THIS IS TO CERTIFY THAT THE PROJECT ENTITLED

"MariaDB Automation on Web Server"

Major Project

Submitted in partial fulfillment of the requirements for the award of the degree of 'BCA (H)'

STUDY CENTER

AKS UNIVERSITY, SATNA (M.P.)

Submitted By:

Mohammad Amir

B2092R10400028

BCA (H), 5 Sem

Submitted To

Guided By	**HOD**
Chandra Shekhar Shukla	Dr. Akhilesh A. Waoo

DEPARTMENT OF COMPUTER APPLICATION AND INFORMATIONTECHNOLOGY AND SCIENCE

AKS UNIVERSITY, SATNA (M.P.)

CERTIFICATE

This is to certify that the dissertation / project report entitle "**MariaDB Database Automation on Web Server**" submitted in partial Fulfillment of the BCA(H) to "**AKS University, Satna, (M.P.)** done by **'Mohammad Amir'** Student Code: B2092R10400028" And is an authentic work carried out by his under my guidance. Requirement for his award of the degree of BCA (H) embodies under the guidance of **Chandra Shekhar Shukla.** The matter in this project work has not been submitted earlier for award of any degree to the best of my knowledge and belief.

Signature of Internal **Signature of Guide**

Signature of External **Signature of HOD**

SELF DECLARATION

I hereby declare that the work presented in this project entitled **"MariaDB Database Automation on Web Server"** towards the partial fulfillment of the requirement for the award of the degree of **Bachelor of Computer Application (BCA)** in Department of Computer Science and Application, **AKS University, Satna, (M.P.)** is an authentic record of my own work.

I have not submitted the matter embodied in the project for the award of any other degree or diploma to any other institute or university.

Date: **Signature of Candidate**

Place: Satna

ACKNOWLEDGEMENT

It is a great for me in taking this opportunity to express my sincere thanks and ineptness to **Dr. Akhilesh A. Waoo**, Head of the Department of Computer Science, AKS University, Satna (M.P.)

I consider myself lucky enough to have such a great project. This project would add as an asset to my profile.

At this moment of accomplishment, first of all I pay homage to my guide, **Chandra Shekhar Shukla** from AKS University Satna. This work would not have been possible without his guidance, support and encouragement. Under his guidance I successfully overcame many difficulties and learned a lot.

I would like to thanks my friends who helped for the completion of this project.

I am deeply and forever indebted to my parents for their love, support and encouragement throughout my entire life.

ABSTRACT

Apache is the most commonly used Web server on Linux systems. Web servers are used to serve Web pages requested by client computers. Clients typically request and view Web pages using Web browser applications such as Firefox, Opera, Chromium, or Internet Explorer.

Users enter a Uniform Resource Locator (URL) to point to a Web server by means of its Fully Qualified Domain Name (FQDN) and a path to the required resource.

The most common protocol used to transfer Web pages is the Hyper Text Transfer Protocol (HTTP). Protocols such as Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS), and File Transfer Protocol (FTP), a protocol for uploading and downloading files, are also supported.

Apache Web Servers are often used in combination with the MySQL database engine, the Hypertext Preprocessor (PHP) scripting language, and other popular scripting languages such as Python and Perl.

This configuration is termed LAMP (Linux, Apache, MySQL and Perl/Python/PHP) and forms a powerful and robust platform for the development and deployment of Web based applications.

Index

INTRODUCTION

1. Introduction
2. Platform Specification
    - Hardware Specification
    - Software Specification

1. Introduction

Apache is the most commonly used Web server on Linux systems. Web servers are used to serve Web pages requested by client computers. Clients typically request and view Web pages using Web browser applications such as Firefox, Opera, Chromium, or Internet Explorer.

Users enter a Uniform Resource Locator (URL) to point to a Web server by means of its Fully Qualified Domain Name (FQDN) and a path to the required resource.

The most common protocol used to transfer Web pages is the Hyper Text Transfer Protocol (HTTP). Protocols such as Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS), and File Transfer Protocol (FTP), a protocol for uploading and downloading files, are also supported.

Apache Web Servers are often used in combination with the MySQL database engine, the Hypertext Preprocessor (PHP) scripting language, and other popular scripting languages such as Python and Perl.

