Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	**1**
Title:	**Software Requirement Specification**
Date of Performance:	**27/07/2023**
Roll No:	**9600**
Team Members:	**Cloyster DSouza**

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Lab Experiment 01

Experiment Name: Software Requirement Specification (SRS) as per IEEE Format

Objective: The objective of this lab experiment is to guide students in creating a Software Requirement Specification (SRS) document following the IEEE (Institute of Electrical and Electronics Engineers) standard format. The IEEE format ensures a structured and consistent approach to capturing software requirements, facilitating effective communication among stakeholders and streamlining the software development process.

Introduction: Software Requirement Specification (SRS) is a formal document that precisely defines the functional and non-functional requirements of a software project. The IEEE standard format provides a systematic framework for organizing the SRS, making it comprehensive, clear, and easily understandable by all parties involved in the project.

Lab Experiment Overview:

1. Introduction to IEEE Standard: The lab session begins with an overview of the IEEE standard format for SRS. Students are introduced to the various sections and components of the SRS as per the standard.
2. Selecting a Sample Project: Students are provided with a sample software project or case study for which they will create the SRS. The project should be of moderate complexity to cover essential elements of the IEEE format.
3. Requirement Elicitation and Analysis: Students conduct requirement elicitation sessions with the project stakeholders to gather relevant information. They analyze the collected requirements to ensure they are complete, unambiguous, and feasible.
4. Structuring the SRS: Using the IEEE standard guidelines, students organize the SRS document into sections such as Introduction, Overall Description, Specific Requirements, Appendices, and other relevant subsections.
5. Writing the SRS Document: In this phase, students write the SRS document, ensuring it is well-structured, coherent, and adheres to the IEEE format. They include necessary diagrams, use cases, and requirements descriptions.
6. Peer Review and Feedback: Students exchange their SRS documents with their peers for review and feedback. This review session allows them to receive constructive criticism and suggestions for improvement.
7. Finalization and Submission: After incorporating the feedback received during the review session, students finalize the SRS document and submit it for assessment.

Learning Outcomes: By the end of this lab experiment, students are expected to:

- Understand the IEEE standard format for creating an SRS document.
- Develop proficiency in requirement elicitation, analysis, and documentation techniques. Acquire the skills to structure an SRS document following the IEEE guidelines.

- Demonstrate the ability to use diagrams, use cases, and textual descriptions to define software requirements.
- Enhance communication and collaboration skills through peer reviews and feedback sessions.

Pre-Lab Preparations: Before the lab session, students should review the IEEE standard for SRS documentation, familiarize themselves with the various sections and guidelines, and understand the importance of clear and unambiguous requirements.

Materials and Resources:

- IEEE standard for SRS documentation
- Sample software project or case study for creating the SRS
- Computers with word processing software for document preparation Review feedback forms for peer assessment

Conclusion: The Software Requirement Specification (SRS) lab experiment in accordance with the IEEE standard format equips students with essential skills in documenting software requirements systematically. Following the IEEE guidelines ensures that the SRS document is well-organized, comprehensive, and aligned with industry standards, facilitating seamless communication between stakeholders and software developers. Through practical hands-on experience in creating an SRS as per the IEEE format, students gain a deeper understanding of the significance of precise requirement definition in the success of software projects. Mastering the IEEE standard for SRS documents prepares students to be effective software engineers, capable of delivering high-quality software solutions that meet client expectations and industry best practices.

Case Study 1—Requirements Specification Document

1. Abstract

The purpose of Training and Placement Management System is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with. Training and Placement Management System, as described above, can lead to error free, secure, reliable and fast management systems. It can assist the user to concentrate on their other activities rather than concentrating on the record keeping. Thus, it will help organizations in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information. The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

2. Introduction

2.1 Purpose

Maximum work goes manually in the present placement system which makes it take time to avail changes. This includes main problems like searching for the data of students and sorting them along with it. Also, updating student data is a cumbersome job and does not have a method to notify the student in time which makes the management of the placements very difficult. In the proposed system, all of these problems become automated. The registration of the student for an upcoming placement, the addition of a new user, notifying students on various platforms such as Email and SMS, sharing information, the privacy of the student, etc is all met. The admin validates the Information and gives the student list based on the criteria required which otherwise would have been very difficult to manage.

2.2 Scope

It may help collect perfect management in detail. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of the past year perfectly and vividly. It also helps in current work relative to the Training and Placement Management System. It will also reduce the cost of collecting the management & collection procedure will go on smoothly. Our project aims at Business process automation, ie, we have tried to computerize various processes of Training and Placement Management System.

2.3 Definitions, Acronyms, Abbreviations

KPI -Key Performance Indicator

2.4 References

[1] D. Pereira, “LinkedIn business model,” Business Model Analyst, 15-Jul-2022. [Online].Available: https://businessmodelanalyst.com/linkedin-business-model/. [Accessed: 09-Nov-2022].

