*SoftUni blogging software*

Version *1.0*

*18.03.2019*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 0.1.1 | Angela Teneva | 15.03.2019 | <name> | <dd.mm.yy> | draft |
| 0.1.2 | Ivan Stalev | 17.03.2019 | <name> | <dd.mm.yy> | Non-Functional Requirements |
| 1.0 | Angela Teneva, Ivan Stalev | 18.03,2019 | <name> | <dd.mm.yy> | Software Requirements Specification |

Table of Contents

[List of Figures 4](#_Toc5032540)

[1. Introduction 5](#_Toc5032541)

[1.1. Purpose 5](#_Toc5032542)

[1.2. Scope of Project 5](#_Toc5032543)

[1.3. Glossary 5](#_Toc5032544)

[1.4. References 6](#_Toc5032545)

[1.5. Overview of Document 6](#_Toc5032546)

[2. Overall Description 7](#_Toc5032547)

[2.1 System Environment 7](#_Toc5032548)

[2.2 Product Functions 8](#_Toc5032549)

[2.2.1. User Types 8](#_Toc5032550)

[2.2.2. Page Content 8](#_Toc5032551)

[2.2.3. Articles 8](#_Toc5032552)

[2.3. Constraints 9](#_Toc5032553)

[2.3.1. General Constraints 9](#_Toc5032554)

[2.3.2. Format Fields Constraints 9](#_Toc5032555)

[2.4. Assumptions and Dependencies: 10](#_Toc5032556)

[2.5 Functional Requirements 10](#_Toc5032557)

[**2.5.1** Visitor Use Cases 10](#_Toc5032558)

[**2.5.2** User Use Cases 12](#_Toc5032559)

[**2.5.3** Author Use Cases 15](#_Toc5032560)

[**2.5.4** Admins Use Cases 17](#_Toc5032561)

[2.6 User Characteristics 19](#_Toc5032562)

[2.7 Non-Functional Requirements 20](#_Toc5032563)

[3. Requirements Specification 20](#_Toc5032564)

[3.1 External Interface Requirements 20](#_Toc5032565)

[3.2 Functional Requirements 20](#_Toc5032566)

[3.2.1 Open Blog Home Page 20](#_Toc5032567)

[3.2.2 Read Article 20](#_Toc5032568)

[3.2.3 Register Account 21](#_Toc5032569)

[3.2.4 User Login 21](#_Toc5032570)

[3.2.5 Create Article 21](#_Toc5032571)

[3.2.6 Logoff 22](#_Toc5032572)

[3.2.7 Edit Article 22](#_Toc5032573)

[3.2.8 Delete Article 22](#_Toc5032574)

[3.2.9 View Users 23](#_Toc5032575)

[3.2.10 Edit User 23](#_Toc5032576)

[3.2.11 Delete User 24](#_Toc5032577)

[3.3 Detailed Non-Functional Requirements 24](#_Toc5032578)

[3.3.1 Logical Structure of the Data 24](#_Toc5032579)

[3.3.2. Security 25](#_Toc5032580)

[3.3.3. Availability 25](#_Toc5032581)

[3.3.4. Performance 25](#_Toc5032582)

[3.3.5. Reliability 25](#_Toc5032583)

[3.3.6. Usability 25](#_Toc5032584)

[4. Other Requirements 26](#_Toc5032585)

[Appendix A SoftUni Blog system views of pages 26](#_Toc5032586)

[4.1 Home Page 26](#_Toc5032587)

[4.2 Register Page 26](#_Toc5032588)

[4.3 Log in Page 27](#_Toc5032589)

[4.4 Create Article Page 27](#_Toc5032590)

[4.5 Read Article Page 28](#_Toc5032591)

[4.5 Edit Article Page 28](#_Toc5032592)

[4.7 Admin pages 28](#_Toc5032593)

[4.7.1 Users Page 28](#_Toc5032594)

[4.7.2 Edit user Page 29](#_Toc5032595)

# List of Figures

[Figure 1 - System Environment 7](#_Toc3821281)

[Figure 2 - Visitor Use Cases 11](#_Toc3821282)

[Figure 3 - User Use Cases 13](#_Toc3821283)

[Figure 4 - Author Use Cases 16](#_Toc3821284)

