



[Home](#) » [COMPUTER SCIENCE](#) • [GENERAL TECHNICAL PAPERS](#) • [PROJECTS](#) » Software Requirements Specification Report for a Project

Software Requirements Specification Report for a Project

By Ravi Bandakkanavar | June 12, 2018

35 Comments

A Software requirements specification (SRS) document describes the intended purpose, requirements, and nature of software/application/project to be developed.

To prepare an SRS document, you would need to have a functional knowledge of your project or application, knowledge of software/hardware/technology to be used.

Generally, the report is prepared with the following format.

INDEX

1.Introduction

1.1 Purpose

1.2 Need/Motivation

2.Literature survey

3.Requirements.

3.1 Functional Requirements

3.2 Non- Functional Requirements

3.2.1 Safety Requirements.

3.2.2 Security Requirements

3.2.3 Software Quality Attributes

- 3.3 Hardware Requirements
- 3.4 Software Requirements
- 3.5 WaterFall Model
- 3.6 Feasibility Study
 - 3.6.1 Economic Feasibility
 - 3.6.2 Technical Feasibility
 - 3.6.3 Operational Feasibility
- 4. System Architecture
 - 4.1 Client-Server Architecture
- 5. Design and Implementation
 - 5.1 Product Features
 - 5.2 class diagram design
 - 5.3 Use case diagram
 - 5.4 Sequence diagram
 - 5.5 E-R Diagram and Normalisation
 - 5.5.1 E-R Diagram
 - 5.5.2 Normalisation
- 6. Snapshots
- 7. Testing And Results
 - 7.1 Unit Testing
 - 7.2 Black Box Testing.
 - 7.3 White Box Testing
 - 7.4 Integration Testing
 - 7.5 Validation Testing
 - 7.6 Acceptance Testing
- 8. Conclusion
- 9. Bibliography

The following software requirements specification report has been prepared for a project named **eAdministration of computer labs.**

Suggested Read:

- [*Software Requirements Specification document with example*](#)

Chapter 1:

INTRODUCTION

The project e-Administration of computer labs is an automated system for lab management. It eases the tasks of the administrator, instructors, head of the department of various departments, technical staff and students.

The main Objective of this project is to provide the solution for an organization which involve computing facilities such as Computers, Printers, LCDs, Digital Boards, ACs etc. on a 24×7 basis to carry out academic experiments for all the branches with all the equipment in good working conditions and also with all the necessary software installed and configured. This software also helps the administrator and instructor(s) to maintain proper documentation of the computing systems.

This software is a web-based application and can be hosted on the Internet. It also provides a clean and user-friendly interface to the users.

1.1 Purpose

The purpose of this project is to provide the solution for an organization to help in monitoring working condition of computing facilities such as Computers, Printers, LCDs, Digital Boards, ACs etc. on a 24×7 basis.

1.2 Need/Motivation

Our college has 10 computer labs. Each lab is managed by an instructor. It is difficult for the administrator and newly hired instructor to know the entire information (such as a number of computers, software installed in each system, vendors etc.) of each individual lab. It is also difficult for an administrator to integrate the entire information of all labs in college. Our software solves these problems.

Chapter 2:

LITERATURE SURVEY

The e-Administration of Computer Labs is a new attempt to speed up the process of managing Computer Labs in an educational institute. The existing systems are time-consuming and there are many difficulties faced by the administrator to get information about each lab within the organization. Presently in labs, most of the tasks are carried on manually such as lodging complaints, extra lab requests etc. There are many difficulties faced by instructors, HODs of various departments for carrying out any lab related activities.



This software provides a solution to these problems. It provides an interactive user interface and helps users in an organization to get information immediately at that instant of time. The different users of the system are the administrator, the heads of various department, technical staff and lab instructor. These users are assigned with different privileges based on the level of administration.

This system helps the technical staff, instructor, and head of various departments to register a specific complaint and to view the status of their complaint. Here the instructor gets SMS on expiry of specific software. The head of the department can send the extra lab request and can also view complaints associated with the respective department. This software establishes a smooth communication between different users and user admin.

Objectives

- Helps the administrator and instructor to keep track of the detailed information of the labs.
- Assists in the smooth interaction between different users.
- Proper maintenance of available Resources.
- Speed up the Activities to overcome the problems associated with the lab.
- Helps Technical staff/instructor to lodge complaints.
- Automate the task of assigning complaints to specific technical staff based on the nature of the problem.
- Helps the HODs of various departments to send an extra lab request to the administrator.

