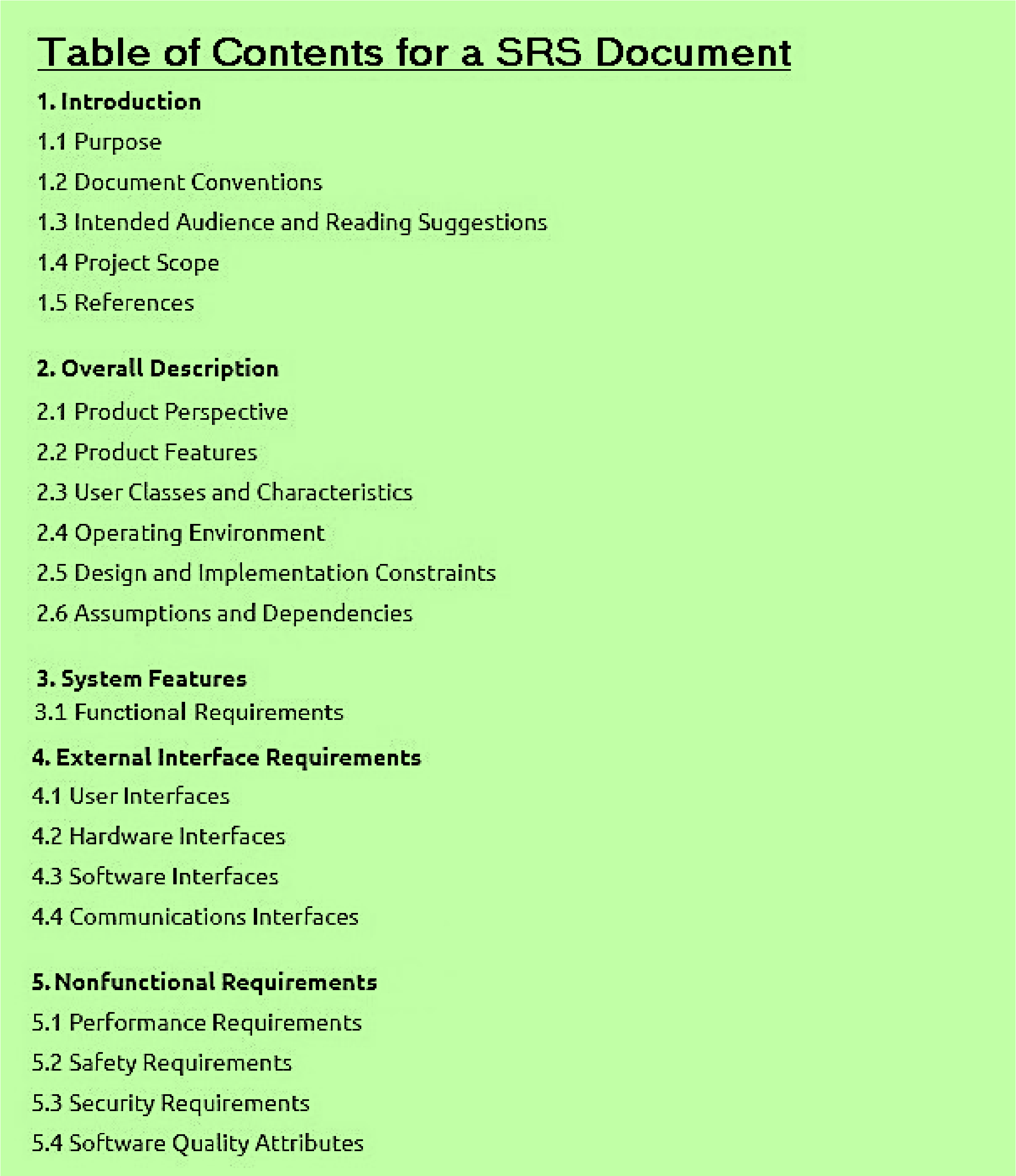


Table of Contents for a SRS Document

1. Introduction

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Project Scope

1.5 References

2. Overall Description

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

3. System Features

3.1 Functional Requirements

4. External Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

5. Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

REQUIREMENTS

(https://krazytech.com/projects/software-requirements-specification-report)

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 Process Model: Waterfall Model

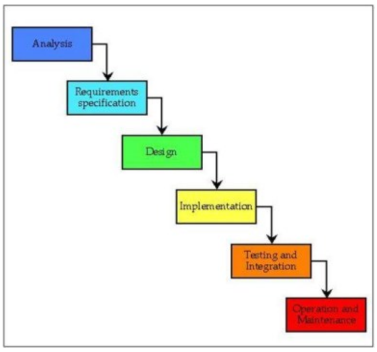


Fig. 3.1 Waterfall Model

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 favour 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 labour.

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.

System Requirements : Architecture

4.1 Clint-Server Architecture

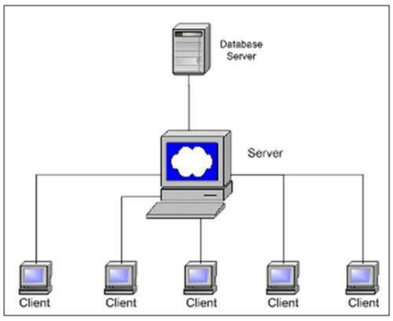


Fig. 4.1 Client-Server Architecture

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.

