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.

Celestial—Requirements Specification Document

1 Abstract

Celestial is an interactive mobile application designed to cater to individual skincare needs. The app enables users to input their specific skin concerns, such as acne, dryness, aging signs, and hyperpigmentation. Based on this input, the app generates personalized skincare product recommendations, custom skincare routines, and valuable skincare tips. Users can easily access suitable skincare products and routines, tailored to their unique skin types and concerns. Moreover, the app offers a convenient feature to locate nearby dermatologists and skin care specialists, fostering accessibility to professional skincare advice and treatments. This app aims to provide users with a comprehensive and user-friendly platform for achieving healthy and vibrant skin.

2 Introduction

2.1 Purpose

The purpose of the app is to assist users in addressing their specific skin problems by providing personalized skincare product recommendations, customized skincare routines, and valuable skincare tips. Additionally, the app allows users to conveniently locate qualified dermatologists in their area for professional skincare advice and treatments.

2.2 Scope

It helps in collecting user details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the skin products vividly as all the products included in this website have good ingredients. It will also help saving time rather than going to many websites to check which product to use.

2.3 Definitions, Acronyms, Abbreviations

Not applicable.

2.4 References

[1] Surber C, Kottner J, Skin care products “What do they promise, what do they deliver”, Journal of Tissue Viability (2016)

[2] 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

[3] 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

3 General Description

3.1 Functions Overview

The skincare app offers the following functions: a) User registration, authentication and quiz b) Skincare products and routines c) Profile management for updating personal information d) Dermatologists within the area e)Feedback

3.2 User Characteristics

The main users of this system will be people who face skincare issues and want to incorporate skincare products in their everyday life.

3.3 General Constraints

The system should run on most devices that can run internet and browsers.

3.4 General Assumptions and Dependencies

Not applicable.

4 Specific Requirements

4.1 Inputs and Outputs

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

Inputs:

1. User's Skin Concerns: The app will accept user inputs regarding their specific skin problems, such as acne, dryness, aging signs, sensitivity, or other dermatological issues.

2. User Profile: Users will provide relevant information, such as age, gender, skin type, lifestyle, and any known allergies or sensitivities, to personalize the skincare recommendations and routines.

Outputs:

1. Personalized Skincare Recommendations: The app will generate tailored skincare product suggestions based on the user's skin concerns and profile, providing a list of suitable cleansers, moisturizers, serums, and other skincare products.

2. Customized Skincare Routine: Users will receive a personalized skincare routine, considering their skin type and concerns, outlining the sequence and frequency of product usage to improve their skin health.

3. 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.

4. 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 Registration and Profile Management:

1.1. Users should be able to create accounts and provide essential information, such as name, email, and password, for registration.

1.2. The app must allow users to edit and update their profile details, including age, gender, skin type, and skincare preferences.

2. Skin Concern Input and Analysis:

2.1. The app should offer an intuitive interface for users to input their specific skin concerns and problems.

2.2. The system must analyze the user's input to identify primary skin concerns, considering multiple factors like severity, frequency, and type of issues.

3. Skincare Product Recommendations:

3.1. Based on the skin concern analysis, the app should provide personalized skincare product recommendations, considering the user's profile and skin type.

3.2. The system should offer a diverse range of suitable skincare products, including cleansers, moisturizers, serums, and treatments.

4. Personalized Skincare Routine Generation:

4.1. The app must generate customized skincare routines based on the user's skin concerns and skincare product recommendations.

4.2. The system should outline the sequence and usage frequency of recommended products in the skincare routine.

5. Skincare Tips and Expert Advice:

5.1. The app should present informative skincare tips and advice from skincare experts and dermatologists.

5.2. The system should offer a variety of tips, covering general skin care practices, remedies for specific concerns, and best practices for optimal skin health.

6. Dermatologist Search and Listing:

6.1. The app must provide a search feature for users to find qualified dermatologists and skin care specialists based on their location.

6.2. The system should display a list of nearby dermatologists with relevant contact information for further consultation.

7. User-Product Interaction Tracking:

7.1. The app should track user interactions with skincare products to evaluate their effectiveness and user preferences.

7.2. The system must allow users to provide feedback and ratings for recommended skincare products.

8. User Feedback and Support:

8.1. The app should provide a mechanism for users to submit feedback and report issues related to the app's functionality.

8.2. The system must offer customer support and assistance for users experiencing technical difficulties or seeking additional information.

4.3 External Interface Requirements

User Interface: The app will have an intuitive and user-friendly interface, with sections for the home feed, feedback, profile, skincare routines and dermatologists.