Chapter 3:

REQUIREMENTS

3.1 Functional Requirements

- It should provide schedule/timetable without any of clashes among faculties, day, time and room that must be visible to all.

- It should generate a report about the registered complaint to the admin and response report to the user who has submitted his queries.
- Secure registration and profile management facilities for different users.
- It should provide details like e-learning facilities, server room details, software installation procedures etc. to students.
- It should generate alerts via SMS.

3.2 Non- Functional Requirements

3.2.1 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

3.2.2 Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

3.2.3 Software Quality Attributes

- **AVAILABILITY:** Since we are hosting our project on the server it will be available all the time.
- **CORRECTNESS:** The system should generate an appropriate report about different activities of the lab and should keep track of all records.
- **MAINTAINABILITY:** The system should maintain correct schedules of labs and the documentation of all the lab equipment.
- **USABILITY:** The system should satisfy the maximum number of users needs.

3.3 Hardware Requirements

- Pentium IV or higher, (PIV-300GHz recommended)
- 256 MB RAM
- 1 Gb hard free drive space

3.4 Software Requirements

- PHP (front end)
- HTML

- JavaScript
- MS Word 97 or later
- Web Browser: Microsoft Internet Explorer, Mozilla, Google Chrome or later
- MySQL Server (back-end)
- Operating System: Windows XP / Windows7/ Windows Vista

3.5 Waterfall Model

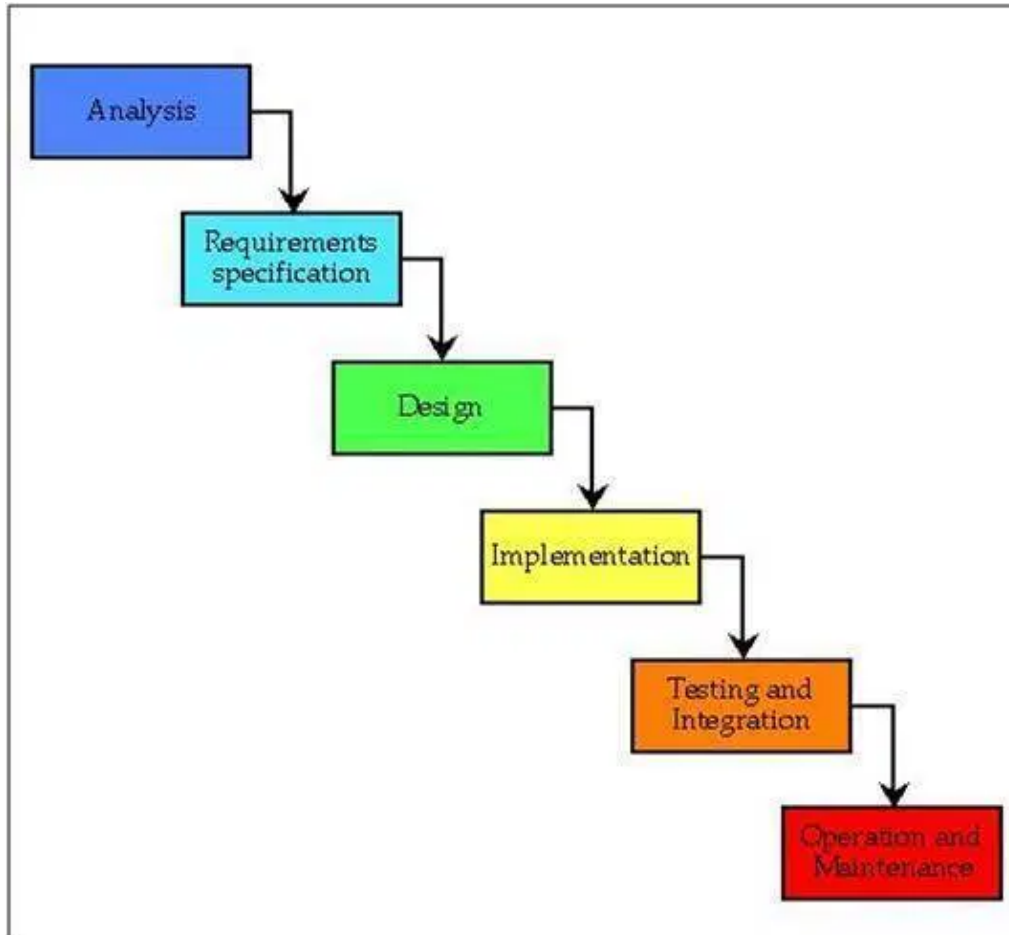


Fig. 3.1 Waterfall Model

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Analysis, Requirement Specification, Design, Implementation, Testing and Integration, and Operation and Maintenance.