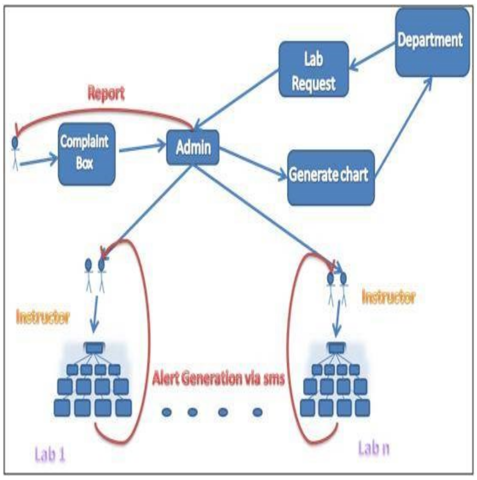


Fig. 4.2 System Architecture of e-Administration of Computer Labs

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.

Design Requirements

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

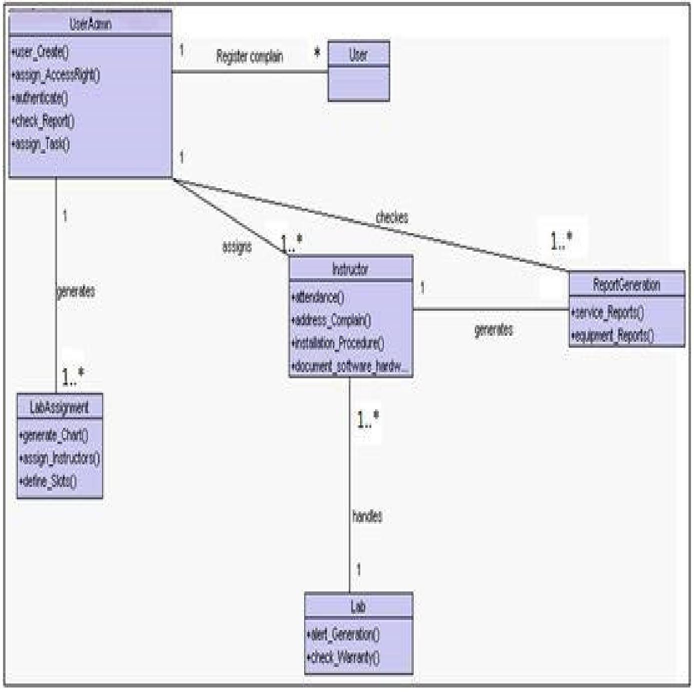


Fig. 5.1 Class diagram

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

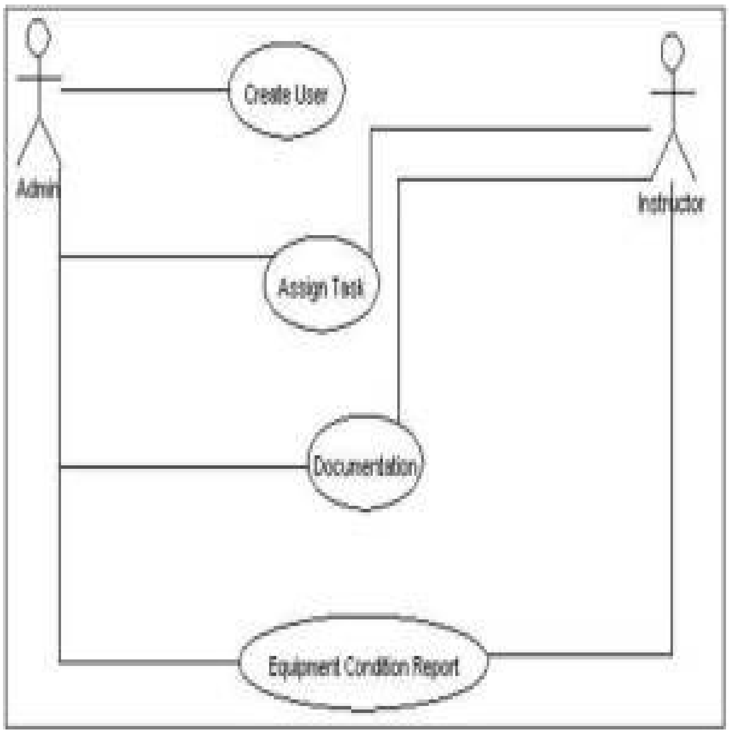
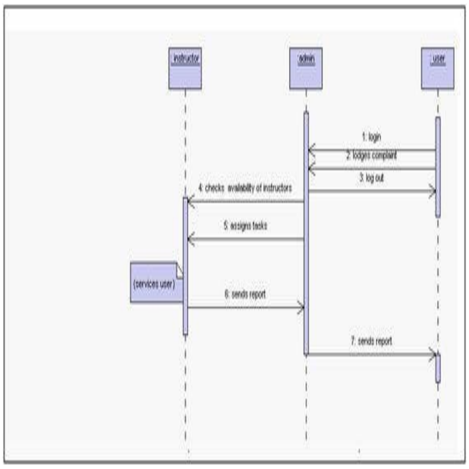


Fig. 5.2

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.