[2] V. Hendelmann, “The indeed business model – how does indeed work & make money?,”productmint, 25-Oct-2022. [Online]. Available: https://productmint.com/the-indeed-business-model-how-does-indeed-work-make-money/ . [Accessed: 09-Nov-2022].

[3] “How indeed works business model and revenue model,” Website Design Company, 23Oct-2019. [Online]. Available: https://futureworktechnologies.com/how-indeed-works-business-revenue-model/. [Accessed: 09-Nov-2022].

[4] M. Qadri, “LinkedIn business model: How linkedin makes money,” BusinessModelOf, 05-Oct-2022. [Online]. Available: https://whatisthebusinessmodelof.com/business-models/how-linkedin-makes-money/. [Accessed: 09-Nov-2022].

[5] D. Pereira, “Glassdoor business model,” Business Model Analyst, 18-Jul-2022. [Online].Available: https://businessmodelanalyst.com/glassdoor-business-model/. [Accessed: 09-Nov-2022].

[6] Gennaro-Cuofano, “How does glassdoor make money? glassdoor business model in a Nutshell,” FourWeekMBA, 13-Oct-2022. [Online]. Available:https://fourweekmba.com/glassdoor-business-model/. [Accessed: 09-Nov-2022].

[7] M. Qadri, “LinkedIn business model: How linkedin makes money,” BusinessModelOf, 05-Oct-2022. [Online]. Available: https://whatisthebusinessmodelof.com/businessmodels/how-linkedin-makes-money/. [Accessed: 09-Nov-2022].

[8] Bootstrap get started. [Online]. Available:https://www.w3schools.com/bootstrap/bootstrap_get_started.asp. [Accessed: 09Nov-2022].

[9] “Connect your app to firebase | firebase realtime database,” Google. [Online]. Available:https://firebase.google.com/docs/database/android/start. [Accessed: 09-Nov-2022].

[10] “Getting started with firebase | firestore | google cloud,” Google. [Online]. Available:https://cloud.google.com/firestore/docs/client/get-firebase.[Accessed:09-Nov-2022].

2.5 Developer's Responsibilities

The developer is responsible for (a) developing the system, (b) installing the software on the client's hardware, (c) conducting any user training that might be needed for using the system, and (d) maintaining the system for a period of one year after installation.

3. General Description

3.1 Product Functions Overview

This software is supported to eliminate and in some cases,reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner. The application is reduced\ as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves its user-friendly. Training and Placement Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on
their other activities rather to concentrate on the record keeping.

3.2 User Characteristics

The main users of this system will be students applying for placement .The training and placement cell and companies that will post the jobs.

3.4 General Assumptions and Dependencies Not

applicable.

4. Specific Requirements

4.1 Inputs and Outputs

The system has two type of inputs and produces three types of outputs.

Input 1: The companies fill the application to submit jobs and corresponding application.
Input 2: The Students will fill the applications and submit relevant information on the job portal after signing in as students.

Output 1: these will be the jobs submitted by the companies , these wil be available to the admin to manage and approve.

Output 2:these will be the applications filled by the students will be visible to both the admin to keep a track of student progress and the companies who post the jobs

4.2 Functional Requirements

1)Grant privileges to various registered users of our website and send messages to the registered users.

2)grant The first ‘Admin’ privilege of storing and accessing the information of registered companies and students.

3)TPO should have similar privileges to admin.

4)Registration should allow students and companies to register for the process. It should give different set privileges according to the account type

5)Students should be able to create their profile, upload their CV, have reminders sent to them

6) Companies should have privileges of setting up their profile, choosing interview dates and to share their vacancies with the system

4.3 External Interface Requirements

User Interface:no external interface requirements

4.4 Performance Constraints

For Admin currently only the super admin supports user creation, blocking and deletion, the other admins only support job verification

4.5 Design Constraints Software

Constraints

The system is to run on laptops or computers only

Hardware Constraints

The system will run on any system that supports a basic web browser

Acceptance Criteria

Before accepting the system, the developer must demonstrate that the system is able to integrate the job requests from companies , applications from students together and give the control of the system to the admin

Postlab Questions:

a) Evaluate the importance of a well defined Software Requirement Specification (SRS) in the software development lifecycle and its impact on project success.

Some key reasons why a well-defined SRS is important and how it impacts project success are:

1. Clear and Shared Understanding: The SRS document outlines the project's objectives, features, functionalities, and constraints in a structured manner. It ensures that all stakeholders have a clear and shared understanding of what needs to be built, which helps avoid misunderstandings and discrepancies throughout the development process.

2. Scope Management: A well-defined SRS helps in defining the project's scope accurately. It outlines the in-scope and out-of-scope functionalities, which assists in preventing scope creep (uncontrolled expansion of project scope) and helps manage changes efficiently.

3. Requirement Validation: The SRS document allows stakeholders to review and validate the requirements early in the project's lifecycle. This validation process helps identify potential issues and ambiguities, reducing the risk of costly changes or rework later on.

4. Basis for Development: Developers rely on the SRS as a reference to design, implement, and test the software. A well-documented SRS provides developers with the necessary details, reducing the chances of misinterpretation and ensuring that the product aligns with the client's expectations.

