Department of Computer Engineering

Academic Term: First Term 2023-24

Class: 'I'.E /Computer Sem — Y / Software Engineering

Practical No:	1
Title:	Software Requirement Specification
Date of Performance:	25-7-2023
Roll No:	9634
Team Members:	Aditya,alex,Rianna,Chris

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On timeCompletion & 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:

Department of Computer Engineering

Academic Term: First Term 2022-23

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

Signature of the Teacher:

Aditya Mahanwar 9620

Hospido-An app that helps you during emergencies

1 Abstract

Hospido, an innovative healthcare application, has been instrumental in tackling the challenges posed by the COVID-19 outbreak in densely populated areas like Mumbai. Doctors, facing unprecedented demands, doubled their efforts to meet urgent patient needs. Hospido's platform connects patients with essential healthcare services, offering quick access to specialized doctors and hospitals. A simple click allows patients to alert top hospitals about their arrival, ensuring swift responses. The app also facilitates online appointments with specialists and provides vital information about hospitals, bed availability, and attending doctors. Hospido's mission is to bridge the gap between medical facilities and patients, ensuring timely and convenient healthcare responses with a strong focus on safety.

2 Introduction

2.1 Purpose

1) **Convenience:** To create an application that provides hospital options to individuals in Mumbai .

2)**Self Booking**:Patients can directly find a hospital and book appointments for themselves through our app.

3)**In emergencies:** Users are provided with an emergency button to use during an emergency. This button when clicked sends a text alert to close a emergency contact given by the user.

2.2 Scope

1. **Convenient Appointment Booking**: The app allows patients to schedule appointments with doctors, specialists, and other healthcare providers conveniently. It can offer a user-friendly interface where patients can browse through available time slots and select the one that suits them best.
2. **Reduced Waiting Times**: By streamlining the appointment process, the app can help reduce patient waiting times. This benefits both patients, who spend less time in waiting rooms, and healthcare providers, who can manage their schedules more efficiently.
3. **Notifications and Reminders**: The app can send automated reminders to patients about their upcoming appointments, reducing the likelihood of no-shows and ensuring that patients are well-prepared for their visits.

2.3 Definitions, Acronyms, Abbreviations

Not applicable.

2.4 References

[1]AndrOid studio reference -

https://www.cOdewithharry.cOm/videos/andrOid-tutOrial/ttps://www.youtube.com/watch?v=sZ8D1-hNeWo&t=66

[2] Firebase for database

[3]PractO- https: //www.practO.cOm/

2.5 Developer's Responsibilities

1. **User Authentication and Authorization**: To ensure the utmost security and protect patient information from unauthorized access, Hospido implements robust user authentication mechanisms. The app takes extra measures by considering the implementation of Two-factor authentication for enhanced security. With these measures in place, patients and medical professionals can have confidence in the app's ability to safeguard their sensitive data effectively.
2. **Testing and Quality Assurance**: The responsibility of developers extends to conducting comprehensive testing of the app to detect and resolve any potential bugs or vulnerabilities. To guarantee the app's optimal performance and a seamless user experience, well-established quality assurance processes are implemented. Through rigorous testing, developers can ensure that the app functions precisely as intended, providing users with a reliable and efficient platform for accessing healthcare services. By adhering to stringent testing protocols, developers strive to deliver a secure and user-friendly application that meets the highest standards of performance and safety.
3. **Accessibility**: The app should be designed with accessibility in mind, ensuring that it can be used by people with disabilities. This includes support for screen readers, adjustable font sizes, and other accessibility features.
4. **User Interface and Experience**: The developers should create an intuitive and user-friendly interface that makes it easy for patients to schedule appointments, view their medical information, and access relevant resources.
5. **Integration with Hospital Systems**: Developers must ensure a smooth and secure data exchange between the app and the hospital's existing systems, like EHR. Robust encryption and testing minimize the risk of data breaches, prioritizing patient information's safety and confidentiality.

3 General Description

3.1 Product Functions Overview

Immediate Medical Support: The app offers prompt options for urgent and emergency medical needs, providing patients with immediate attention or guidance during critical situations.

Effortless Appointment Management: Patients can easily reschedule or cancel appointments through the app, adhering to the hospital's policies, for a hassle-free experience.

Tailored Doctor Selection: With a comprehensive search feature, patients can find doctors or specialists based on their specialty, location, availability, patient reviews, and other relevant criteria. This empowers patients to select their preferred healthcare provider for appointments, ensuring a personalized and satisfactory healthcare experience.

