**Software Requirements** **Specification**

**(SRS)**

**mNotes**

***The Best Note Keeping Application***

**Krasimir Kostov**

**Elena Avramova**

**Plamen Vladimirov**

**Nadezhda Metodieva**

**Georgi Angelov**

**Borislav Ivanchev**

**Andrei Kozarev**

**Ani Arsova**

SRS Table of Contents

[1. Introduction 3](#__RefHeading___Toc618_686983108)

[1.1 Purpose 3](#__RefHeading___Toc620_686983108)

[1.2 Scope 3](#__RefHeading___Toc622_686983108)

[2. Overall description 3](#__RefHeading___Toc624_686983108)

[2.1 Design 3](#__RefHeading___Toc626_686983108)

[2.1.1 Front end: 3](#__RefHeading___Toc1791_2815156087)

[2.1.2 Back-end: 3](#__RefHeading___Toc1793_2815156087)

[2.2 Operating Environment 3](#__RefHeading___Toc628_686983108)

[3. User Interface Requirements 4](#__RefHeading___Toc630_686983108)

[3.1 User Interface - Administration module: 4](#__RefHeading___Toc632_686983108)

[3.1.1 Login Screen: 4](#__RefHeading___Toc634_686983108)

[3.1.2 Registration Screen 4](#__RefHeading___Toc636_686983108)

[3.1.3 Profile Screen: 5](#__RefHeading___Toc638_686983108)

[3.2 User Interface - Functional module: 5](#__RefHeading___Toc640_686983108)

[3.2.1 Main Screen: (From Left to Right and from Top to Down): 5](#__RefHeading___Toc642_686983108)

[3.2.2 Note screen: 7](#__RefHeading___Toc644_686983108)

[4. Functional Requirements (Use Cases) 8](#__RefHeading___Toc646_686983108)

[4.1 Administration module Use Cases: 8](#__RefHeading___Toc664_686983108)

[4.2 Functional module Use Cases: 8](#__RefHeading___Toc666_686983108)

[4.2.1 Use Case FR4.2.1: Create a New Note 8](#__RefHeading___Toc1795_2815156087)

[4.2.2 Use Case FR4.2.2: Edit a Note 8](#__RefHeading___Toc1797_2815156087)

[4.2.3 Use Case FR4.2.3: Delete 9](#__RefHeading___Toc1799_2815156087)

[4.2.4 Use Case FR4.2.4: Archive 9](#__RefHeading___Toc1801_2815156087)

[4.2.5 Use Case FR4.2.5: Sort notes by status Done/ Not done 9](#__RefHeading___Toc1803_2815156087)

[4.2.6 Use Case FR4.2.6: Sort notes by due date 9](#__RefHeading___Toc1805_2815156087)

[4.2.7 Use Case FR4.2.7: Sort notes by Priority – ranged as 1st, 2nd and 3d 10](#__RefHeading___Toc1807_2815156087)

[4.2.8 Use Case FR4.2.8: Archived notes in drop down menu 10](#__RefHeading___Toc1809_2815156087)

[4.2.9 Use Case FR4.2.9: Search a note 10](#__RefHeading___Toc1811_2815156087)

[5. Non-Functional Requirements 11](#__RefHeading___Toc648_686983108)

[5.1 Performance Requirements 11](#__RefHeading___Toc650_686983108)

[5.2 Safety Requirements 11](#__RefHeading___Toc652_686983108)

[5.3 Security Requirements 11](#__RefHeading___Toc654_686983108)

[5.4 Software Quality Atributes 11](#__RefHeading___Toc656_686983108)

[6. Other Requirements 11](#__RefHeading___Toc658_686983108)

**SRS** Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **SRS-mNote\_v03** | 25.01.2019 |  | Version 03 |
| **SRS-NIM\_v02** | 22.01.2019 |  | Version 02 |
| **SRS-NIM\_v01** | 21.01.2019 |  | Version 01 |

# 1. Introduction

This section gives a scope description and overview of this Software Requirements Specification (SRS).

## 1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “mNotes“ software. It will describe the required features and functionality for the development of system. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## 1.2 Scope

The “mNotes” is a WEB-based responsive application which helps people to keep notes, to organize their tasks with notes which can notify them with remainders before the expire date and time, those note to be cheked as “done”, and to automatically be archived after the expired date and time if done. Notes should have the option to be categorized and tagged, edited, filtered by category or tag, The output is notes to be printed, converted to a PDF file or to be exported as CSV file.