If at the beginning of the project failures are detected, it takes less effort (and therefore time and money) for this error. In the waterfall model phases to be properly sealed first before proceeding to the next stage. It is believed that the phases are correct before proceeding to the next phase. In the waterfall model lay the emphasis on documentation. It is a straightforward method. The way of working ensures that there are specific phases. This tells you what stage it is. One can use this method of milestones. Milestones can be used to monitor the progress of the project to estimate.

In our Project, all the requirements are clear and well known and the project is large. All the activities in our project are carried out in the above-mentioned phases of the waterfall model.

3.6 Feasibility Study

The prime focus of the feasibility is evaluating the practicality of the proposed system keeping in mind a number of factors. The following factors are taken into account before deciding in favor of the new system.

3.6.1 Economic Feasibility

Report generation in the proposed system in precise that is reports are generated as per user requirements, which reduces the use of papers and manual labor.

3.6.2 Technical feasibility

Keeping in view the above fact, nowadays all organizations are automating the repetitive and monotonous works done by humans. The key process areas of the current system are nicely amenable to automation and hence the technical feasibility is proved beyond doubt.

3.6.3 Operational Feasibility

The present system has automated most of the manual tasks. Therefore the proposed system will increase the operational efficiency of the administrator and instructors.

Chapter 4:

System Architecture

4.1 Clint-Server Architecture

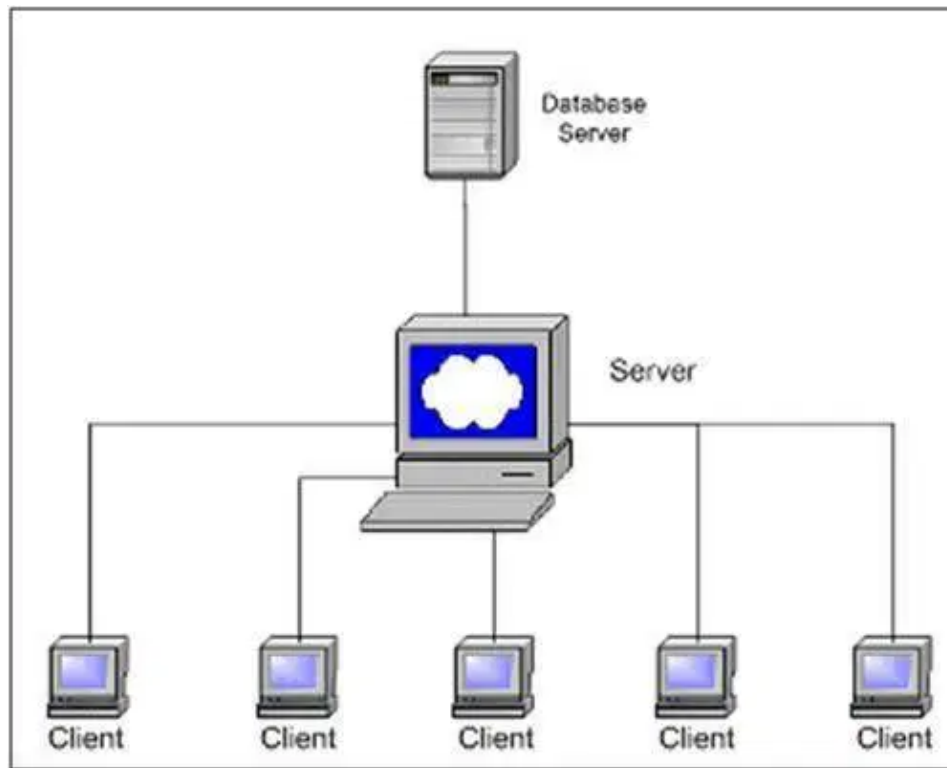


Fig. 4.1 Client-Server Architecture

The client-server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients.

Client-server software architecture is versatile and flexible in today's fast-changing IT landscape. It is modular in structure and relies on messaging services for communication between components. They were designed to improve flexibility, usability, scalability, and interoperability. Software flexibility implies the ability for a program to change easily according to different users and different system requirements.