This configuration is termed LAMP (Linux, Apache, MySQL and Perl/Python/PHP) and forms a powerful and robust platform for the development and deployment of Web based applications.

2. PLATFORM SPECIFICATION

SOFTWARE REQUIREMENT

Operating System	Linux OS
Front end	HTML, CSS, JavaScript
Database	MariaDB
Web Browser	Chrome, Opera etc.
Web Server	Mozilla Firefox, Google Chrome.
Editor	VI or VIM Editor

HARDWARE REQUIREMENT

Processor	Intel Core Due 2.0 GHz or more
RAM	512 MB or more
Hard disk	20 GB or more
Monitor	15" CRT, or LCD monitor or any
Keyboard	Normal
Mouse	Compatible Mouse

SYSTEM ANALYSIS

1. Objectives of the Project
2. Project Organization
3. Preliminary Investigation

1. OBJECTIVES OF THE PROJECT

A web server can be hardware or software that relies on the HTTP request sent by various clients connected to the network to get some information.

The primary purpose of the web servers is to provide the requested data in the form of any images, text, audio or video displayed on the web browser that several clients are accessing. Besides supporting HTTP, web servers support SMTP and FTP, mainly for transferring files, storing data, and emailing.

The transfer only happens whenever the web server's hardware is connected to the internet and helps exchange the data between various devices. The web server's job is to manage what content is accessed by the user. There are multiple filters and limitations to access the data on the internet. You can access the data if you are an authorized user of a particular server. The web server works on the client-server model for displaying the information. Any website must have the web server software for hosting sites.

2. PROJECT ORGANIZATION:-

The Waterfall model

Winston Royce introduced the Waterfall Model in 1970.This model has five phases: Requirements analysis and specification, design, implementation, and unit testing, integration and system testing, and operation and maintenance. The steps always follow in this order and do not overlap. The developer must complete every phase before the next phase begins. This model is named "Waterfall Model", because its diagrammatic representation resembles a cascade of waterfalls.

Analysis:-

1. Requirements analysis and specification phase:

The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirement of the software. It describes the "what" of the system to be produced and not how. In this phase, a large document called Software Requirement Specification (SRS) document is created which contained a detailed description of what the system will do in the common language.

2. Design Phase:

This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

3. Implementation and unit testing:

During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD.

During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.

4. Integration and System Testing:

This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers,

lower maintenance costs, and accurate results. Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.

5. Operation and maintenance phase:

Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

3. PRELIMINARY INVESTIGATION:-

IDENTIFICATION OF NEED:-

1) Review Of Written Method
2) Onsite Observation
3) Interview
4) Questionnaires

1) Review of Written Method

In this fact-finding method, all documentation on data carried is organized and evaluated. The analyst needs find out how they are filled out, how they are easy to user, what changes need to make and how easy they are to be read.

2) Onsite Observation

Other fact-finding method used by the system is onsite direct observation .The analyst role is that of information seeker. One purpose of onsite observation is to study the real system. As an observer, is to get close as possible to the real system being studied.

As an observer, the analyst follows sets of rules. The analyst observes the physical layout of the current system, the location and movement of the people, and the workflow.

3) Interview

In an interview, since the analyst and the person interview meet face to face, there is an opportunity for greater flexibility in eliciting information. The

Interviewer is also in a natural position to observe the subjects and the situation to they are responding.

4) Questionnaires

This fact-finding method is used when the staff is located over a wide geographical area. The information obtained through the questionnaires is limited to the written response of the subject to predefined questions.

FEASIBILITY STUDY

1. Feasibility

    - Operational Feasibility
    - Technical Feasibility
    - Economical Feasibility

1. FEASIBILITY

Feasibility is the measure of how beneficial or practical the development of an information system will be to an organization. Unfortunately, development of a computer-based system is plagued by scarcity of resources, and limited time constraints.

Feasibility analysis is the process by which feasibility is measured. The Feasibility analysis in a project that is feasible at one point in time may become infeasible at a later point in time. These identify specific times during the life cycle when feasibility is reevaluated and management review should be conducted at the end of the prior phase. A project can be canceled or revised in scope, schedule, or budget at any of these checkpoints.

Most analysts agree that there are three categories of feasibility test.