# 2. Overall description

## 2.1 Design

### 2.1.1 Front end:

* HTML
* Bootstrap
* CSS

### 2.1.2 Back-end:

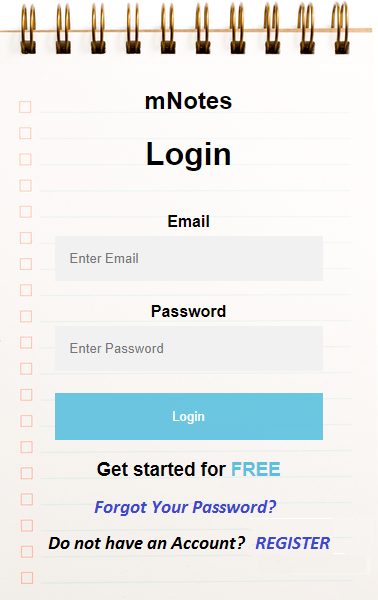
* PHP
* MySQL

## 2.2 Operating Environment

WEB Brouser

# 3. User Interface Requirements

## 3.1 User Interface - **Administration module:**



### 3.1.1 Login Screen:

1. Title "Log in";

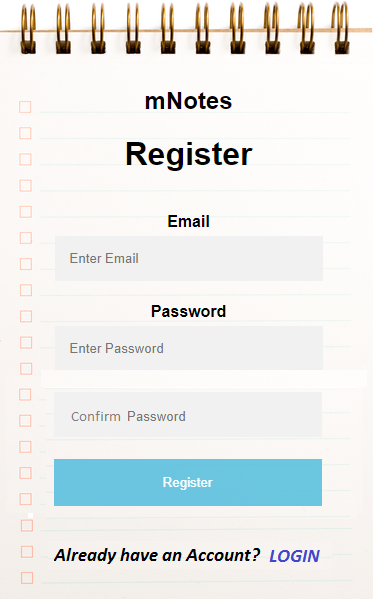
2. Field "email" (Username);

3. Field "Password";

4. Button "Log in";

5. Text "Do not have an account?" Register;

6. Link to Register screen from the above text "Register"



### 3.1.2 Registration Screen

1. Title "Register";

2. Field "email" (username);

3. Field "Password";

4. Field "Password Confirmation";

5. Radio-Button “Register”;

6. Text "Already have an account?" Login;

7. Link to Login screen from the above text "Login"

### 3.1.3 Profile Screen:

1. Title: Profile Screen”;

2. Field "email" (username);

3. Field "new email" (new username);

4. Field "New Password";

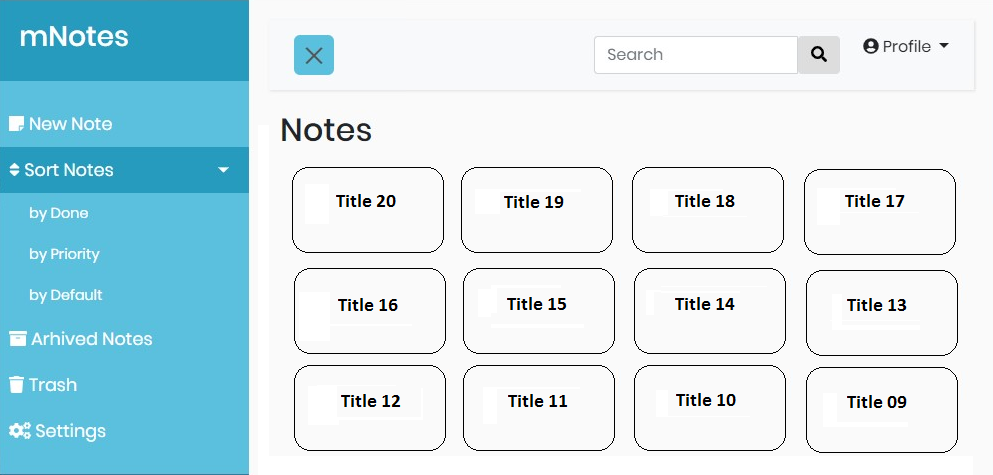
5. Field "Password Confirmation";

6. Radio-button “Edit”;

7, Radio-button “Save”.