5. Project Planning and Estimation: The SRS serves as the basis for project planning and estimation. It helps project managers determine the required resources, timeline, and budget for successful project execution.

6. Risk Mitigation: By identifying and documenting requirements clearly, the SRS helps in risk assessment and management.

7. Client Satisfaction: When the SRS accurately captures the client's needs and expectations, it enhances the likelihood of delivering a product that meets or exceeds those requirements. This, in turn, leads to higher client satisfaction and better chances of future business opportunities.

8. Traceability and Accountability: A well-structured SRS allows for easy traceability of requirements throughout the development process. This traceability aids in maintaining accountability, as each requirement can be tracked from conception to implementation.
9. Reduced Development Time and Cost: With a clear SRS in place, development teams can work more efficiently and avoid unnecessary rework or iterations, resulting in reduced development time and cost.

10. Legal and Contractual Compliance: In projects with formal contracts, the SRS serves as a legal document that defines the scope of work and ensures compliance with contractual obligations.

b) Analyse a given SRS document to identify any ambiguities or inconsistencies and propose improvements to enhance its clarity and completeness.

1. Ambiguous Language:

- Look for vague or unclear statements that could lead to different interpretations. - Identify terms or phrases with multiple meanings or lack specific details.

Improvement:

- Replace ambiguous terms with specific and well-defined vocabulary. - Provide clear and concise descriptions of requirements.

2. Inconsistent Information:

- Check for conflicting or contradictory requirements within the document. - Look for discrepancies in terminology, measurements, or formatting.

Improvement:

- Cross-reference related sections or requirements to ensure consistency. - Standardize terminology and units of measurement throughout the document.

3. Missing Information:

- Identify any gaps or incomplete requirements that lack necessary details. - Look for omitted sections or aspects that should be addressed.

Improvement:

- Fill in missing information to provide a comprehensive view of the project.
- Include relevant context, assumptions, and dependencies to avoid ambiguity.

4. Ambiguous Use Cases or Scenarios:

- Review use cases or scenarios for unclear steps or undefined inputs/outputs. - Check for inconsistent use case representations or missing alternative flows.

Improvement:

- Ensure each use case is well-defined with clear steps, preconditions, and post-conditions.
- Add alternative flows and exceptions to cover various scenarios comprehensively.

5. Unverifiable or Unrealistic Requirements:
- Identify requirements that cannot be objectively measured or validated.
- Look for requirements that may be impractical or beyond the project scope.

Improvement:
- Make sure all requirements are verifiable and measurable.
- Remove or revise requirements that are unrealistic or unattainable.

c) Compare and contrast different techniques for requirement elicitation, such as interviews, surveys, and use case modelling, and determine their effectiveness in gathering user needs.

1. Interviews:
Description: Interviews involve one-on-one or small group interactions between the requirement analyst and stakeholders. It allows for direct communication and discussion of specific topics.

Strengths:
- Real-time communication enables in-depth exploration of stakeholder needs.
- Analysts can ask follow-up questions to clarify ambiguities or delve into details.
- Personal interactions build trust and rapport with stakeholders, leading to more honest and open responses.

Limitations:
- Time-consuming, especially when dealing with multiple stakeholders.
- Responses may be biased due to the presence of the interviewer.
- Stakeholders may not always be available for interviews, leading to scheduling challenges.

2. Surveys:
Description: Surveys involve distributing questionnaires or forms to a large number of stakeholders to collect their opinions, preferences, and requirements.

Strengths:
- Efficient for gathering data from a large number of stakeholders simultaneously.
- Responses can be collected anonymously, encouraging honest feedback.
- Cost-effective, especially when dealing with geographically dispersed stakeholders.

Limitations:
- Limited scope for follow-up questions, which may result in less detailed responses.
- Stakeholders may not respond to the survey, leading to potential non-response bias.

- It might be challenging to capture complex or nuanced requirements through fixed-choice questions.

3. Use Case Modeling:

Description: Use case modeling is a technique used to capture functional requirements of the system by representing interactions between users and the system through scenarios.

Strengths:

- Provides a visual representation of how users interact with the system, making it easier to understand requirements.
- Helps in identifying system functionalities and boundary conditions.
- Encourages stakeholders to think in terms of user interactions and system responses.

Limitations:

- May not fully capture non-functional requirements or system constraints.
- Requires a good understanding of system behavior and user interactions for effective modeling.
- Focuses on what the system should do, but not necessarily on how it should be implemented.

Effectiveness in Gathering User Needs:

- Interviews are highly effective in gathering user needs, especially when in-depth understanding and clarification are required. They foster rich communication and allow for a deeper exploration of requirements.
- Surveys are efficient for gathering a wide range of opinions from a large number of stakeholders. However, they may not capture the same level of detail as interviews or use case modeling.
- Use case modeling is effective in capturing functional requirements and illustrating system-user interactions. It is particularly useful for understanding the system's behavior from a user's perspective.