- Economic
- Technical
- Behavioral

ECONOMIC FEASIBILITY:-

This feasibility is the measure of the cost effectiveness of a project or solution. This is often called a cost-benefit analysis

There must have sufficient benefits in creating the system to make the cost acceptable. A **"MariaDB Database Automation on Web Server"** can be developed technically and that will be used if installed must still be a good investment for the organization. Financial benefits must equal or exceed the costs. The financial and economic questions raised by analyst during the preliminary investigation are for the purpose of estimating the following:-

- The cost to conduct a full system investigation.
- The cost of hardware and software.
- The cost if nothing changes.
- The cost can also be calculated on the basis of Line of Code (LOC).

TECHNICAL FEASIBILITY

This feasibility is the measure of the practicality of a specific technical solution and the availability of the technical resources and expertise.

Technical feasibility focuses on the existing computer hardware and software, as to what extent it can support the proposed system. It deals with the availability of the required technology for implementing the proposed system. The ***"MariaDB Database Automation on Web Server"*** is developed using HTML, CSS, JavaScript and MariaDB.

BEHAVIORAL FEASIBILITY

This is the measure of how well the solution will work in the organization. It is also a measure if how people feel about the system/project.

The ***"MariaDB Database Automation on Web Server"*** system follows Behavioral Feasibility because of its friendliness in nature. Anyone can operate easily, for this we have developed user interface and user friendly system. We have provided maximum information and instruction to the user going to use the system.

REQUIREMENTANALYSIS

1. Existing System

2. Proposed System

1. EXISTING SYSTEM:

- This existing system is not provides multiple web hosting.
- This manual system gives us very less security for saving data and some data may be lost due to mismanagement.
- More Expensive.

2. PROPOSED SYSTEM:

A web server can be hardware or software that relies on the HTTP request sent by various clients connected to the network to get some information.

The primary purpose of the web servers is to provide the requested data in the form of any images, text, audio or video displayed on the web browser that several clients are accessing. Besides supporting HTTP, web servers support SMTP and FTP, mainly for transferring files, storing data, and emailing.

The transfer only happens whenever the web server's hardware is connected to the internet and helps exchange the data between various devices. The web server's job is to manage what content is accessed by the user. There are multiple filters and limitations to access the data on the internet. You can access the data if you are an authorized user of a particular server. The web server works on the client-server model for displaying the information. Any website must have the web server software for hosting sites.

SYSTEM DESIGN

1. System Design
    - DFD Diagram
    - E-R Diagram
2. UML View
    - Use Case Diagram

3 Database Description

1. SYSTEM DESIGN

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system. Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. System design in terms of software engineering has its own value and importance in the system development process as a whole. To mention it may though seem as simple as anything or simply the design of systems, button a broader sense it implies a systematic and rigorous approach to design such a system which fulfills all the practical aspects including flexibility, efficiency and security.

DATA FLOW DIAGRAM (DFD)

As information moves through software, it is modified by a series of transformation. A data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or a bubble.

The data flow diagram may be used to represent a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling.

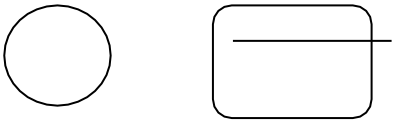
Data Flow Diagrams Notations

- **Process:** Any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.
- **Data store:** Files or repositories that hold information for later use, such as a database table or a membership form.
- **Dataflow:** Dataflow are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.
- **External Entity:** External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.

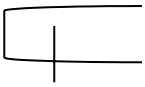
Following symbols are used for process, entity, database and dataflow.

External Entity

Process



Data Store



Data Flow

Data Flow Diagram Levels

❖ Context Diagram

A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities.

