

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 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.

Skin Care App—Requirements Specification Document

1 Abstract

This project aims to provide a solution to various skin .The ultimate aim of this skincare website is to suggest products to the user based on their skin concern. Skin is one of the many organs in the human body. It acts as a protective barrier between many injuries and also acts as a sensory organ. Hence, one should also protect the skin from various dangers. This website will give the user various options to choose from a variety of different brands and budgets. It also suggests many skincare tips to the user. It is time efficient and it brings all the information under one roof. It aims to make skincare simple for all irrespective of gender or skin type. The project's main objective on Skincare is to manage all the details of users, products, and tips. Here we will provide the user with appropriate results according to the type of skin and based on their skin problems.

2 Introduction

2.1 Purpose

The skin care app employs a data-driven approach to offer personalized solutions for various skin problems. By utilizing user input and analyzing their skin concerns, the app suggests suitable products with optimal ingredient concentrations. It incorporates a robust recommendation engine that takes into account gender, age, and individual preferences to cater to a diverse user base. Continuous improvement is facilitated through user feedback mechanisms, allowing for iterative enhancements and optimization. By prioritizing usability, data accuracy, and the user experience, the app aims to become a reliable and comprehensive resource for skincare enthusiasts of all ages and genders.

2.2 Scope

The skin care app ensures efficient data collection from users with minimal processing time. The user registration and profile creation process is straightforward, allowing for quick and seamless data input. The app's backend infrastructure is designed to handle a high volume of user interactions, ensuring availability and responsiveness even during peak usage. When users explore skin care products on the app, the information about each product, including its ingredients, is presented in a clear and concise manner. The app's data retrieval and presentation capabilities are optimized for swift response times, enabling users to gain comprehensive insights into the products without experiencing delays.

2.3 Definitions, Acronyms, Abbreviations

Not applicable.

2.4 References

[1] Suruchi Bhardwaj, Shweta Parashar, Kajal Verma, Radhika Arora, B.S.

Chhikara “Evaluation of awareness about beauty products composition and proper utilization among college students” - August 2019

- [2] A. Mohanapriya, M. Padmavathi, A. Prasathkumar “A Comparative Study on the Impact of Skin Care Products on the Self-Esteem of Women Users and Non-Users in Coimbatore City” - January 2019

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.

1

3 General Description

3.1 Functions Overview

In the computer science department there are a set of classrooms. Every semester the department offers courses, which are chosen from the set of department courses. A course has expected enrollment and could be for graduate students or undergraduate students. For each course, the instructor gives some time preferences for lectures.

The system is to produce a schedule for the department that specifies the time and room assignments for the different courses. Preference should be given to graduate courses, and no two graduate courses should be scheduled at the same time. If some courses cannot be scheduled, the system should produce a “conflict report” that lists the courses that cannot be scheduled and the reasons for the inability to schedule them.

3.2 User Characteristics

The main users of this system will be department secretaries, who are somewhat literate with computers and can use programs such as editors and text processors.

3.3 General Constraints

The system should run on Sun 3/50 workstations running UNIX 4.2 BSD.

3.4 General Assumptions and Dependencies

Not applicable.

4 Specific Requirements

4.1 Inputs and Outputs

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

Input 1: User's Details

Here, the user will be entering its details eg. name, age, gender, etc so that a personalized

experience will be given to the user.

Input 2: User's Skin Type and Problems

The user will add the type of skin it has and also the various skin problems it is facing.

Outputs:

1. Skincare Tips: The app will offer expert skincare tips and advice to users, covering general skin care practices, targeted remedies, and best practices for maintaining healthy skin.

2. Dermatologist Search Results: Users will be presented with a list of qualified dermatologists and skin care specialists in their local area, along with contact information for professional consultations and treatments.

4.2 Functional Requirements

1. User Authentication:

The user will be able to create a profile by entering various details like name, gender, age, etc so that there will be a personalised user experience and the data of the user is not shared. The user will only have to login next time in spite of entering the details every time.

2. Specialized Tips and Tricks:

This app provides specialised tips for the users according to their skin problems and also gives various suggestions to improve their skin.

3. Personalised Mentor:

The user's will be allocated a mentor which will guide them for various skin problems they are facing and provide them with appropriate solutions or products to use.

4.3 External Interface Requirements

User Interface: Only one user command is required. The file names can be specified in the command line itself or the system should prompt for the input file names.

4.4 Performance Constraints

For input file 2 containing 20 courses and up to 5 preferences for each course, the reports should be printed in less than 1 minute.

4.5 Design Constraints

Software Constraints

The system is to run under the UNIX operating system.