## **3.2 User Interface - Functional module:**

### 3.2.1 Main Screen: (From Left to Right and from Top to Down):



**Button Open/Closed Side Bar Menu (Top-Left Side)**

1. Option "New Note";

2. Option “Sort Notes”; (sorting by Due Date, Priority, Done/Not Done);

3. Option “Archive Notes”;

4. Trash;

5. Settings.

**Search Field and Search Button(Top-Center):**

1. Field “Search” for input of Keyword;

2. Button “Search”;

**Profile Icon (Top-Right):**

1. Option “Edit Profile”;

2. Option “Sign Out”.

**Main Screen - Notes Icons sorted (Center):**

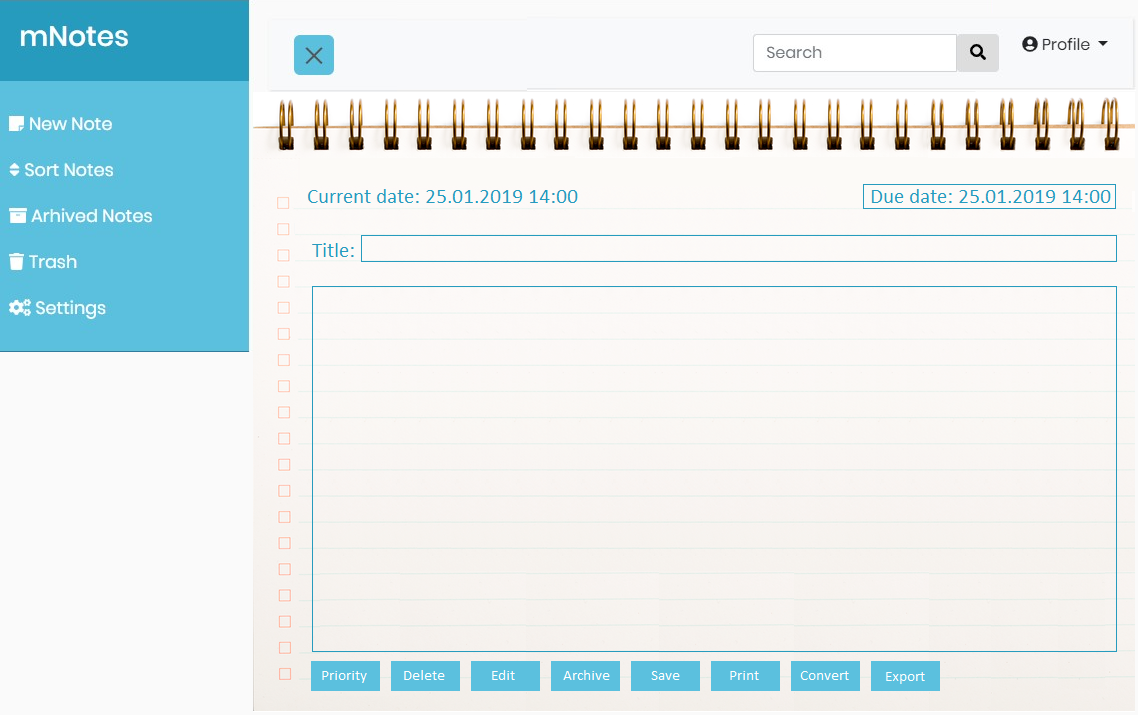
1. Notes Large Icons organized in rows and columns;

2. Number of Icons should be up to 6 on a row maximum down to 1 on a row minimum, depending on screen size and screen view, as well as user device type (responsive app.).

3. In vertical direction the number of rows is unlimited depending on overall number of Notes divided by number of Notes Icons in a row.

4. Default sorting is by Due Date of the Notes, from the Note with last Date and Time first.

5. Top-Down Scrolling should be available for the large lists of Notes.



### 3.2.2 Note screen:

1. Field Creation Date and Time (automatically takes system Date and Time on New Note – Save Button click);

2. Field ”Due Date and Time”

3. Field “Title”;

4. Field “Note Body”;

5. Button “Priority”;

6. Button “Delete”;

7. Button “Edit”;

8. Button “Archive”;

9. Button “Save”.

10. Button “PRINT NOTE”;

11. Button “Convert to PDF”;

12. Button “EXPORT” to text/csv;

# 4. Functional Requirements (Use Cases)

## 4.1 **Administration module Use Cases:**

## **4.2 Functional module Use Cases:**

### 4.2.1 Use Case FR4.2.1: Create a New Note

1. Given the user wants to create a New Note;
2. And is logged in;
3. And has the main screen with the Notes sorted by due date;
4. And choose from the left Side Bar menu “New Note” option;
5. And a Note window opens;
6. And current date field automatically takes system date and time;
7. And the user fills in the due date field;
8. And fills in the title field;
9. And fills in the text field or leaves it empty;
10. When the user clicks on Save button;
11. Then the Note window closes;
12. And the main screen appears with the New Note icon on the first place.

