

# **Software Requirement Specification**

for

## **Notelog**

By:

Amaan Mithani – 19070122108

Ronak Malkan – 19070122100

Nitya Mehta - 19070122118

# Contents:

1. INTRODUCTION .....	3
1.1. Purpose .....	3
1.2. Definition.....	3
1.3. Overview .....	3
2. OVERALL DESCRIPTION .....	4
2.1. Product perspective .....	4
2.1.1 System interfaces .....	4
2.1.2 User interfaces.....	4
2.1.3 Hardware interfaces .....	4
2.1.4 Communications interfaces.....	4
2.2 Product functions .....	5
2.3 User characteristics.....	6
2.4 Constraints.....	6
2.5 Assumptions & dependencies.....	6
3. SPECIFIC REQUIREMENTS .....	
3.1 Functional Requirements .....	7
3.1.1 Admin.....	7
3.1.2 User.....	9
3.2 Non- Functional Requirements.....	10
3.2.1 Performance Requirements.....	10
3.2.2 Security Requirements.....	10
3.2.3 Software Quality Attributes.....	11
4. DIAGRAMS .....	12
4.1 Use Case .....	12
4.2 Activity Diagram .....	12
4.1.1 User Login.....	12
4.2.1 Create .....	14
4.2.2 Edit .....	15
4.2.3 Delete .....	16
4.3 Deployment Diagram .....	17

# 1. INTRODUCTION

## 1.1 Purpose

This document specifies all the requirements for the Notelog web application. These requirements relate to the functionality, constraints, performance, attribute and the system interface.

Notelog will have an easy to use interface connected directly to a central server. On the server, users can create notes using date and time that can be accessed from any device by logging on to one's account. It will have create, delete and edit operations for the notes.

## 1.2 Definitions

Log in- allows users to access their account.

Sign Up- allows users to create an account.

Create- create a new note.

Delete- remove an existing note.

Edit- modify an existing note.

View- to display the note.

Save- to store the created/modified note.

Share- send notes to others.

## 1.3 Overview

The rest of this document describes the system requirements for the Notelog.

## 2. OVERALL DESCRIPTIONS

### 2.1 Product perspective

People need a device independent way of saving notes or thoughts. Hence this web application will allow users to save their notes and access them from any device with internet connection. The notes can include anything characters, emojis, text. The note also provides space to input the date and if the users chooses not to input any date then while saving, the website itself will include the current date in the note.

#### 2.1.1 System interfaces

Notelog web application integrates two internal systems to provide functionality:

**Client:** Notelog has an interface to the user's client to receive user input.

**Network:** Notelog has an interface to the network in order to transmit information.

It uses servlets, Java language and servers to interact with users. Depending on the web browser some features may or may not be affected.

#### 2.1.2 User interfaces

Notelog includes several web pages, one for log in, one for sign up, one for viewing the list of notes, etc. It integrates all these web pages into one web application. It will utilize HTML, CSS, JS to make a frontend which will be user friendly and easy to use.

#### 2.1.3 Hardware interfaces

Notelog is web application which is hosted on a local server, it uses MYSQL database for storing user's notes. The server used for the same is Tomcat server which works with Java Language. There are no particular hardware requirements from user's side as long the device used by the user has active internet connection and updated web browsers.

#### 2.1.4 Communications interfaces

Communication between the user and the local server is done using HTTP protocol. User's web browser sends request to the local server and then the server depending on whether the request is

dynamic or static, sends them to servlets, this servlets then process request and according to instructions given in jsp files sends a response back to the user's web browser.

## 2.2 Product functions

### 1 Log in –

User need to login into his/her account in order for them to access their notes or create new notes.

### 2 Sign Up –

If user does not have an account, they need to sign up and create a new account to use the features provided.

### 3 Create –

This option allows the user to create a new note. When user clicks on button it sends a request and is processed by opening a new note on the screen which allows the user to type the required text.

### 4 Delete –

This option allows the user to remove/delete the selected note permanently from the database.

### 5 Edit –

This option allows the user to open the existing note and make changes required by the user.

### 6 View –

This option allows the user to open the exiting file only for viewing purpose. User cannot manipulate text using this option

### 7 Save –

This option allows the user to save the created/edited note on the database.

### 8 Share –

If the user wants to share the note this option allows them to do it via e-mail.

### 9 Theme –

It allows users to choose which theme they want to see the website in, for example: Matrix Mode might consist of black background colour and green text colour, Dark Mode might consist of black background and white text, Light mode might consist of white background and black text and so on.