❖ DFD Layers

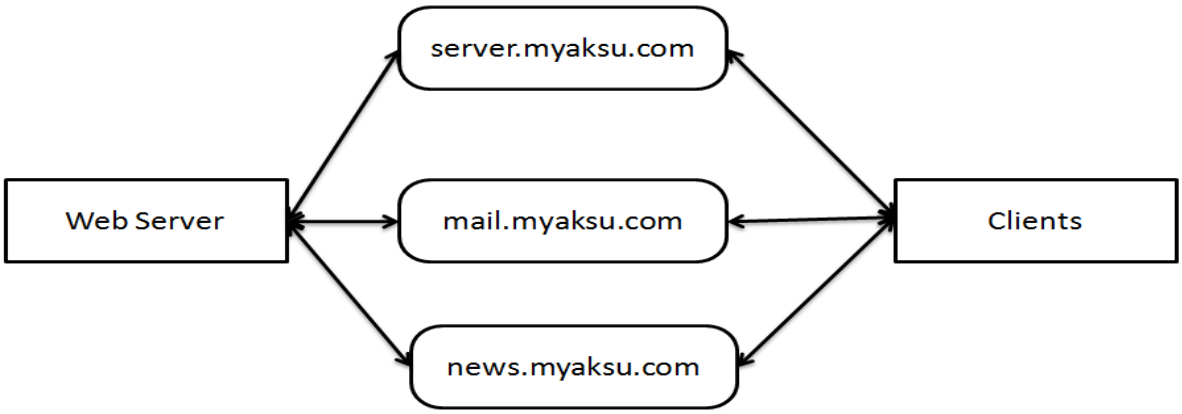
Draw data flow diagrams can be made in several nested layers. A single process node on a high level diagram can be expanded to show a more detailed data flow diagram. Draw the context diagram first, followed by various layers of data flow diagrams.

❖ DFD Levels

The first level DFD shows the main processes within the system. Each of these processes can be broken into further processes until you reach pseudo code.

0-level DFD Diagram

The following shows the 0-Level Data flow diagram.



2. E-R DIAGRAM

What is the ER Model?

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them.

Entity Relationship Diagrams are a major data-modeling tool and help organize the data in the project into entities and define the relationships between the entities. This process has proved to enable the analyst to produce a good database structure so that the data can be stored and retrieved in a most efficient manner. An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

Components of the ER Diagram

This model is based on three basic concepts:

- Entities
- Attributes
- Relationships

Entity -

An entity can be a person, place, event, or object that is relevant to a given system. For example, a school system may include students, teachers, major courses, subjects, fees, and other items. Entities are represented in ER diagrams

by a rectangle and named using singular nouns.

Entity set:

An entity set is a group of similar kind of entities. It may contain entities with attribute sharing similar values.

Types of Entities

- Strong Entity
- Weak Entity

Attribute -

An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. For example, the attribute Inventory Item Name is an attribute of the entity Inventory Item. An entity can have as many attributes as necessary. Meanwhile, attributes can also have their own specific attributes. Note that some top level ER diagrams do not show attributes for the sake of simplicity. In those that do, however, attributes are represented by oval shapes.

Types of Attributes

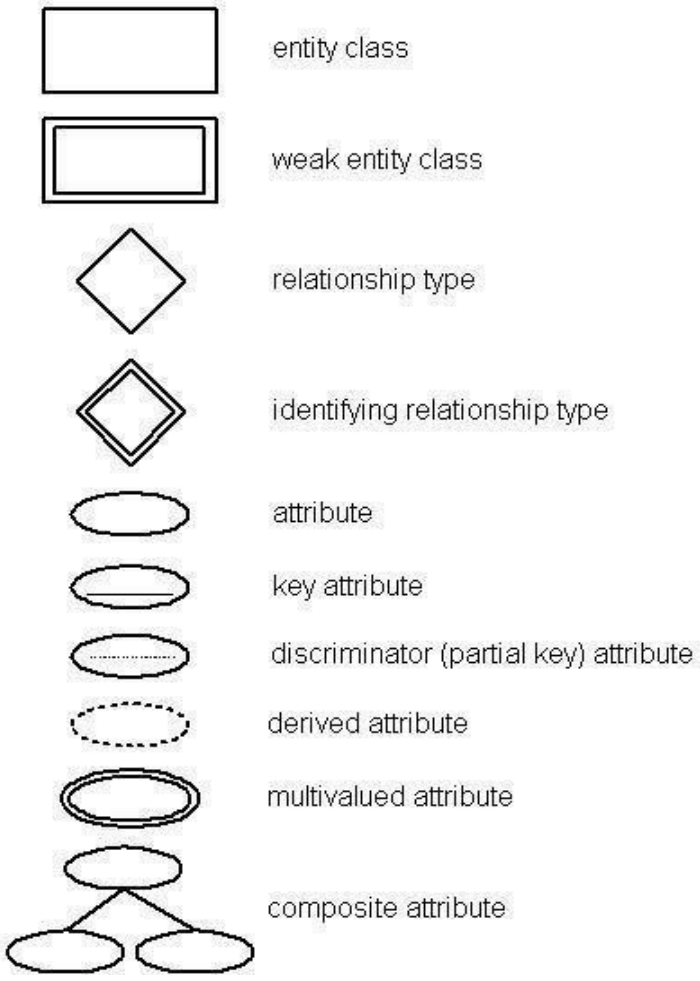
- Simple attribute
- Composite attribute
- Derived attribute