[Figure 5 - Admins Use Cases 18](file:///D:\SoftUni\QA_Automation\TeamWork\QA-Automation\Documentation\SRS-Blog1-0.docx#_Toc3821285)

[Figure 6 - Logical Structure of the Article Data 27](file:///D:\SoftUni\QA_Automation\TeamWork\QA-Automation\Documentation\SRS-Blog1-0.docx#_Toc3821286)

# 1. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the SoftUni Blog System. It will explain the goals and features of the system, what the system will do, the interfaces of the system, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

## 1.2. Scope of Project

The main goal of this software system is a SoftUni blog website in which students and university lectors (called "Authors") from the Software University are provided with tools to share their thoughts and publish articles in the Web. They may discuss courses and programming topics in general, using the simplified article publishing of the system. The blog system is public and easy to understand and use. It allows read access to any visitor who is interested in SoftUni events, resourses and university life.

More specifically, this system is designed to allow an Author to publish articles to the blog website as well as edit their own articles. User accounts and authorization are administered by Admins.

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Article | The document that is published in the system and posted to the public website. |
| User | Registered Reader or Author. |
| Author | User, authorized to publish articles in the system. |
| Reader | User, authorized to read articles. |
| Visitor | Non-registered site visitor - allowed to read articles. |
| Admins | User who administrates the SoftUni Blog System. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Database | Collection of all the information monitored by this system. |

## 1.4. References

*29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering*

## 1.5. Overview of Document

This SRS is aimed at specifying requirements of software to be developed.

**Chapter 2** - "Overall Description" of this document gives an overview of the functional and data requirements of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

**Chapter 3** - "Requirements Specification" is written primarily for the developers and describes in technical terms the details of the functionality of the blog system application.

Both sections of the document describe the software product in its entirety, but are intended for different audiences and thus use different terms. We are going to focus on describing the system in terms of product perspective, product functions, user characteristics and dependencies.

Finally, specific requirements for the system, performance requirements, and other requirements are outlined.

**Appendix A** contains immages of the SoftUni Blog system pages.

# 2. Overall Description

## 2.1 System Environment

The Web Publishing System has three active actors and one cooperating system.

The Author, Reader, or Visitor access the SoftUni Blog through the Internet.

Create   
Read   
Edit   
Delete

Create Read Edit Administer

Read

Author

Admins

Visitor

Reader

Stakeholder

**Article**

**Article**

Read

Not a developer, with an interest in the Blog System

Data Base

SRS

SoftUni Blog System

Figure - System Environment

## 2.2 Product Functions

The main purpose of the blog is to meet students, lectors and fans of SoftUni in order to discuss courses and programming in general.

There will be the following pages:

1. Home Page
2. Register Page
3. Log in Page
4. Create Article Page
5. Edit Article Page
6. Manage Profile Page
7. For Admin users there will be the below extra pages

- Users Page

- Edit user Page

Detailed look of these pages is discussed in section 4 of this document.

### 2.2.1. User Types

There will be three types of users: Regular (not logged in), Logged user and Admin.

* Regular users will be able to read articles.
* Logged users will be able to read articles, create new article, edit and delete their own articles.
* Admin users will have the rights of logged user, but they will also be able to edit and delete other user’s articles. Admin user will be able to edit and delete user profiles. This includes providing other users with admin access rights.

### 2.2.2. Page Content

Each page will have header, page content and footer. The blog logo - “SOFTUNI BLOG” should be visible in the header on every page. Upon click on the logo users will be redirected to home page. “© 2019 - SoftUni Blog” text should be in the footer on every page.

### 2.2.3. Articles

Each article will have title, content, author and date of creation.

## 2.3. Constraints

### 2.3.1. General Constraints

- Application requires Internet connection.

- The application must be Web-based.

- The product will only work within a particular network environment

- The system should be up 24/7

### 2.3.2. Format Fields Constraints

*“Email address”* field:

Emails are considered to be in format <**user**>**@**<**host**>, where:

**<user>** is a sequence of letters and digits, where '.', '-' and '\_' can appear between them.

**<host>** is a sequence of at least two words, separated by dots '.'. Each word is sequence of letters and can have hyphens '-' between the letters

*“Full Name” field:*

- <**full name>** is a sequence of letters, special symbols, blank spaces and digits

- Small and capital letters from Latin alphabet

- Minimum length of 3 characters

- Maximum length of 50 characters

- Each name must start with small or capital letter

*“Password” field:*

- A minimum of 6 characters is required.