### 4.2.2 Use Case FR4.2.2: Edit a Note

1. Given the user wants to edit an existing Note;
2. And is logged in;
3. And has the main screen with the Notes sorted by due date;
4. And clicks on chosen icon of a Note;
5. And the Note window opens;
6. And the user presses the Edit button;
7. And edits text in the Title or Text field of the Note;
8. When the user clicks on Save button;
9. Then the Note is saved;
10. And the Note window closes;
11. And the main screen appears with existing Notes sorted by due date;

### 4.2.3 Use Case FR4.2.3: Delete

1. Given the user wants to delete a note;
2. And is logged in;
3. And has the main screen with the Notes sorted by due date;
4. When the user opens the Note to be deleted by clicking on it;
5. Then the user clicks on Delete button and removes the Note from the list.
6. (The Note deleted goes to Trash, it could be another table in Database);
7. And the main screen appears with existing Notes sorted by due date;

### 4.2.4 Use Case FR4.2.4: Archive

1. Given the user wants to put a note in the archive;
2. And is logged in;
3. And has the main screen with the Notes sorted by due date;
4. And opens a selected note by clicking on it;
5. And having the note opened in full size;
6. When clicks the Button Archive;
7. Then the note will no longer appear in the list (will not be seen in the main screen);
8. And will go to Archive.

### 4.2.5 Use Case FR4.2.5: Sort notes by status Done/ Not done

1. Given the user wants to sort notes;
2. And is logged in;
3. And has the main screen with the Notes sorted by due date;
4. And opens left Side Bar menu;
5. And choses the option Sort;
6. When the user clicks Sort by Done/Not Done option;
7. Then the user have the notes sorted by Done/Not Done starting with Done.

### 4.2.6 Use Case FR4.2.6: Sort notes by due date

1. Given the user wants to sort notes;
2. And is logged in;
3. And has the main screen with the Notes sorted by due date;
4. And opens left dropdown menu;
5. And choses the option Sort;
6. Then the user have the notes sorted by due date;
7. When the user enters the Sort menu and ticks the box by Due Date;

### 4.2.7 Use Case FR4.2.7: Sort notes by Priority – ranged as 1st, 2nd and 3d

1. Given the user wants to sort notes;
2. And is logged in;
3. And has the main screen with the Notes sorted by due date;
4. And opens the left Side Bar menu;
5. And choses the option Sort;
6. When the user clicks the Sort by Priority
7. Then the user have the notes sorted by 1st, 2nd and 3rd Priority;

### 4.2.8 Use Case FR4.2.8: Archived notes in drop down menu

1. Given the user is loged in
2. And has the main screen with the Notes sorted by due date;
3. And cliks on Note button;
4. And Side Bar menu is open;
5. And choses the option Archived notes
6. When the user clicks on the Archived notes
7. Then a list of archived notes appears

### 4.2.9 Use Case FR4.2.9: Search a note

1. Given the user is loged in
2. And has the main screen with the Notes sorted by due date;
3. And cliks on Note button;
4. And Side Bar menu is open;
5. When clicks on Archied Notes option;
6. Then a list of archived notes appears on Main Screen.

# 5. Non-Functional Requirements

## 5.1 Performance Requirements

The mNote application should provide excellent quality features as speed, short reaction time accuracy etc..

## 5.2 Safety Requirements

The mNote application should assures customer data confidentiality.

## 5.3 Security Requirements

The mNote application should be protected from unauthorized external or internal access with secure program codes and security solutions.

## 5.4 Software Quality Atributes

The mNotes should be easy to use and easy to learn application with user-friendly high intuitive user interface. As a responsive WEB Aplication mNotes should have high adaptability and flexibility. For example the main Screen of the application should be flexible enough to reflect screen size and/or screen view parameters of different user devices, delivering best possible readiness of the user notes content and the best access to navigation tools, menus and radio-buttons.

# 6. Other Requirements

1. The mNote application should create a positive ambiance provoking good emotions for the users trough it’s state of the art graphic design and ease of use.