## 2.3 User Characteristics

Basic knowledge of computer/mobile is adequate to use this web application. The user interface will be friendly enough to guide the user.

## 2.4 Constraints

Some features will suffer based on the version of the web browser used. Sufficient internet speed is necessary to be able to load the web page. Since the website is hosted locally and it's database also exists on one single device, memory is limited and hence if more users use this application it will decrease the memory available for each individual person.

## 2.5 Assumptions and dependencies

Notelog is device independent and will work on any up to date web browser. It is assumed that the user is familiar with an internet browser and also familiar with handling keyboard and mouse if application is accessed on computer/laptop, if accessed by mobile they should be familiar with touch screen.

### 3. SPECIFIC REQUIREMENTS

#### 3.1 Functional Requirements

##### 3.1.1 Admin

###### i. Login

Purpose: Provides admin authentication

Inputs: admin username and password.

Processing: The input is verified by checking if the member already exists in the database and is the admin.

Outputs:

A. Valid- Confirmation message that admin is logged in and admin profile will be displayed.

B. Invalid- Error message shown and not permitted access.

###### ii. Add users

Purpose: Enables administrators to add new users to database.

Inputs: User identity and user information

Processing: The input is verified by checking if the member already exists in the database.

Outputs:

A. Valid- Confirmation message that user successfully added to database.

B. Invalid- Error message shown, user not added.

###### iii. Change password

Purpose: Enables administrator to change password

Inputs: old password, new password, confirm password.

Processing: Old password is checked with the password stored earlier in database.

Outputs:

A. Valid- Confirmation message that password successfully changed.

B. Invalid- Error message shown, password not changed.

iv. **Remove user**

Purpose: Enables administrator to remove user

Inputs: Username of user

Processing: The input is verified by checking if the member already exists in the database.

Outputs:

A. Valid- Confirmation message that user successfully removed.

B. Invalid- Error message shown, user not removed/ not found.

v. **Manage database**

Purpose: Enables Admin to view all users in database.

Inputs- admin login

Processing: The input is verified by checking if the member already exists in the database and is the admin.

Outputs:

A. Valid- Confirmation message – Admin can manage database.

B. Invalid- Error message shown, access not permitted to manage database.



### 3.1.2 User

#### i. Login

Purpose: Enables user to login to the system

Inputs: username and password.

Processing: The input is verified by checking if the member already exists in the database.

Outputs:

- A. Valid- Confirmation message that user is logged in.
- B. Invalid- Error message shown of incorrect username or password.

#### ii. Signup

Purpose: Registration of a non-member.

Inputs: user name, password, confirm password

Processing: The input is validated using client side as well as server side validation. The client side validation will include check for username. The server side validation will involve checking if the username entered is already used by a member in the database. The appropriate error messages are displayed if the input is not acceptable

Outputs:

- A. Valid- The member is directed to the main page on successful sign up.
- B. Invalid- Error message shown if username already exists.

#### iii. Change password

Purpose: Enables user to change password

Inputs: old password, new password, confirm password.

Processing: Old password is checked with the password stored earlier in database.

Outputs:

- A. Valid- Confirmation message that password successfully changed.
- B. Invalid- Error message shown, password not changed.

## 3.2 Non- Functional Requirements

### 3.2.1 Performance Requirements

- i. The software shall support use of multiple concurrent users at a time.
- ii. The software should be compatible with all the operating systems.

### 3.2.2 Security Requirements.

- i. Administrator to have a secure username and password to prevent unwanted logins.
- ii. The system has an authorization mechanism for users to identify their personal profiles. Therefore, different users will have different authorization levels to access the data.
- iii. Password Setting Policies:
  - a) Avoid single words, or a word preceded or followed by a single number ( e.g., 'password1'). Hackers will use dictionaries of words and commonly used passwords to guess your password.
  - b) To increase complexity of the password, include upper- and lower-case letters, numbers, and special characters. A password should use at least 3 of these choices and should be of minimum 8-10 characters long. (e.g., a11Black\$).

### 3.2.3 Software Quality Attributes

Notelog intends to deliver availability, portability, maintainability, and reliability.

i. Availability: The software should always be available to the user. In case of any failure, an error message will be displayed.

