of queries each day. My SQL comes in two versions: My SQL server system and My SQL embedded system.

RDBMS TERMINOLOGY

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

* **Database:**A database is a collection of tables, with related data.
* **Table:**A table is a matrix with data. A table in a database looks like a simple spadsheet.
* **Column:**One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row:**A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Redundancy:**Storing data twice, redundantly to make the system faster.
* **Primary Key:**A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.
* **Foreign Key:**A foreign key is the linking pin between two tables.
* **Compound Key:**A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index:**An index in a database resembles an index at the back of a book.
* **Referential Integrity:**Referential Integrity makes sure that a foreign key value always points to an existing row.

**CHAPTER 5 : TESTING AND IMPLEMENTATION**

The term implementation has different meanings ranging from the conversation of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

**5.1 : UNIT TESTING**

**5.1.1 Description**

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

**5.1.2 Test case**

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

**5.2 : INTEGRATION TESTING**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

**5.1.2 Test case**

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

**5.2.2 Test Case**

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

**5.3 : SOFTWARE VERIFICATION AND VALIDATION**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements.This is done through dynamic testing and other forms of review.Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

* Validation : Are we building the right product?
* Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

From Testing Perspective

* Fault – wrong or missing function in the code.
* Failure – the manifestation of a fault during execution.
* Malfunction – according to its specification the system does not meet its specified functionality

Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required.Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar:

* M&S Verification is the process of determining that a ⦁ computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.
* M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).

**5.3.2 Test Cases**

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

**5.4 : Black-Box Testing**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

**5.4.1 Test Procedures**

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

**5.4.2 Test Cases**

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

**5.5 : White-Box Testing**

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

**5.5.1 Levels**

**1 ) Unit testing :**White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.

**2 ) Integration testing :**White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.

**3 ) Regression testing :**White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

**5.5.2 Procedures**

White-box testing's basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

* Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.
* Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.
* Output involves preparing final report that encompasses all of the above

**CHAPTER 6 : CONCLUSION AND FUTURE WORK**

The conclusion of this project is A Hotel management system is a computerized management system. This system keeps the records of hardware assets besides software of this organization. The proposed system will keep a track of Workers, Recidents, Accounts and generation of report regarding the present status. This project has GUI based software that will help in storing, updating and retrieving the information through various user-friendly menu-driven modules.The project “Hotel Management System” is aimed to develop to maintain the day-to-day state of admission/Vacation of Residents, List of Workers , payment details etc. Main objective of this project is to provide solution for hotel to manage most there work using computerized process. This software application will help admin to handle customers information, room allocation details, payment details, billing information.etc. Detailed explanation about modules and design are provided in project documentation. The existing system is a manually maintained system. All the Hotel records are to be maintained for the details of each customers, Fee details, Room Allocation , Attendance etc. All these details are entered and retrieved manually,because of this there are many disadvantages like Time Consuming ,updating process, inaccuracy of data.For avoiding this we introduced or proposed a new system in proposed system the computerized version of the existing system. provides easy and quick access over the data.

**CHAPTER 7 : REFERENCE**

##### Reference to theses and technical reports or books should include:

last name of each author followed by their initials, year of publication (in bracket),

full title in quotes, title capitalization report number (if any) publisher or institution name, city A typical illustrative list is given below.

* 1. Ning, X., and Lovell, M. R., “On the Sliding Friction Characteristics of Unidirectional Continuous FRP Composites,” ASME J. Tribol., 124(1), pp. 5-13, 2002.
  2. Barnes, M., “Stresses in Solenoids,” J. Appl. Phys., 48(5), pp. 2000–2008, 2001.
  3. Jones, J., (2000), Contact Mechanics, Cambridge University Press, Cambridge, UK, Chap. 6.
  4. Lee, Y., Korpela, S. A., and Horne, R. N., “Structure of Multi-Cellular Natural Convection in a Tall Vertical Annulus,” Proc. 7th International Heat Transfer Conference, U. Grigul et al., eds., Hemisphere, Washington, DC, 2, pp. 221–226, 1982.
  5. Watson, D. W., “Thermodynamic Analysis,” ASME Paper No. 97-GT-288, 1997.
  6. Tung, C. Y., (1982), “Evaporative Heat Transfer in the Contact Line of a Mixture,” Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY.
  7. Kwon, O. K., and Pletcher, R. H., (1981), “Prediction of the Incompressible Flow Over A Rearward-Facing Step,” Technical Report No. HTL-26, CFD-4, Iowa State Univ., Ames, IA.
  8. Smith, R., (2002), “Conformal Lubricated Contact of Cylindrical Surfaces Involved in a Non-Steady Motion,” Ph.D. thesis, <http://www.cas.phys.unm.edu/rsmith/homepage.html>

##### General Instructions:

* + - Corrections, interlineations and crossing out of letters or words will not be permitted in any of the copies of the thesis intended for submission. Erasures, if made, should be neatly carried out in all copies.

A sub-heading at the bottom of a page must have at least two full lines below it or else it should be carried over to the next page.

The last word of any page should not be split using a hyphen.

One and a half spacing should be used for typing the general text.

The general text shall be typed in Font Style “Times New Roman” and Font Size 12.

Single spacing should be used for typing:

1. Long Tables
2. Long quotations
3. Foot notes
4. Multiline captions
5. References
   * + All quotations exceeding one line should be typed in an indented space - the indentation being 15mm from either margin.
     + The headings of all items 2 to 14 listed in section 4 should be typed in capital letters without punctuation and centered 50mm below the top of the page.
     + The text should commence 4 spaces below this heading.
     + The page numbering for all items 1 to 9 should be done using lower case Roman numerals and the pages thereafter from Chapter – 1 onwards should be numbered using Arabic numerals.
     + All page numbers (whether it is in Roman or Arabic numbers) should be typed without punctuation on the upper right hand corner 15mm from top with the last digit in line with the right hand margin.
     + The title page will be numbered as (i) but this should not be typed. The page immediately following the title page shall be numbered (ii) of consecutively thereon for the remaining pages till the main text of chapter 1. Pages of main text, starting with Chapter 1 should be consecutively numbered using Arabic numerals.
     + Any statement to be highlighted or emphasized must be done using regular bold letters. Italics or underlining is not permitted for this purpose.

Abstract: An abstract is not an introduction. It summarizes the main highlights of your thesis, including the results. The abstract should not exceed 4 pages typed with double line spacing, Font Style Times New Roman and Font Size 12 and not include any citations or figures. The abstract should briefly

**Numbering of Tables and Figures:** Tables and Figures appearing anywhere in the thesis should bear appropriate numbers. The number of the Table should be given at the top of the table with a caption. The number of the figures should be given at the bottom of the figure with caption. The rule for assigning such numbers is illustrated through an example. The fourth figure in chapter 3 will be designated as Figure 3.4. Similar rules apply for tables except that the word Figure is replaced by the word Table. If figures (or tables) appear in appendices, then figure 3 in Appendix 2 will be designated as Figure A 2.3. If a table to be continued into the next page this may be done, but no line should be drawn underneath an unfinished table. The top line of the table continued into the next page should, for example read Table 2.1 (continued) placed centrally and underlined.