- Maximum length of 50 characters

- Small and capital letters from Latin alphabet

- Password can be mix of uppercase and lowercase characters, numbers and special characters.

*“Confirm password” field:*

- Exact match with <**Password**> field input

*“Current password” field:*

- Exact match with <**Password>** field input upon login

*“Article Title” field:*

- Minimum length of 5 characters

- Maximum length of 50 characters

*“Article Content” field:*

- Minimum length of 100 characters

- Maximum length of 2000 characters

## 2.4. Assumptions and Dependencies:

The availability and load time of the Blog depends on user’s internet connection strength which also depends on the media from which the product is run.

The performance depends upon hardware components of the user’s machine.

## 2.5 Functional Requirements

This section outlines the use cases for each of the active actors separately. The reader, the author and the visitor have only one use case.

### **2.5.1** Visitor Use Cases

The Visitor has the following use cases:

Visitor

Read Article

Register

Open Blog Website

Figure - Visitor Use Cases

#### Use case: Visitor Open Blog Home Page

Diagram:

Visitor

Open Blog Website

**Brief Description**

The Visitor should be able to access the SoftUni Blog Website

**Initial Step-By-Step Description**

Given that the Visitor opens a web browser

When the Visitor navigates to the SoftUni Blog Website address

Then the Blog Home Page is open

**Xref:** Section 3.2.1, Open Blog Home Page

#### Use case: Visitor Reads Article

Diagram:

Visitor

Read Article

**Brief Description**

The Visitor should be able to read articles

**Initial Step-By-Step Description**

Given that the Visitor opens the SoftUni Blog Website

And there exists at least one article

When the Visitor chooses an article from the list

And the Visitor clicks on its link

Then the selected article opens in the same browser window.

**Xref:** Section 3.2.2, Read Article

#### Use case: Visitor Registers Account

Diagram:

Visitor

Register

**Brief Description**

The Visitor should be able to register an Account, in order to login to the System.

**Initial Step-By-Step Description**

Given that the Visitor opens the Online Blog Website

When the Visitor chooses the Register option

And the Visitor populates necessary fields in the Register form

And the Visitor submits the Register form to the System

Then an account is created for the Visitor

**Xref:** Section 3.2.3, Register Account

### **2.5.2** User Use Cases

The User has the following use cases:

Login

User

Create Article

Read Article

Logoff

Figure 3 - User Use Cases

#### Use case: User Login

Diagram:

User

Login

**Brief Description**

The User should be able to Login, in order to use the System according to his credentials.

**Initial Step-By-Step Description**

Given that the User opens the SoftUni Blog Website

And the User has valid credentials

When the User enters the credentials to Login page

And the User clicks Login button

Then the User is logged in

**Xref:** Section 3.2.4, User Login

#### Use case: User Create Article

User

Create Article

Diagram:

**Brief Description**

As a User I want to be able to create a new article.

**Initial Step-By-Step Description**

Given that the User opens the SoftUni Blog Website

And the User has an account with the System

And the User is logged-in

When the User chooses Create Article option

And the User populates the necessary fields in the Create Article form

And the User clicks the Create button

Then the Article is created and is published in the blog.

**Xref:** Section 3.2.5, Create Article

#### Use case: User Read Article

User

Read Article

Diagram:

**Brief Description**

The User should be able to read articles

**Initial Step-By-Step Description**

Given that the User opens the SoftUni Blog Website

And there exists at least one article

When the User chooses an article from the list

And the User clicks on its link

Then the selected article opens in the same browser window.

**Xref:** Section 3.2.2, Read Article

**Use case**: User Logoff

Diagram:

User

Logoff

**Brief Description**

The User should be able Logoff from the System options

**Initial Step-By-Step Description**

Given that the User opens the SoftUni Blog Website

And the User is logged-in

When the User clicks Log off Button

Then the User is logged off the System

**Xref:** Section 3.2.6, Logoff

### **2.5.3** Author Use Cases

The Author has the following use cases:

Create Article

Author

Read Article

Edit Article

Delete Article

Figure - Author Use Cases

#### Use case: Author Edit Article

Diagram:

Author

Edit Article

**Brief Description**

The Author should be able to edit their articles, so that they can alter their content on the Blog System

**Initial Step-By-Step Description**

Given that the Author opens the SoftUni Blog Website

And the Author has valid credentials

And the Author is logged-in

And the Author has at least one Article created

When the Author enters own article

And the Author clicks Edit button

And the Author edits the Article content

And the Author clicks Edit button

Then the Article content changes according the edits and is re-saved to the Blog System

**Xref:** Section 3.2.8, Edit Article

#### Use case: Author Delete Article

Author

Delete Article

Diagram:

**Brief Description**

The Author should be able to delete his article and the content in them from the Blog System.