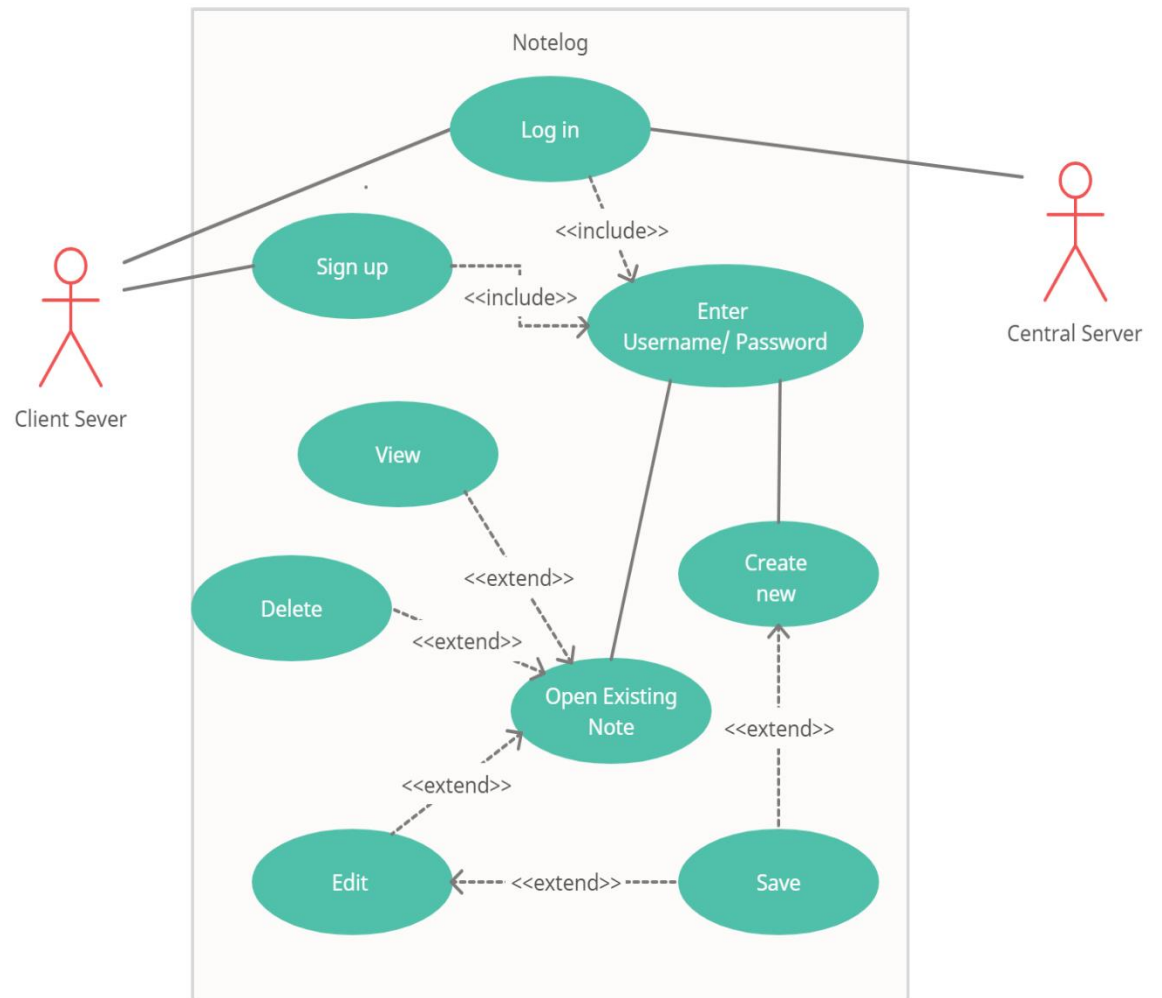
ii. Portability: The software will be accessible on any operating system and any widely used web browser.

iii. Maintainability: The software will require minimal maintenance. The database will be checked regularly to ensure data integrity.

iv. Reliability: The software has been tested and developed to ensure unexpected terminations will be prevented. Databases will be frequently checked to ensure there is no malicious data.