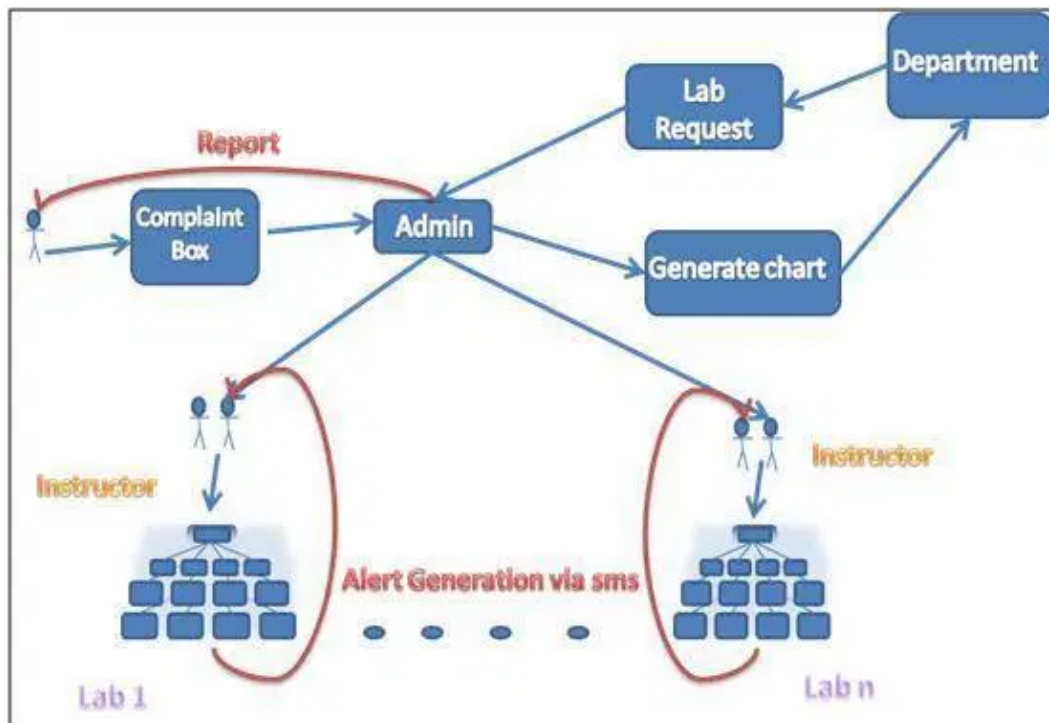


Fig. 4.2 System Architecture of e-Administration of Computer Labs

The admin creates different users. Admin can generate lab assignment and assigns an instructor to a specific complaint or any problem and checks the report to ensure whether the complaints responded or pending.

Users include Technical staff and Instructors. The user can register complaint pertaining to any lab equipment. The registered complaints will be sent to Admin. The instructor maintains documentation of software, hardware and he also provides the report of equipment conditions to the admin. He performs the tasks assigned by admin. He gets alerted via SMS about the warranty of the product if it is out of date.

Another profile in our project is the HODs profile. Here HODs can see the complaints lodged by his/her department users. HOD can also request for extra labs to the Admin. Based on the available slots, admin defines the slots and send a reply to HOD.

This software also facilitates students in getting syllabus copy, software installation procedures, different lab related software etc.

Chapter 5:

Design and Implementation

5.1 Product Features

The major features of e-administration of computer lab system are as listed below.

- Unauthorized access is prevented. Because only authorized user can address the complaints and access the resources.
- It helps the administrator to keep track of the detailed information of the labs such as maintaining each user's profile, the number of available resources etc.
- It helps instructors to dump the software into the server to facilitate students to take the required software.
- It increases the availability of lab resources by their proper maintenance.
- A smooth communication link establishes between different users and user admin.

By achieving the above features, Responsiveness and hence outcomes of each lab increases.