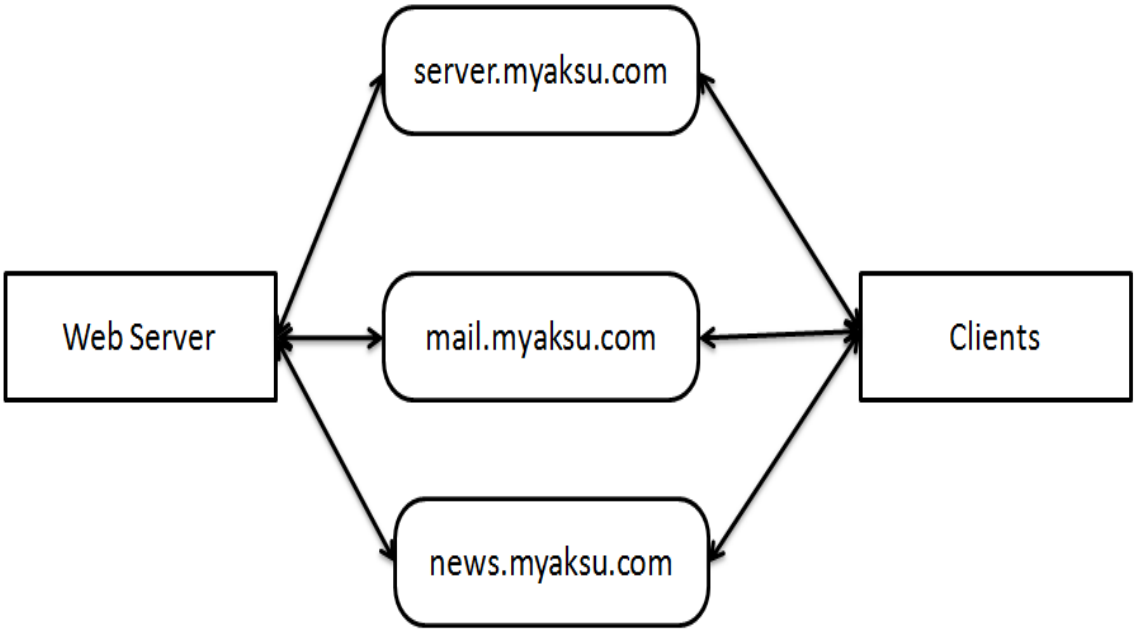
Relationship –: A relationship describes how entities interact.

Types of Relationships

- One-to-One Relationships
- One-to-Many Relationships
- May to One Relationships
- Many-to-Many Relationships

Symbols of ER Diagram




server.myaksu.com
Web Server
mail.myaksu.com
Clients
news.myaksu.com

2. UML View

UML is an acronym that stands for Unified Modeling Language**.** Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques. It is based on diagrammatic representations of software components. As the old proverb says "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes. UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates.

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

You can also create your own set of diagrams to meet your requirements. Diagrams are generally made in an incremental and iterative way.

There are two broad categories of diagrams and they are again divided into subcategories –

- Structural Diagrams
- Behavioral Diagrams

Structural UML diagrams

- Class diagram
- Package diagram
- Object diagram
- Component diagram
- Composite structure diagram
- Deployment diagram

Behavioral UML diagrams

- Activity diagram
- Sequence diagram
- Use case diagram
- State diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

USE CASE DIAGRAM-

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as **actors**.

Use Case Diagram Notations

- Actor
- Use Case
- Relationship
- System Boundary

Actor:

Actors are usually individuals involved with the system defined according to their roles. The actor can be a human or other external system.