## 4. DIAGRAMS

### 4.1 Use Case

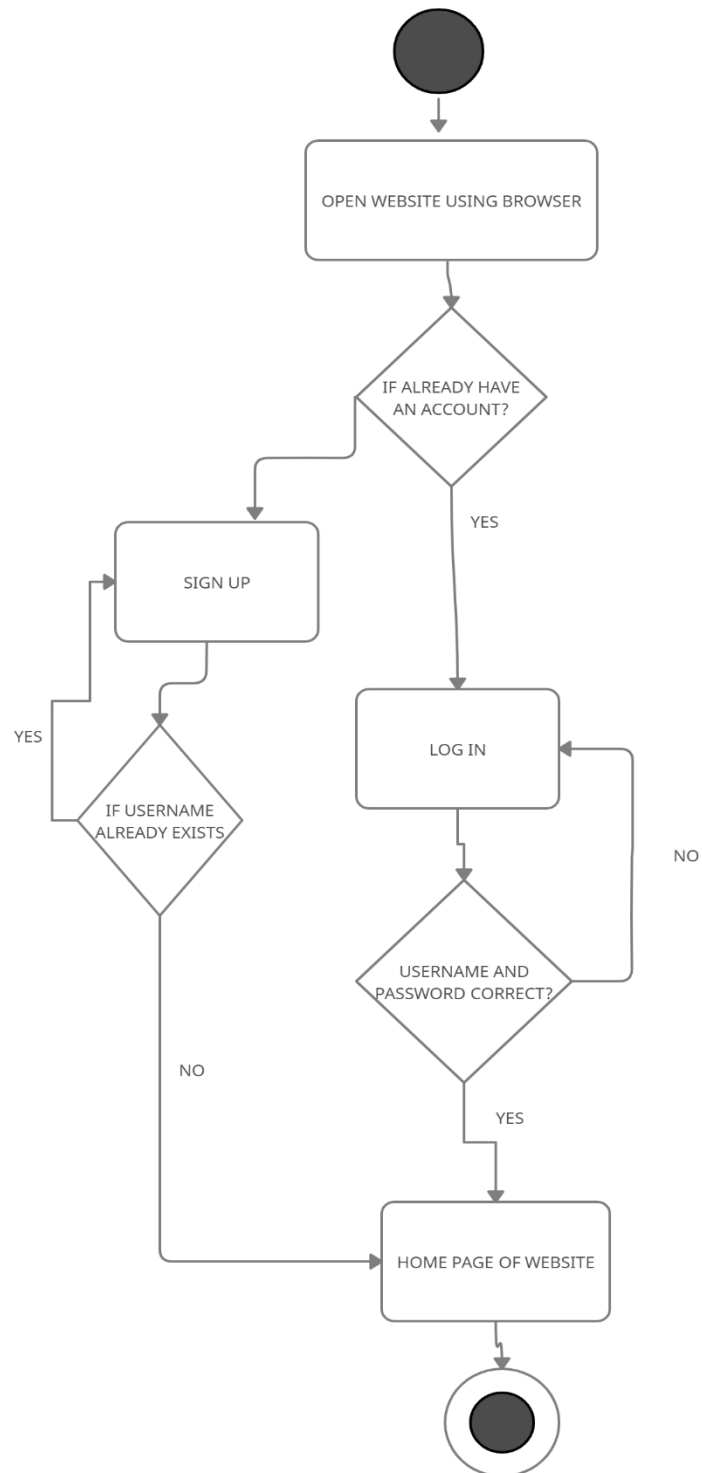


### 4.2 Activity Diagram

#### 4.2.1 User Login

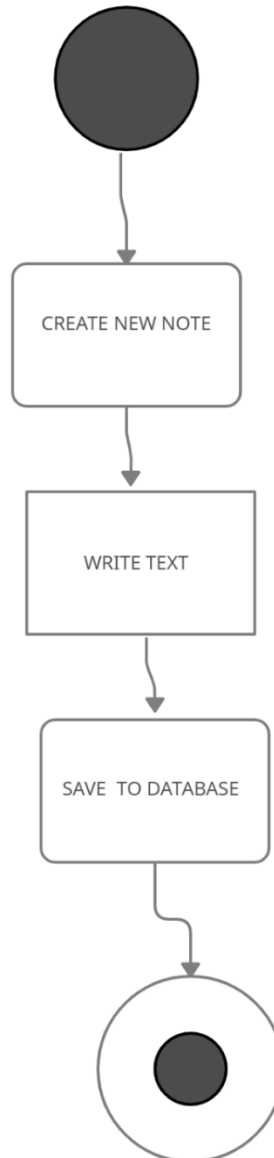
# NOTELOG

## ACTIVITY DIAGRAM FOR USER LOGIN



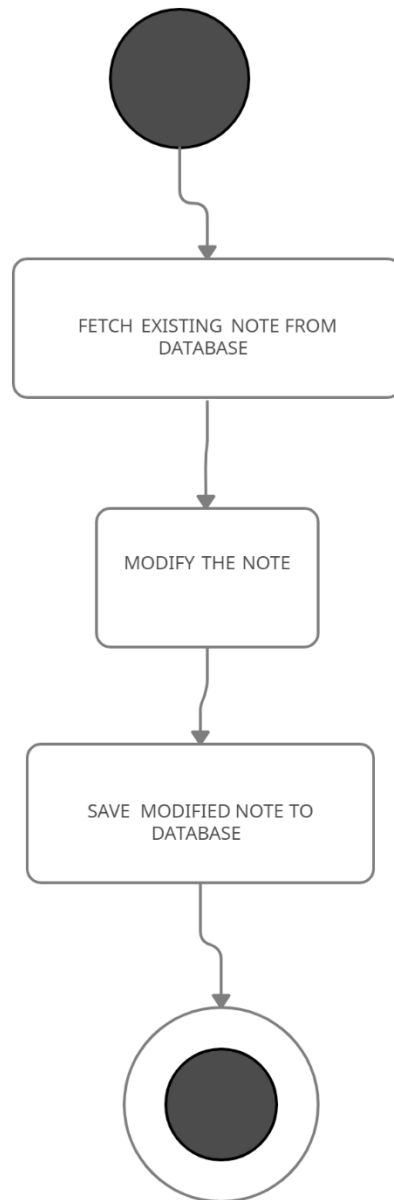
