Software Requirements Specification

for

Sobdobarta

Prepared by

Sanjida akter Samanta

BFH2025010F

Institute of Information Technology

Noakhali Science and Technology University

Contents

[1. Introduction 1](#_Toc127738272)

[1.1 Problem Statement 1](#_Toc127738273)

[1.2 Purpose 1](#_Toc127738274)

[1.3 Project Scope 1](#_Toc127738275)

[1.4 Overview 2](#_Toc127738278)

[2. Stakeholders and Characteristics 2](#_Toc127738279)

[2.1 User 2](#_Toc127738280)

[2.2 Developer 2](#_Toc127738281)

[3. Design and Implementation Constrains 3](#_Toc127738288)

[3.1 User Interface Technology 3](#_Toc159107183)

[3.2Programming Language 2](#_Toc159107184)

[3.3 Hardware Interfaces 3](#_Toc159107185)

[4. Requirement Specification 3](#_Toc127738294)

[4.1 Functional Requirement 3](#_Toc127738295)

[4.1.1 Add username 3](#_Toc159107188)

[4.1.4 Send Message 4](#_Toc159107191)

[4.1.5 Receive Message 4](#_Toc159107192)

[4.2 Non-Functional Requirements 4](#_Toc159107193)

[4.2.1 Performance 4](#_Toc159107194)

[4.2.2 Scalability 5](#_Toc159107195)

[4.3 Usability Requirements 5](#_Toc159107196)

[4.3.1 Simple and Intuitive navigation 5](#_Toc159107197)

[4.3.2 Clear and Consistent Visual Design 5](#_Toc159107198)

[4.4 Security Requirements 6](#_Toc159107199)

[4.4.1 Secure Transmission 6](#_Toc159107200)

[4.5 Performance Requirements 6](#_Toc159107201)

[4.5.1 Responsiveness 6](#_Toc159107202)

[5. Requirement Engineering Process 6](#_Toc127738333)

[5.1 Requirement Elicitation Techniques 6](#_Toc127738334)

[5.2 Requirement Validation 7](#_Toc127738341)

[5.2.1 Walkthrough 7](#_Toc127738342)

[5.2.2 Inspection 7](#_Toc127738343)

[6. Use Case Diagram 7](#_Toc127738347)

[7. Use Case Description 8](#_Toc127738348)

[8. Requirement Traceability Matrix 10](#_Toc127739020)

[9. Appendix 10](#_Toc127739021)

[9.1 Prioritization of requirements 10](#_Toc159107215)

[9.1.1 Three-level Scale 10](#_Toc159107216)

[9.1.2 Prioritization of the requirements of Shobdobarta 10](#_Toc159107217)

[9.2 Traceability Matrix 10](#_Toc159107218)

**List of Tables**

[Table 1: Send Message 08](#_Toc159107223)

[Table 2: Receive Message 09](#_Toc159107224)

[Table 3: Traceability Matrix 10](#_Toc159107220)

List of Figure

[Figure 1:Use Case Diagram 7](file:///C:\Users\fazil\Downloads\SRS-for-NSTU-101-1.docx#_Toc127739025)

# Introduction

In today's interconnected world, communication is the cornerstone of social interaction and collaboration. Whether it's coordinating with friends, colleagues, or family members, the ability to engage in group conversations is paramount. However, traditional messaging platforms often lack the functionality to facilitate seamless group communication, leading to inefficiencies and frustrations. In response to this need, we present a solution tailored for the modern era: a group chatting application.

## Problem Statement

Many existing messaging platforms are designed primarily for one-on-one conversations, making group communication cumbersome and disjointed. Users often encounter difficulties in creating, managing, and participating in group chats efficiently. This can result in confusion, missed messages, and overall dissatisfaction with the user experience. Furthermore, the lack of robust features for group communication inhibits collaboration and hampers productivity in both personal and professional settings.

## Purpose

The purpose of this document is to outline the features and requirements of our group chatting application (Shobdobarta), addressing the shortcomings of existing messaging platforms. By providing a comprehensive solution for creating, viewing, sending, and receiving messages within group conversations, we aim to enhance the user experience and promote seamless communication among participants. This document will serve as a roadmap for the development team, guiding the implementation of key features and ensuring alignment with user needs and expectations.