5.2 class diagram design

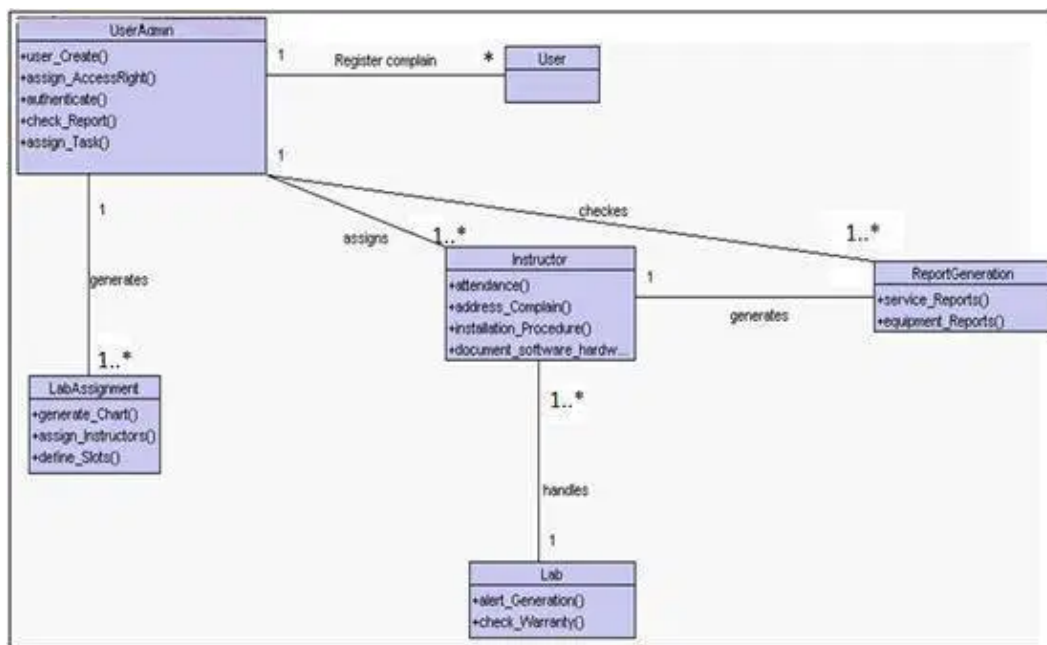


Fig. 5.1 Class diagram

Modules

1. Role of admin

Creates a user, assigns access rights, authenticates users, checks report, and assigns the task. These functions are used to keep track of various activities in labs.

2. Service complaint

It includes complaint registration, assigning a task to the instructor, addressing complain. These functions may be used to register a complaint about improper conditions of devices (in the lab) to admin in order to get response/service.

3. Report generation

This module generates a service report and reports for the working condition of the equipment in the lab.

4. Software or hardware maintenance

This module checks warranty of particular software generates alert messages and helps in proper documentation of available software or hardware in labs.

5. Lab Assignment:

This module defines slots, generates charts, assigns instructors, and would specify alternate instructor to particular labs.

5.3 Use case diagram

a) Interaction between user and administrator

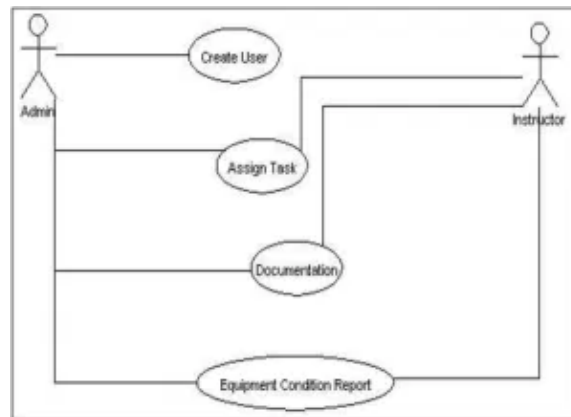
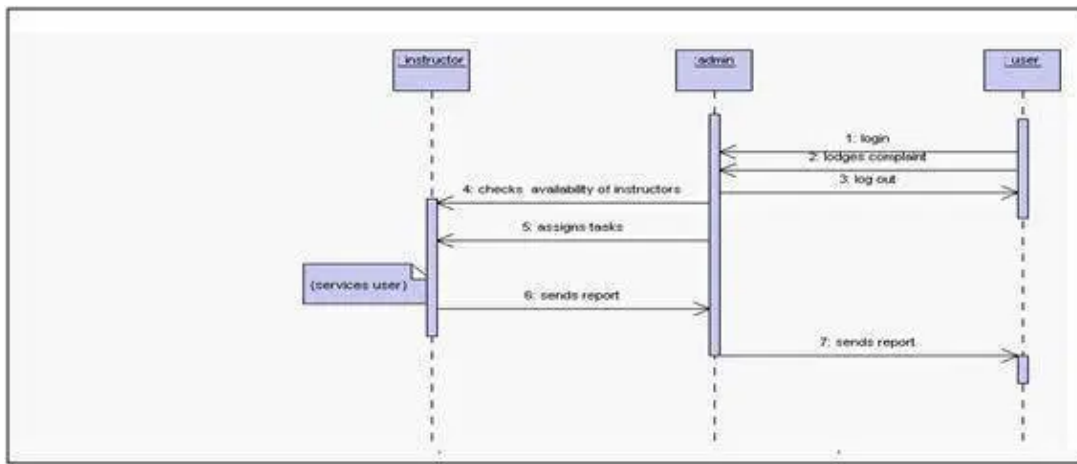


Fig. 5.2

Admin Creates a user, checks report and assigns a task to the user. These functions are used to keep track of various activities in labs.