Use Case:

A use case describes how actors use a system to accomplish a particular goal. Use cases are typically initiated by a user to fulfill goals describing the activities and variants involved in attaining the goal.

Relationship:

The relationships between the actors and the use cases.

System Boundary:

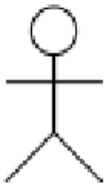
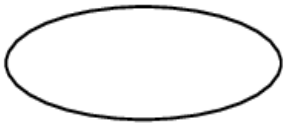
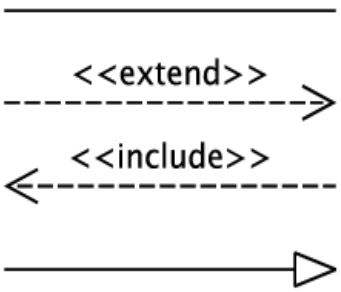
The system boundary defines the system of interest in relation to the world around it.

How to Draw a Use Case Diagram

A Use Case model can be developed by following the steps below.

- Identify the Actors (role of users) of the system.
- For each category of users, identify all roles played by the users relevant to the system.

- Identify what are the users required the system to be performed to achieve these goals.
- Create use cases for every goal.
- Structure the use cases.

Symbol	Reference Name
	Actor
	Use case
	Relationship

INPUT OUTPUT SCREEN

2. CODING –

FRONT END : HTML, CSS, JavaScript

Database : MariaDB

TESTING

Testing Objectives

1. Testing Methodology
2. Quality Attributes
3. Characteristics of Language used

1. TESTING OBJECTIVES

The completion of a system is achieved only it has been thoughly tested. Thought this gives a feel the project is completed, there cannot be any project without going through this stage. Hence is this stage it is decided whether the project can undergo the real time environment execution without any break down, therefore a package can be rejected even at this stage.

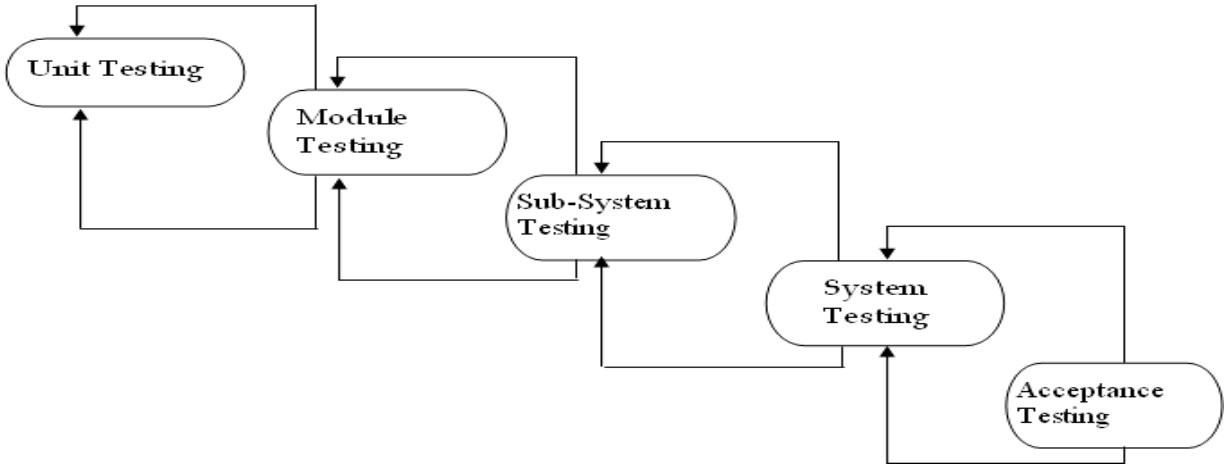
The primary objectives of software testing are as follows:

1. Testing is a process of executing a program to find an error in it.
2. A good test case should have a high probability of finding an as-yet-undiscovered error.
3. A test case will be considered successful if it uncovers an as-yet-undiscovered error.

3. 1.1 Testing Principles

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- The Pareto principle applies to software testing.
- Testing should begin "in the small" and progress towards testing "in the large".
- Exhaustive testing is not possible.
- To be most effective, testing should be conducted by in independent third party.