## Project Scope

Our group chatting application is designed to cater to a diverse range of users, including individuals, teams, and communities. Registered users will have the ability to create groups, view group conversations, send messages, and receive real-time updates from group members. Overall, the scope of the project encompasses providing a user-friendly, feature-rich platform for seamless group communication and collaboration.

## Overview

Our application is designed to cater to a diverse range of users, each with their own unique perspectives and needs. With its versatile features and user-friendly interface, the app offers a seamless experience for creating new groups, viewing existing ones, and engaging in real-time messaging through socket programming.

# GroupWise System: GroupWise systems organize users into distinct groups based on criteria, facilitating efficient management and communication. They often allow group-specific settings, permissions, and communication channels.

# Live Base Communication: Live-based communication, such as chat applications, video conferencing, and live streaming platforms, allows instant interaction and feedback, promoting engagement and collaboration among users.

# Username Add: "Username add" is the process of adding a unique username to a system or application, essential for authentication, personalized experiences, and communication within the system.

# Stakeholders and Characteristics

## Users

They are the primary stakeholders as they will be using the application for communication and collaboration. Their needs and preferences should be central to the design and development process.

## Developers

Those responsible for building and maintaining the application, ensuring that it meets the functional requirements and is bug-free.

# Design and Implementation Constrains

We have employed design and implementation constraints to ensure the success of this project. It also refers to a tool that allows developers and testers to inspect and interact with the application's user interface (UI) elements.

## Programming Language

The programming languages and frameworks utilized for the real-time chat application are:  
  
Java: Primarily used for Android application development.  
XML: Utilized for defining the layout and UI elements in Android development.  
Node.js: Employed as the server-side runtime environment.  
Express.js: A minimal and flexible Node.js web application framework used for building the server-side application.  
[Socket.IO](https://socket.io/): A library that enables real-time, bidirectional, and event-based communication between web clients and servers, utilized for real-time chat functionality.

## Technical Constraints

The visual layout of the components that a user could interact with on a website or technical product is referred to as user interface design, or UI design. In other terms, it is a website's visual design.

## Hardware Interfaces

**Smartphone or Tablet:** The main hardware interface where users install and interact with the chat app.

**Touchscreen Display:** Users input messages and interact with the app through the touchscreen display.

**Physical Buttons:** Some devices may have physical buttons for navigation or input, though these are less common in modern smartphones.

**Network Interface:** Wi-Fi or mobile data connection is essential for sending and receiving messages in real-time.

1. Top of Form

# Requirement Specification

All the requirements based on the elicitation process are described in this section.

## Functional Requirement

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data the system should hold and the interfaces with the user.

### Send Message

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-1** | The user should be able to send message. | | |
| **Description** | Users should have the capability to send messages to groups they are a part of, facilitating communication and collaboration among group members within the app.. | | |
| **Stakeholders** | User | **Priority** | High |

### Receive Message

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-2** | User should be able to receive message. | | |
| **Description** | Users should have the capability to receive messages in the groups they are a part of, facilitating communication and collaboration among group members within the app. | | |
| **Stakeholders** | User | **Priority** | High |

### Add username

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-3** | User should be able to add a username | | |
| **Description** | Users should have a username to create their own identity | | |
| **Stakeholders** | User | **Priority** | Medium |

## Non-Functional Requirements

### Performance

|  |  |  |  |
| --- | --- | --- | --- |
| **NFR-1** | Performance | | |
| **Description** | The application should load quickly and be responsive to its user interactions. | | |
| **Stakeholders** | User | **Priority** | High |

### Scalability

|  |  |  |  |
| --- | --- | --- | --- |
| **NFR-2** | Scalability | | |
| **Description** | The application should be able to handle a high number of concurrent users. | | |
| **Stakeholders** | User | **Priority** | High |

## Usability Requirements

### Simple and Intuitive navigation

|  |  |  |  |
| --- | --- | --- | --- |
| **UR-1** | Simple and Intuitive navigation | | |
| **Description** | The application should have a simple, consistent and easy-to-use navigation structure. | | |
| **Stakeholders** | User | **Priority** | High |

### Clear and Consistent Visual Design

|  |  |  |  |
| --- | --- | --- | --- |
| **UR-2** | Clear and consistent visual design | | |
| **Description** | The application should have a clear and consistent visual design, which helps users understand the app's structure and organization. | | |
| **Stakeholders** | User | **Priority** | High |

## Security Requirements

### Secure Transmission

|  |  |  |  |
| --- | --- | --- | --- |
| **SR-1** | Secure Transmission | | |
| **Description** | Data transmission between the user's device and the server should be secured using HTTPS. | | |
| **Stakeholders** | User | **Priority** | High |

## Performance Requirements

### Responsiveness

|  |  |  |  |
| --- | --- | --- | --- |
| **PR-1** | Responsiveness | | |
| **Description** | The application should be responsive to user interactions and provide instant feedback for actions taken. | | |
| **Stakeholders** | User | **Priority** | High |

# Requirement Engineering Process

Requirements Engineering (RE) determines software requirements according to customer requirements or needs. Requirements engineering process includes requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management.