Hardware Constraints

The system will run on a Sun workstation with 256 MB RAM, running UNIX. It will be connected to an 8-page-per-minute printer.

Acceptance Criteria

Before accepting the system, the developer must demonstrate that the system works on the course data for the last 4 semesters. The developer will have to show through test cases that all conditions are satisfied.

PostLab:

Q1.Ans.

A well-defined Software Requirement Specification (SRS) is a critical document in the software development lifecycle as it serves as the foundation upon which the entire project is built. Its importance lies in providing a clear and detailed description of the software's functional and nonfunctional requirements, as well as the scope, constraints, and expectations for the project. The SRS acts as a contract between the stakeholders (clients, users, developers, and testers) and sets the direction for the entire development process. Here are some key reasons why a well-defined SRS is essential and its impact on project success:

1.Validation and Verification: With a well-defined SRS, it becomes easier to validate the software's functionality against the requirements. Testing and verification teams can use the SRS as a reference to ensure that the software meets the intended specifications.

2.Customer Satisfaction: A clear and well-defined SRS ensures that the final product aligns with the client's expectations. When customers' needs are met, it

increases their satisfaction with the product and enhances the chances of repeat business or positive referrals.

3.Change Management: Requirements may change throughout the project. A well-documented SRS provides a baseline against which any changes can be measured and assessed for their impact on the project.

4.Alignment with Business Goals: The SRS helps in aligning the development efforts with the overall business goals and objectives. This alignment is crucial for the success of the project and its relevance to the organisation.

Q2.Ans.

Software Requirement Specification(SRS):

1. Introduction:

The "Skincare Website" aims to address various skin issues like acne, hyperpigmentation, wrinkles, etc. People often rely on remedies they find online, but these can be harsh and harmful to the skin. Quality skincare products with scientifically researched ingredients are essential for maintaining healthy, glowing, and youthful skin. The website recommends specialised formulations to tackle specific skincare problems, emphasising the importance of a well-developed skincare routine including cleanser, moisturiser, sunscreen, and serums for targeted concerns.

2. Functional Requirements

1.User Authentication:

The user will be able to create a profile by entering various details like name, gender, age, etc so that there will be a personalised user experience and the data of the user is not shared . The user will only have to login next time in spite of entering the details every time.

2.Specialized Tips and Tricks:

This app provides specialised tips for the users according to their skin problems and also gives various suggestions to improve their skin.

3.Personalised Mentor:

The user's will be allocated a mentor which will guide them for various skin problems they are facing and provide them with appropriate solutions or products to use.

Q.3.Ans

Requirement elicitation is a crucial phase in the software development process where the needs and expectations of users and stakeholders are gathered and documented. Various techniques can be employed for requirement elicitation, each with its strengths and weaknesses. Let's compare and contrast three common techniques: interviews, surveys, and use case modelling, and assess their effectiveness in gathering user needs.

1. Interviews:

- Interviews involve direct communication with stakeholders, users, or subject matter experts.
- **Advantages:** Provides in-depth understanding, allows for follow-up questions, clarifications, and the opportunity to build rapport. Facilitates the exploration of complex requirements and capturing tacit knowledge.
- **Disadvantages:** Time-consuming, requires skilled interviewers, and might be influenced by biases or limited perspectives of participants.

2. Surveys:

- Surveys are questionnaires distributed to a group of users or stakeholders to collect their opinions and preferences.
- **Advantages:** Can reach a large number of participants, provides quantifiable data, and allows for anonymity, which may encourage honest feedback.
- **Disadvantages:** Limited in-depth understanding, no room for follow-up questions or clarifications, and may not capture complex or context-dependent requirements effectively.

3. Use Case Modelling:

- Use case modelling involves creating scenarios to represent how users will interact with the system to achieve specific goals.

- **Advantages:** Helps visualise system behaviour and user interactions, identifies functional requirements, and clarifies system boundaries.

- **Disadvantages:** May not capture all non-functional requirements effectively, and it requires some technical expertise to create and understand the models.

Effectiveness in Gathering User Needs:

- **Interviews:** Highly effective in gathering comprehensive and detailed user needs. They facilitate a deeper understanding of users' pain points, expectations, and context, resulting in accurate and actionable requirements.

- **Surveys:** Effective for gathering a large number of opinions and preferences from a diverse group of users. They provide valuable insights into user preferences and general trends but may lack the depth necessary to capture complex requirements.

- **Use Case Modelling:** Effective in capturing functional requirements and system behaviour. It helps identify key features and user interactions. However, it might not be as effective in uncovering non-functional requirements or the emotional aspects of user needs.