#### 4.2.2 Create Note

### ACTIVITY DIAGRAM FOR CREATING NOTE

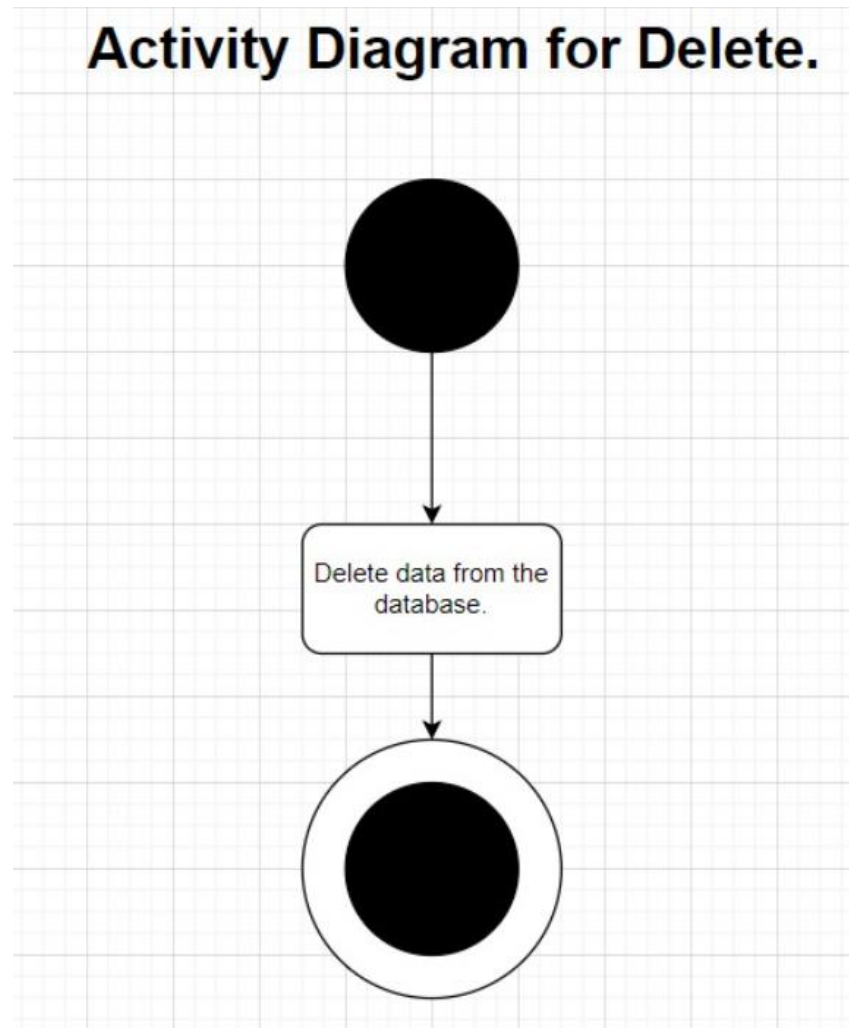


### 4.2.3 Edit Note

#### ***ACTIVITY DIAGRAM FOR EDITTING NOTE***



#### 4.2.4 Delete Note





### 4.3 Deployment Diagram