4.4 Performance Constraints

The app should provide real-time data retrieval and messaging for a seamless user experience, maintaining responsiveness during peak usage.

4.5 Design Constraints

Software Constraints

The app will be developed for Android devices using Android Studio and Java programming language to be run under the Windows Operating system

Hardware Constraints

The system will run on Android devices under internet connectivity.

Acceptance Criteria

Before acceptance, the app will undergo rigorous testing to ensure smooth functionality and bug-free operation. User feedback will be taken into account for improvements, ensuring a flawless user experience.

POSTLAB:

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

Ans : A well-defined Software Requirement Specification (SRS) is a crucial document in the software development lifecycle, and its importance cannot be overstated. The SRS serves as the foundation upon which the entire software project is built. Its significance lies in its ability to establish clear and unambiguous communication between stakeholders, guide development teams, and ensure the project's success. Here's an evaluation of its importance and impact:

Clear Communication: The SRS serves as a common reference point for all stakeholders involved in the project, including clients, developers, designers, testers, and project managers. It outlines the project's goals, functionalities, constraints, and expectations, ensuring that everyone has a shared understanding of what the software is meant to achieve.

Minimized Ambiguity and Misunderstanding: A well-detailed SRS helps eliminate ambiguity and reduces the chances of misunderstandings or misinterpretations. It provides a detailed description of requirements, use cases, and interactions, leaving little room for assumptions or guesswork.

Accurate Cost and Time Estimates: With a clear and comprehensive SRS, project managers can provide accurate estimates for costs, timeframes, and resource allocation. This leads to more realistic planning and scheduling, helping the project stay on track.

Effective Project Planning: The SRS serves as the basis for creating a project plan that outlines tasks, milestones, and deliverables. It ensures that all aspects of the project are considered and planned for, reducing the risk of scope creep or sudden changes.

Guidance for Development Teams: Developers rely on the SRS to understand the technical and functional requirements of the software. A well-defined SRS helps developers write clean, efficient code that aligns with the project's goals and client expectations.

Ans b)

Software Requirement Specification (SRS) - Celestial System

1. Introduction: The Celestial System aims to create an online platform to enhance skin of users by incorporating good skincare products.

2. Functional Requirements:

2.1 User Registration and Login:

● Ambiguity: "Users can specify the required user details for registration, such as full name, email address, password, and optional social media account integration ."

● Improvement: Users can register using their email or social media accounts.

2.3 Skin Details:

● Inconsistency: "User tells the skin problem."

● Improvement: Users can upload a pic of their skin which will recommend the skin issue.

Ans c)

Requirement elicitation is a crucial phase in the software development process, where the goal is to gather and understand user needs, expectations, and specifications. Various techniques can be employed for requirement elicitation, including interviews, surveys, and use case modeling. Each technique has its strengths and weaknesses, and their effectiveness in gathering user needs depends on the context, project, and stakeholders involved.

1. Interviews:

- **Description:** Interviews involve direct, one-on-one conversations between the software developers and stakeholders, such as users, clients, and subject matter experts.
- **Advantages:**
    - Provides in-depth information by allowing open-ended discussions.
    - Enables clarifications and follow-up questions for better understanding.
    - Allows for building rapport and trust with stakeholders.
- **Disadvantages:**
    - Time-consuming, especially for large groups of stakeholders.
    - May be influenced by biases or misinterpretations of interviewers. May not represent the views of all potential users.

2. Surveys:

- **Description:** Surveys are questionnaires or forms distributed to a group of stakeholders to gather their opinions, preferences, and requirements.
- **Advantages:**
    - Can reach a larger number of stakeholders efficiently.
    - Provides quantitative data for analysis.
    - Allows anonymity, encouraging honest feedback.
- **Disadvantages:**
    - Limited to the scope and format of the questions asked.
    - May lack the depth and context provided by interviews.
    - Limited opportunity for clarification of responses.

3. Use Case Modeling:

- **Description:** Use case modeling involves creating scenarios that describe how users interact with the system to achieve specific goals. Use cases define interactions between actors (users or external systems) and the system.
- **Advantages:**

- Provides a visual representation of user-system interactions.
- Helps identify system functionalities and user roles.
- Supports identification of potential errors or gaps in requirements.

Effectiveness in Gathering User Needs:

- **Interviews:** Interviews are highly effective for gathering in-depth and nuanced information. They allow for personalized interactions, which can reveal hidden or unanticipated requirements. However, they can be time-intensive and might not scale well for large user groups.
- **Surveys:** Surveys are efficient for collecting data from a larger audience, providing insights into common preferences and patterns. They are particularly useful for collecting quantitative data. However, they might lack the depth and context needed to fully understand complex requirements