3. 2. TESTING METHODOLOGIES



5. **Unit Testing**

Unit testing aims the verification effort on the smallest unit of software design i.e., a software component or module. It uses procedural design as a guide to test major control paths and uncover errors within the module boundary. It is white box oriented and the step can be conducted in parallel for multiple components.

Unit testing is a dynamic method for verification, where the program is actually compiled and executed. It is one of the most widely used methods, and the coding phase is sometimes called “coding and unit testing phase”. The goal of unit testing is to test modules or “units”, not the whole software system. Unit testing is most often done by the programmer himself/herself. The goal of unit testing is to isolate each part of the program and show that the individual parts

are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

- **Integration Testing:-**

Integration testing is a phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing. The major objective of integration testing is to tackle the problem of interfacing i.e. putting all the acceptable imprecision (view) may be magnified to unacceptable levels; global data structure can cause problems and to truncate this list of problems we use integration testing.

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by the design.

Integration testing strategy used is Bottom-Up Integration Testing. In it all the bottom or low level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.

❖ Common Quality Attributes

1. Conceptual Integrity:- Conceptual integrity defines the consistency and coherence of the overall design.

2. Maintainability:- Maintainability is the ability of the system to undergo changes with a degree of ease.

3. Reusability:- Reusability defines the capability for components and subsystems to be suitable for use in other applications and in other scenarios.

4. Availability:- Availability defines the proportion of time that the system is functional and working. It can be measured as a percentage of the total system downtime over a predefined period.

5. Inter compatibility:- Interoperability is the ability of a system or different systems to operate successfully by communicating and exchanging information with other external systems.

6. Manageability:- Manageability defines how easy it is for system administrators to manage the application, usually through sufficient and useful instrumentation.

7. Performance:- Performance is an indication of the responsiveness of a system to execute any action within a given time interval.

8. Security:- Security is the capability of a system to prevent malicious or accidental actions outside of the designed usage, and to prevent disclosure or loss of information.

   A crucial phase in the system life cycle is the successful implementation of the new system design. Implementation simply means converting a new system design into operation. This involves creating computer compatible files, training the operating staff and installing hardware terminals, and telecommunication network before the system is up and running.

   In system implementation, user training is crucial for minimizing resistance to change and giving the new system a chance to prove its worth. Training aids such as user-friendly manuals, a data dictionary and job performance aids that communicate information about the new system and help screens. Provide the user with a good start on the new system.

4. CHARACTERISTICS OF LANGUAGE USED

There were five primary characteristics in the creation of the Java language:

1. It should be "simple, object-oriented and familiar".
2. It should be "robust and secure".
3. It should be "portable".
4. It should execute with "high performance".
5. It should be "interpreted, threaded, and dynamic".

CONCLUSION &DISCUSSION

1. Limitations
2. Future Enhancement
3. Conclusion

1. LIMITATION

- The size of the database increases day-by-day, increasing the load on the database and data maintenance activity.

2. FUTURE ENHANCEMENT

- It satisfies the user requirement.
- Be easy to understand by the user and operator.
- Be easy to operate.
- Have a good user interface.
- Be expandable.

3. CONCULSION

Web server helps in supporting large data storage, thus handling various websites. It also helps manage the bandwidth that regulates the incoming traffic to the web server avoiding downtime for any website. You can even create FTP websites as web servers help move the data from one site to another without any effort. It helps manage the download speed for any web-based application and improves performance. It is secured as they are stored in protective infrastructure. During high traffic, there may be a server crash situation.

With web servers, you will be able to host the websites on the internet, making them accessible to various users connected to the internet. It also allows you to communicate between the web clients and the servers within the local network area.

BIBLIOGRAPHY &REFERENCES

- Reference Books
- Reference Websites

1. REFERENCE BOOKS

1. MJ Bach “Design of Unix OS”
2. Y Kanetkar “Unix shell programming” BPB Pub

2. REFERENCE WEBSITES

1. https://www.javatpoint.com/linux-tutorial
2. https://www.tutorialspoint.com/unix/index.htm
3. https://www.guru99.com/introduction-linux.html
4. https://www.geeksforgeeks.org/introduction-to-linux-operating-system/
5. https://www.youtube.com/watch?v=_JheM-2_0kg&list=PLPTwEjck6TZ1wL0h-jOPLamWA-VTtdXHj