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:	9568
Team Members:	

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(A ll used)	02 (Partial)	01 (rarely followe d)	
4	Post Lab Questions (04)	04(don e 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.

Event Management System

Abstract:

An Event Management System is an online website which helps users to know about a particular event or Keeping track of a particular event which users prefer. The main goal of this website is to achieve protection as well as simplicity for user seeking better booking of the event. Searching for an event may be entertainment, sports, socio-cultural or political of your preference while sitting in remote corners of the world becomes a difficult task for a user to search for an event of his convenience. Also information can be miscommunicated or misunderstood for a particular event. Hence a website/Application needs to be built in order to solve these problems. Event Management System(website) provides a smart solution to this problem. The Event Management System will work on providing exact data/information about a particular event. Through this website User can book multiple event of his choice, also he or she can update her profile and can also update his/her event, this website ensure high level security for user and also help in getting time to time updates, of new upcoming events so that the particular person can get proper information at the time of particular event. .It will also help users to know the price of a seat for a particular event. Through the Event Management System, various malpractices can also be prevented. The Event Management System will Keep monitoring user activity too. Hence Event Management System will not only create an easy life for users but also help in increasing the popularity of that event which were unknown to users.

Introduction:

2.1 Purpose:

The Event Management System is a smart System, which will help users to get time to time updates of recent events, once user login to a website. Through this website, Organizer will get the opportunity to connect with a wide range of people. The sole purpose of this website is to help both users and organizers get flexibility to view, communicate, organize with each other. Further, it increases security and enhances confidence between user and organizer.

2.2 Scope:

The objective of this application is to develop a system that effectively manages all the data related to the various events that take place in an organization. The purpose is to maintain a centralized database of all event related information. The goal is to support various functions and processes necessary to manage the data efficiently.

2.3 Definitions, Acronyms, Abbreviations:

An Event Management System (or Event Management Software) is a tool to assist organizers with the planning, execution, and management of their events. Examples of such events include music concerts, conferences, exhibitions, festivals, ceremonies, sporting events, and many more.

2.4 References:

[1]. L. McCathie and K. Michael, "Is it the End of Barcodes in Supply Chain Management?", Proceedings of the Collaborative Electronic Commerce Technology and Research Conference LatAm, 2005.
[2] Glenn Bowdin, Johnny Allen, William ÓToole, Rob Harris, Ian McDonnell, Events Management, Routledge London and New York 2011.
[3] Silvers, Julia Rutherford, Risk Management for Meetings and Events, Elsevier Amsterdam 2011.
[4]. M.Mahalakshmi, S.Gomathi and, S.Krithika, "Event Management System", 2016.
[5]. Sandeep Misal, Segar Jadhav, Tushar Jore , Archana Ugale ,"Event Management System .

2.5 Developer's Responsibilities:

The Developer's Responsibilities is a) To help users get in touch with the list of events the user has enrolled in. b) To keep the website updated with new events c) Help user as well as organizer in solving their doubts d) Taking care of user's as well as Organizer's personal information.

3 General Description:

3.1 Literature Overview:

The aim of the study is to show the dimensions applied while managing and marketing a successful special sport event and is going to make several contributions for the sport event managers and sport marketing managers about how to use event management dimensions effectively and shape strategies based on this perspective. Events are leisure activities and work possibilities for people. Events bring people together and make them have a good time. They enhance the quality of people’s life; they can provide significant economic benefits and can also provide revenue for special projects. Regardless of size, events require a high degree of planning, a range of skills and a lot of energy. Android has become popular among smartphone users. This paper discusses the intent to solve the problems of propagating news and information, and also alleviate the problem of traditional event managing procedures such as lots of paperwork, or long queues at the registration desk. The objective of this project is to develop an android application which will provide interesting news and events so that users will be able to manage their event smoothly. The objective of this project is to develop an android application which will provide interesting news and events so that users will be able to manage their event smoothly. Event organizers are continually appreciating the benefits of an online event management system such as an Event management application. Crowd management involves keeping track of the crowd, the space available and balancing between the crowd and space. In this paper, we propose a prototype for counting the people as a part of developing a better crowd monitoring system.

3.2 User Characteristics:

The main Users of this System will be the people who want to view or enroll in a particular event. Further, it will be the organizer as well, who will be organizing the event.

3.3 General Constraints:

The system should have clock speed 3.0 GHZ , RAM size 512 MB ,Hard Disk Capacity:400 GB.

3.4 General Assumptions and Dependencies:

Not Applicable.

4 Specific Requirements:

4.1 Inputs and Outputs:

The User logins in the website and can check the list of events, in the event's page(Input). He can check the book event by going to the booked event’s page(Output). The organizer can create events using the create event option(Input). And it will appear in the Event page(Output).

4.2 Functional Requirements:

Proper access to the internet.

4.3 External Interface Requirements:

The user can login and view a list of events present in the website or if wants to organize an event , he/she can do so.

4.4 Performance Constraints:

Not Applicable.

4.5 Design Constraints:

Software Constraints:

1. HTML/CSS/BootStrap/Js
2. Atom Editor
3. Vs code(extensions)
4. MySQL, XAMPP
5. PhpMyAdmin
6. Apache
7. PHP

Hardware Constraints:

1. Processor Intel dual core and above
2. Clock speed 3.0 GHZ
3. RAM size 512 MB
4. Hard Disk Capacity:400 GB
5. 15 inch color monitor.

Acceptance Criteria:
The System has worked with a high acceptance rate. It has been viewed by Mentor as well as external mentor, and has approved its working. We have got good responses from users and organizers using this website.

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.

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.

1. 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.

Unclear or unrefined user sources:

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.