

Міністерство освіти і науки України  
Національний Технічний Університет України  
«Київський Політехнічний Інститут»  
Навчально-науковий комплекс «Інститут прикладного системного аналізу»  
Кафедра системного проектування

Лабораторна робота No2  
з дисципліни  
“Проектування інформаційних систем”  
Опис передпроектної документації  
(Software requirement specification)

Виконав:  
студент групи ДА-72  
Черномаз В.С.  
В-28

Київ – 2020

**Мета роботи:** вивчити основні етапи створення передпроектної документації (SRS).

**Задача:** використати приклад SRS для створення передпроектної документації згідно індивідуальної теми для виконання лабораторних робіт.

**Завдання:**

1. Вивчити вимоги до передпроектної документації.
2. Скласти опис передпроектної документації для об'єкта проектування.
3. Скласти 5-7 приймальних тестів для ПО об'єкта проектування.
4. Оформити технічне завдання згідно опис передпроектної документації (використовувати рекомендації IEEE 830).

Notes App

Software Requirements Specification Version

1.0

16.11.2020

Chernomaz Vadym

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>iii</b>
<b>Document Approval</b>	<b>iii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Project Scope	2
1.5 References	2
<b>2. Overall Description</b>	<b>3</b>
2.1 Product Perspective	3
2.2 Product Features	3
2.3 Operating Environment	3
2.4 Design and Implementation Constraints	3
2.5 User Documentation	4
<b>3. Specific Requirements</b>	<b>5</b>
3.1 External Interfaces	5
3.1.1 Software Interfaces	5
3.1.2 Hardware Interfaces	5
3.2 Functional Requirements	5
3.2.1 Create a Note	5
3.2.2 Edit a Note	6
3.2.3 Delete a Note	6
3.2.4 View a Note	6
3.2.5 Settings	6
3.3 Performance Requirements	7
3.4 Security Requirements	7
3.5 Availability Requirements	7
3.6 Portability Requirements	7
3.7 Maintainability Requirements	7
3.8 Logical Database Requirements	7
<b>4. Tests</b>	<b>8</b>

## Revision History

Name	Date	Reason for changes	Version
Chernomaz V.S.	2020-11-16	Document creation	1.0

## Document Approval

Signature	Name	Title	Date
	Chernomaz V.S.		2020-11-16

# 1. Introduction

## 1.1 Purpose

The purpose of this SRS is to define functional and nonfunctional requirements, use cases that describe user interactions with the application to provide them with a perfect experience.

## 1.2 Document Conventions

- Application - “Notes” app, a main object of this SRS.
- End User - any person who uses the application.
- Functional Requirements - requirements for a special feature: how it works, where stores data, how provides user experience.
- Nonfunctional Requirements - requirements that can not be described as a feature, for example Security requirements.
- Front-end - part of the application end user interacts with.
- Back-end - server-side of the application. Can not be seen by the end user, main calculations happen there.
- FAQ - frequently asked questions.
- Software Requirements Specification - a document which completely describes all of the functions of a proposed system and constraints under which it must operate.

→ **X. Bold Text** - name of the paragraph

→ **X.X Bold Text** - name of the subparagraph

→ *Italic Text* - name of the resource

## 1.3 Intended Audience and Reading Suggestions

This document is written for developers, designers, marketing staff, testers and managers. Rest of the documents contain requirements for the application, its features and implementation details. Document is divided into paragraphs and subparagraphs which can be easily reached via Table of Contents.

## **1.4 Project Scope**

The application name is “Best Note”. Its main purpose is to compete with existing note-taking apps and show end users that it is real to take notes everywhere easily and quickly. The application itself is an app for Android and iOS devices which can also synchronize with the web version of the app. The main benefit of the app is that it provides intuitive user experience and allows users to minimize their taps on the screen. The application will be able to organize notes by categories, tags and dates so it can be used for such systems as “Getting Things Done”. Its trial period and fixed price also will allow users to try the application and pay once to obtain all its features.

## **1.5 References**

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

## **2. Overall Description**

### **2.1 Product Perspective**

The application fulfills user needs in a way of providing intuitive user experience and providing an ability to quickly take a note and do not be distracted by opening and creating a new note. A big part in this process takes synchronization among devices.

### **2.2 Product Features**

1. Create, Edit, Delete notes
2. Create a widget at the main page of the device to easily create a note
3. Organize notes by categories, tags, dates
4. Synchronize notes among devices

### **2.3 Operating Environment**

Back-end of the app will be put into containers and launched at DigitalOcean cloud platform. MongoDB will be used as a main storage system. Front-end part of the application can be launched at Android devices and iOS devices.