3.2 User Characteristics

1. **Patients**: The app's primary user group comprises patients seeking medical care, appointments, and health-related information. Patients can be of various ages, backgrounds, and medical needs.
2. **Healthcare Professionals:** The app is also utilized by healthcare providers, including doctors, specialists, nurses, and other medical experts. They rely on the app to efficiently manage their appointment schedules, access patient information, and communicate with their patients.
3. **Supportive Administrative Staff:** The app extends its benefits to the hospital's administrative staff, such as receptionists and scheduling coordinators, who can use it to assist patients with appointment bookings and efficiently manage the healthcare provider's schedules. This collaborative approach streamlines the overall healthcare process, benefitting both patients and medical staff.

3.3 General Constraints

The system should run on phones with Android with software OreO.

4 Specific Requirements

4.1 Inputs and Outputs

Inputs :

1]User Registration/Login
- Full name

- Email address

- Password

-Emergency contact.

Input2]Hospitals in mumbai.

-Selection of hospital by name

-Booking of appointment

-Location of hospital (postal address and on map)

Input 3]Emergency button

-Giving contact number to be listed for emergencies

Output:

1]Hospitals in mumbai

-All details of hospitals

-Display location of hospital

-Photo of the hospital

2]Emergency button

-sends a automated message to the emergency contact and also sends current location of the device.

4.2 Functional Requirements

Not applicable

4.3 External Interface Requirements

Medical history of the user is required to provide help during an emergency. Contact information of users and their close peers are required to contact in case of an emergency.

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.

Hardware Constraints :

An Android phone is required to run this software, with proper internet connection and network

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

POST LAB QUESTIONS FOR SOFTWARE ENGINEERING LAB

A. Importance of a well-defined Software Requirement Specification (SRS):-

A well-defined Software Requirement Specification (SRS) plays a pivotal role in the software development lifecycle and has a significant impact on project success. It ensures clear communication between stakeholders, reducing misunderstandings and ambiguities. By defining the project scope, the SRS prevents scope creep and helps manage expectations. Additionally, it establishes requirement traceability, ensuring that customer needs are met and the final product is validated accordingly. The SRS provides a solid foundation for estimations, aiding in project planning and resource allocation. It also enables effective risk mitigation by identifying potential issues early in the development process. Moreover, the SRS facilitates change management, allowing the assessment of proposed changes' impact on existing requirements.Quality assurance efforts are enhanced through the SRS, as it enables comprehensive testing based on the defined specifications. Overall, a well-structured SRS contributes to project success by promoting clarity, minimizing risks, and ensuring that the final software product aligns with customer expectations and requirements.

B. **Analyze a given SRS document to identify any ambiguities or inconsistencies and propose improvements to enhance it clarity and completeness.**

1.Inconsistent Information: - Check for conflicting or contradictory requirements within the document

- Look for discrepancies in terminology, measurements, or formatting.

->Proposed Improvement: - Cross-reference related sections or requirements to ensure consistency.

- Standardized terminology and units of measurement throughout the document.

2.Missing Information: - Identify any gaps or incomplete requirements that lack necessary details.

- Look for omitted sections or aspects that should be addressed.

->Proposed Improvement:

- Fill in missing information to provide a comprehensive view of the project.

- Include relevant context, assumptions, and dependencies to avoid ambiguity

3.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.

-> 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.

-> Proposed improvements: conduct stakeholder reviews, use diagrams, define acceptance criteria, document assumptions, address non-functional requirements.

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.

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. Interviews and focus groups are effective for specific needs but time-consuming and resource-intensive.

Surveys involve distributing questionnaires or forms to a large number of stakeholders to collect their opinions, preferences, and requirements. Surveys are efficient for broad overviews but may lack detailed insights. Combining methods offers a comprehensive understanding of user requirements while optimizing time and resources.

Use case modeling is a technique used to capture functional requirements of the system by representing interactions between users and the system through scenarios.It provides a visual representation of how users interact with the system, making it easier to understand requirements and also helps in identifying system functionalities boundary conditions. ,Encourages stakeholders

to think in terms of user interactions and system. But use case modeling may not fully capture non-functional requirements or system constraints. It Requires a good understanding of system behavior and user interactions for effective modeling. Use case modeling Focuses on what the system should do, but not necessarily on how it should be implemented.

Effectiveness in Gathering User Needs: Interviews capture detailed requirements but are more suitable for critical projects. Surveys provide quantitative data from a larger audience but lack qualitative depth. Use case modeling complements other techniques and visually represents system behavior. Ultimately, a combination of these techniques can ensure a comprehensive understanding of user needs and contribute to a successful software development project.