# Software Requirements Specification

# For

# [Notes Web Application]

# [2016/11/19]

# [v1.0]

# Prepared by:

# Yugandhar Gangu

# **Table of Contents**

[Revision History](#__RefHeading___Toc216541873) 3

[1 Introduction 4](#__RefHeading___Toc216541874)

[1.1 Overview 4](#__RefHeading___Toc216541875)

[1.2 Goals and Objectives 4](#__RefHeading___Toc216541876)

[1.3 Scope](#__RefHeading___Toc216541877) 4

[2 General Design Constraints](#__RefHeading___Toc216541881) 5

[2.1 Product Environment](#__RefHeading___Toc216541882) 5

[2.2 User Characteristics](#__RefHeading___Toc216541883) 5

[2.3 Mandated Constraints](#__RefHeading___Toc216541884) 5

[3 Nonfunctional Requirements](#__RefHeading___Toc216541886) 6

[3.1](#__RefHeading___Toc216541887) [Operational Requirements](#__RefHeading___Toc216541888) 6

[3.2 Security Requirements](#__RefHeading___Toc216541890) 6

[3.3 External Interface](#__RefHeading___Toc216541895) 6

[3.3.1 User Interface](#__RefHeading___Toc216541896) 6

[3.3.2 Software Interface](#__RefHeading___Toc216541897) 6

[4 System Features](#__RefHeading___Toc216541898) 7

[4.1 **Required** Feature](#__RefHeading___Toc216541899) 7

[4.1.1](#__RefHeading___Toc216541904) [Use Case:](#__RefHeading___Toc216541905) 17

[4.1.2 Use Case: 2](#__RefHeading___Toc216541905) 7

[4.1.3](#__RefHeading___Toc216541906) [Use Case:](#__RefHeading___Toc216541905) 38

[4.1.4](#__RefHeading___Toc216541904) [Use Case:](#__RefHeading___Toc216541905) 48

[4.1.5 Use Case: 5](#__RefHeading___Toc216541905) 8

[4.1.6](#__RefHeading___Toc216541906) [Use Case:](#__RefHeading___Toc216541905) 69

[4.1.7](#__RefHeading___Toc216541904) [Use Case:](#__RefHeading___Toc216541905) 79

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| V1.0 | 2016/11/19 | Yugandhar Gangu | Basic Requirements |
|  |  |  |  |

# Introduction

## Overview

This document defines the requirement for the Notes Web Application that is being developed for Infoteria Corporation. The purpose of this document is to represent the system requirements in a readable way so that clients can understand them and verify them for correctness but with enough detail that I can design and implement a software system from them.

This document doesn’t address *project* issues such as schedule, development methods, development phases, deliverables and testing procedures. Those are addressed in a separate project document.

The Notes Web Application is a web-based tool for managing a list of notes. It provides a way for users to create, view, edit and delete the notes.

## Goals and Objectives

The three main goals of the innovative publishing product are:

1. Provide a simple mechanism for managing a list of notes.
2. Provide a security for view, edit and delete of notes.

## Scope

The innovative publishing system will solicit feedback from readers including written comments. Aggregate feedback from readers may be offered directly to other readers (i.e. articles might be rated), but unedited comments from readers will not automatically be made available to other readers. The reason for this is the quality of unedited comments is hard to control.

# General Design Constraints

## Product Environment

The Notes Web Application has built-in support for authentication. The Notes Web Application will use the user credentials and authenticate the users. The Notes Web Application also includes a relational database which will be used through the Sequelize interface.

## User Characteristics

**Users:** All users who registered in Notes Web Application can use this system. New users can registers and existing users can login then create, view, edit and delete notes.

## Mandated Constraints

The application will run on Node server. This platform chosen based on client requirements.

# Nonfunctional Requirements

## Operational Requirements

Usability: All users can sign up and use the Notes Web Application to manage his/her own notes.

## Security Requirements

Access to notes and features may be limited to specific users. Users can access his/her own notes. Other users can't able to access another user notes.

## External Interface

### User Interface

The user interface will be eye-catching and visually appealing. When users access their Notes application account, the interface will provide a smooth transition with Notes Web Application which has straightforward, understated look and feel. No training will be provided and it is expected that 99% of users will be able to use the application without any training.

### Software Interface

The Node server will work as server to serve for The Notes Web Application. The AngularJs will be interact with the browser and act as client. The web browser will communicate with the Node server and visualize to users.

# System Features

## Required Features

### Use Case: 1

***Description: User Sign up***

Actors: User

Basic Path

1. User inputs Full name, Email Id, Password and Confirm Password.
2. User click on Sign up button.
3. System will check users information and register new user or prompt a error message in case users already exists.
4. System exits.

### Use Case: 2

***Description: User Sign in***

Actors: User

Basic Path

1. User inputs Email Id, Password.
2. User click on Sign in button.
3. System will check users credentials and logged in into users account or prompt a error message in case of wrong credentials.
4. User can create, view, edit and delete notes.
5. User clicks Logout.
6. System exits.

### Use Case: 3

***Description: Create Note***

Actors: User

Basic Path

1. User login into his/her account with his/her credentials.
2. User click on Add Note on Header.
3. User inputs subject and note information.
4. User click on Save button.
5. Note will be saved and displays in notes list.
6. System exits.

### Use Case: 4

***Description: View Note***

Actors: User

Basic Path

1. User login into his/her account with his/her credentials.
2. User will be in notes screen.
3. At least one note has already created.
4. User click on view button on particular note.
5. Note view screen will be shown with information.
6. System exits.

### Use Case: 5

***Description: Edit Note***

Actors: User

Basic Path

1. User login into his/her account with his/her credentials.
2. User will be in notes screen.
3. At least one note has already created.
4. User click on edit button on particular note.
5. Note edit screen will be shown with information.
6. User can change only notes not the subject of the note.
7. User clicks on Save button.
8. Note will be updated with version number.
9. Updated note will be displayed in notes list.
10. System exits.

Alternative Path

1. User login into his/her account with his/her credentials.
2. User will be in notes screen.
3. At least one note has already created.
4. User click on view button on particular note.
5. Note view screen will be displayed.
6. User click on edit button
7. Note edit screen will be shown with information.
8. User can change only notes not the subject of the note.
9. User clicks on Save button.
10. Note will be updated with version number.
11. Updated note will be displayed in notes list.
12. System exits.

### Use Case: 6

***Description: Note List***

Actors: User

Basic Path

1. User login into his/her account with his/her credentials.
2. User will be in notes screen.
3. System exits.

Alternative Path

1. User login into his/her account with his/her credentials.
2. User click on Notes menu.
3. User will be in notes screen.
4. System exits.

### Use Case: 7

***Description: Change Password***

Actors: User

Basic Path

1. User login into his/her account with his/her credentials.
2. User clicks on Change Password from Menu.
3. Change Password screen will display.
4. User inputs Old Password, New Password and Re-enter New Password fields.
5. Users click on Update button.
6. System will update user password.
7. System exits.