5.4 Sequence diagram



Sequence diagram for lodging a complaint

5.5 E-R Diagram and Normalization

5.5.1 E-R DIAGRAM

- E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.
- **ENTITIES:** Which specify distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
- **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

Here the Entities are LAB, EMPLOYEE, COMPLAINT, DEPARTMENT, and EQUIPMENT.

5.5.2 NORMALIZATION

- The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and an increase in the total size of the data stored.
- If a Database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.
- Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most

practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications. Here We have Normalized up to 3NF.

Chapter 7:

Snapshots

a) Homepage



This is the home page of e-Administration of computer labs. Through this page, students, tech staff, and nontech staff can access information pertaining to different labs.

Chapter 8:

TESTING AND RESULT

The reason behind testing was to find errors. Every program or software has errors in it, against the common view that there are no errors in it if the program or software is working. Executing the programs with the intention of finding the errors in it is therefore testing; hence a successful test is one which finds errors. Testing is an activity, however, it is restricted to being performed after the development phase is complete, but is carried parallel with all stages of system development, starting with requirement specification.

Test cases were devised with a purpose in mind. A test case is a set of data that a system will process as normal input. The software units developed in the system are modules and routines that are assembled and integrated to perform the required function of the system. Test results once gathered and evaluated, provide a qualitative indication of the software quality and reliability and serve as the basis for design modification if required. In this phase, testing is done at different levels. Actually testing phase of the implementations works accurately and efficiently before live operation commences.

7.1 Unit Testing

Unit testing was done after the coding phase. The purpose of the unit testing was to locate errors in the current module, independent of the other modules. Some changes in the coding were done during the testing phase. Finally, all the modules were individually tested following bottom to top approach, starting with the smallest and lowest modules and then testing one at a time.

7.2 Black Box Testing

This method of software testing tests the functionality of an application as opposed to its internal structures or working(i.e. white box testing). Specific knowledge of the application's code/internal structure and programming knowledge, in general, is not required. Test cases are built to specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and design to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object's internal structure.

7.3 White Box Testing

This method of software testing 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 required and used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

7.4 Integration Testing

Once the unit was over, all the modules were integrated for integration testing. External and internal interfaces are implemented and work as per design, the performance of the module is not degraded.

7.5 Validation Testing

At the culmination of integration testing, the software is said to be completely assembled as a package; interfacing errors have been uncovered and corrected. Then as a final series of software

test, validation tests were carried out.

7.6 Acceptance Testing

This is the final stage in the testing process before the system is accepted for operational use. Any requirement problem or requirement definition problem revealed from acceptance testing are considered and made error free.

Chapter 9:

CONCLUSION

e-Administration of computer labs is an automated system for lab management. It eases the tasks of the administrator, instructors, technical staff, HODs of various departments and students.

The administrator creates different users and assigns specific privileges to users. He is responsible for handling most of the tasks such as extra lab requests, assigning tasks to instructors, etc. Here the user lodges a complaint about improper functioning of the lab equipment. This complaint will be automatically assigned to technical staff based on the nature of the problem.

The instructor maintains the documentation of software, hardware and also reports the condition of equipment to the admin. He also gets alerts via SMS about the warranty or expiry of the product, if it is outdated.

HODs of the various departments can also send a request for the extra lab. Based on the available slots, admin defines the slots and send a reply to HOD. This software also facilitates students in getting syllabus copy, software installation procedures, different lab related software, etc.

BIBLIOGRAPHY

Books:

- 1) Programming the World Wide Web, 4th Edition, by Robert Sebesta
- 2) PHP 6 and MySQL 5 for Dynamic Web Sites **by** Larry Ullman
- 3) **Learn PHP, MySQL, and JavaScript** by Robin Nixon
- 4) Programming PHP by Rasmus Lerdorf, Kevin Tatroe, and Peter MacIntyre
- 5) Advanced Javascript by Chuck Easttom
- 6) JavaScript, A Beginner's Guide by John Pollock

[Facebook](#)[Twitter](#)[Tumblr](#)[Pinterest](#)[WhatsApp](#)[More](#) 115