Minimal Android version requirement - Android 10.

Minimal iOS version requirement - iOS 10.3.4.

### **2.4 Design and Implementation Constraints**

Back-end part programming language - Go

Android part programming language - Kotlin

iOS part programming language - Swift

Web version programming language - JavaScript

Database - MongoDB

Connection between front-end part and back-end part will be via REST architecture.



## **2.5 User Documentation**

Documentation for end users will be in the form of a web page and tutorial right inside the app. Also it will be a web page with FAQ and a support chat.

### **3. Specific Requirements**

#### **3.1 External Interfaces**

##### **3.1.1 Software Interfaces**

The mobile device has to have an iOS or Android operating system. Minimal requirements for operating systems see in 2.4. Also to obtain full functionality of the app it is necessary for device to have connection with the Internet to synchronize user's data.

##### **3.1.2 Hardware Interfaces**

Hardware interface is represented in the form of a mobile device or personal computer. It is a must for mobile device to support touch input.

#### **3.2 Functional Requirements**

##### **3.2.1 Create a Note**

A note can be created in two ways: via widget or in the app after pressing a button "Create Note".

**Widget:** widget is represented in the form of a window and a button "Save" below it. User can type a note and save it immediately with the button. All the characteristics of the note will be empty.

**App:** user enters the app and sees a "Create Note" button in the upper right corner. After the user presses the button one is presented with a window where a note can be entered. Above the text editor user can see a panel of additional characteristics of a note: due date, category, tag. Saving is executed after pressing a button "Save" in the upper right corner.

##### **3.2.2 Edit a Note**

A note can be edited only via app. After the user enters the app one sees a list of notes. After swiping the note to the left a window of the editor is opened where the user can change the text of the note or its characteristics.

### **3.2.3 Delete a Note**

A note can be deleted only via app. After the user enters the app one sees a list of notes. After swiping the note to the right it is deleted. By default there is a confirmation alert “Are you sure you want to delete the note?” which can be disabled in settings.

### **3.2.4 View a Note**

A note can be viewed only via app. After the user enters the app one sees a list of notes. After tapping on the note a window with note information is opened. The note can be edited if the user presses the button “Edit Note” in the upper right corner. To come back to the note list user has to press the button “Back” in the upper left corner.

### **3.2.5 Settings**

There are several parameters which can be managed in settings: account, theme color, confirmation alert. Account parameter is a text field with the user's email after entering of which synchronization is executed automatically. Theme color parameter specifies what theme is used in the app right now, there are two options: light and dark. Confirmation alert parameter is represented as a switch which turns on or turns off the alert.

## **3.3 Performance Requirements**

The app itself does not have specific performance requirements except working smoothly and not to exit suddenly. There are requirements to the server which contains data and is responsible for synchronization between devices. It is necessary for the server to process two thousands requests per minute.

## **3.4 Security Requirements**

All the user's personal data is secured and encrypted, all the passwords are hashed. Databases have passwords and LDAP provides access only to authorized users.

### **3.5 Availability Requirements**

The app is absolutely free and can be downloaded via Google Play or App Store.

### **3.6 Portability Requirements**

The app is ported to different systems as was mentioned above: Android, iOS, Web.

### **3.7 Maintainability Requirements**

The app is always maintained and supported. New versions of the app will be supplied once two months.

### **3.8 Logical Database Requirements**

All the information will be stored in MongoDB. It has two main collections: accounts and notes. Accounts' collection record contains the user's email, hashed password and name. Notes collection record contains notes's text, categories, tags, date.

## 4. Tests

The main goal	Preliminary requirements	Steps to perform the test	Expected result	Steps to revert the test
Create a note via app	The app is installed to the mobile device	Open the app; Press the button “Create Note”; Enter the note text; Press the button “Save Note”;	The note is displayed in the note list	Delete the note, see test 4
Create a note via widget	The app is installed to the mobile device	Open the screen where widget is located; Enter the note text in the editor; Press the button “Save Note”;	The note is displayed in the note list	Delete the note, see test 4
Edit a note using swipe	The app is installed to the mobile device; The note is created	Open the app; Swipe the note you want to edit to the left; Edit the note; Press the button “Save Changes”	The note is changed	Repeat steps to perform the test but insert the old note data
Edit a note after viewing it	The app is installed to the mobile device; The note is created	Open the app; Tap at the note and view it; Press the button “Edit Note”; Edit the note; Press the button “Save changes”	The note is changed	Repeat steps to perform the test but insert the old note data
Delete a note	The app is installed to the mobile device; The note is created	Open the app; Swipe the note you want to delete to the right;	The note is deleted	It is necessary to create a new note, see test 1