## Requirement Elicitation Techniques

Requirements elicitation is the practice of researching and finding system requirements for users, customers, and other stakeholders, also referred to as "requirement gathering". Requirement elicitation can be done by contacting participants directly or by doing some research, analysis and testing.

## Requirement Validation

Requirement validation ensures that the requirements are correct and reflect the quality you want from this program. In the beginning, our requirements looked good but when I read them and tried to work with them, they came out having ambiguities and gaps.

### Walkthrough

A walkthrough is a review of the requirement documents conducted by a team of stakeholders, developers, and experts. They review the requirements and provide feedback on their completeness, correctness, and consistency.

### Inspection

An inspection is a formal review process where each requirement is checked for completeness, accuracy, and consistency by a team of experts. The goal is to find and correct defects early in the development process.

# Use Case Diagram

Figure 1:Use Case Diagram

# Use Case Description

Table 1: Send Message

|  |  |  |
| --- | --- | --- |
| **Use Case** | Send Message | |
| **Goal** | Send message to the group. | |
| **Preconditions** | User is logged in to the system. | |
| **Success End Condition** | Successfully send the message to the group. | |
| **Failed End Condition** | Unable to send the message. | |
| **Primary Actors:**  **Secondary Actors:** | User  N/A | |
| **Trigger** | “Send” Button needs to be clicked. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Click on individual group. |
| 2 | User input text message. |
| 2 | User click to “Send” button. |
| 4 | Message is sent to the group and showing in the interface. |
| **Alternative Flows** | **Step** | **Branching Action** |
|  | N/A |
| **Quality Requirements** | **Step** | **Requirement** |
| 3 | Message is sent instantly to the group. |

Table 2: Receive Message

|  |  |  |
| --- | --- | --- |
| **Use Case** | Receive Message | |
| **Goal** | User receive the message from any group. | |
| **Preconditions** | User is logged in to the system. | |
| **Success End Condition** | User receive the message. | |
| **Failed End Condition** | Unable to receive the message. | |
| **Primary Actors:**  **Secondary Actors:** | User  System | |
| **Trigger** | Login to the system and also when click on individual group. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | User login to the application. |
| 2 | Last message in the group is showing to the user. |
| 3 | Users clicks on individual group. |
| 4 | User receives all the message. |
| 5 | System changes the password. |
| **Alternative Flows** | **Step** | **Branching Action** |
|  | N/A |
| **Quality Requirements** | **Step** | **Requirement** |
| 1 | Take not more than 1 second to load the messages. |

# Appendix

We’ve prioritized the functional requirements by following Three-level Scale technique.

### 7.1.1 Three-level Scale

When a Business Analyst categorizes the requirements in any of the ordering or ranking scale, it is subject to the analyst’s understanding of the business. Many analysts suggest that this method has some drawbacks and advocate methods that have more than one scale.

### Prioritization of the requirements of Shobdobarta

FR1-High priority: The user should be able to send message. As it is chat application everybody enter here to communicate with each other’s.

FR2-High priority: User should be able to receive message.

FR3-Medium priority: User should be able to add his/her username for knowing their own identity.

### Traceability Matrix

**Use Cases:**

UC1: Add nickname

UC2: Send Message

UC3: Receive Message

**Functional Requirements:**

FR1-The user should be able to send message.

FR2-User should be able to receive message.

FR3-User should be able to add a username.

*Table 3 :Traceability Matrix*

|  |  |  |  |
| --- | --- | --- | --- |
| **FR/UC** | **UC1** | **UC2** | **UC3** |
| **FR1** | 🗸 | 🗸 |  |
| **FR2** | 🗸 |  | 🗸 |
| ***FR3*** | 🗸 |  |  |

# References

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. Software Engineering 9th Edition by Lan Sommerville.
3. Requirements Engineering Fundamentals by Klaus Pohl