**Initial Step-By-Step Description**

Given that the Author opens the Online Blog Website

And the Author has valid credentials

And the Author is logged-in

And the Author has at least one Article created

When the Author enters own article

And the Author clicks Delete button

And the Author confirms with clicking Delete button again

Then the Article is deleted from the System.

**Xref:** Section 3.2.9, Delete Article

### **2.5.4** Admins Use Cases

The Admins has the following use cases:

Figure - Admins Use Cases

Admins

View Users

Edit User

Delete User

#### Use case: View Users

Diagram:

Admins

View Users

**Brief Description**

Admins should be able to view the list of all registered users in the SoftUni Blog System.

**Initial Step-By-Step Description**

Given that the Admins is logged-in the SoftUni Blog Website

When the administrator clicks the Admins link

Then the "Manage Users" page opens in the same browser window.

List of all SoftUni Blog users is displayed with their ID and Email.

Links to Edit and Delete actions are enabled, related to each user.

**Xref:** Section 3.2.10, View Users

#### Use case: Edit Users

Diagram:

Admins

Edit Users

**Brief Description**

The Admins should be able to edit the account of every registered user in the SoftUni Blog System.

**Initial Step-By-Step Description**

Given that the Admins is logged-in the SoftUni Blog Website

And the "Manage Users" page is open.

List of all SoftUni Blog users is displayed with their ID, Email.

"Edit" and "Delete" links for each user account are enabled.

Admins clicks the "Edit" link.

The Edit Page is open in the same browser window.

The Admins updates the user's data and/or the User status (from user to admin and vice versa).

And the Admins clicks Edit button

Then the User account changes according to the edits and is re-saved to the Blog System.

**Xref:** Section 3.2.11, Edit User

#### Use case: Delete User

Diagram:

Admins

Delete User

**Brief Description**

The Admins should be able to delete the account of every registered user in the SoftUni Blog System.

**Initial Step-By-Step Description**

Given that the Admins is logged-in the SoftUni Blog Website

And the "Manage Users" page is open

List of all SoftUni Blog users is displayed with their ID, Email

At least one user exists in the list

"Edit" and "Delete" links for each user account are enabled.

Admins clicks the "Delete" link of an user in order to delete the user's account

The Delete Page opens in the same browser window.

And the Admins clicks Delete button

Then the User account is deleted.

Admins is redirected to "Users" page.

**Xref:** Section 3.2.12, Delete User

## 2.6 User Characteristics

The User/Visitor is expected to be Internet literate and have basic website navigation skills. The main screen of the SoftUni Blog Website will display the list of all existing articles and a menu bar.

## 2.7 Non-Functional Requirements

The SoftUni Blog will be deployed on a server with high speed Internet capability. The physical machine to be used will be determined later. The speed of the User’s connection will depend on the hardware used and the software characteristics of this system.

# 3. Requirements Specification

## 3.1 External Interface Requirements

User interface to the SoftUni Blog System should be available at the university Web Site via Internet.

## 3.2 Functional Requirements

The logical structure of the data is described in Section 3.3.1.

### 3.2.1 Open Blog Home Page

|  |  |
| --- | --- |
| **Use Case Name** | Open Blog Home Page |
| **XRef** | SRS Section 2.5.1, Visitor Open Blog Home Page |
| **Trigger** | The Visitor decides to open Online Blog Website |
| **Precondition** | Web browser is open |
| **Basic Path** | 1. The Visitor navigates to Online Blog Website address |
| **Alternative Paths** | None |
| **Post condition** | The Blog Home Page is open in browser |
| **Exception Paths** |  |
| **Other** |  |

### 3.2.2 Read Article

|  |  |
| --- | --- |
| **Use Case Name** | Read Article |
| **XRef** | Section 2.5.1, Visitor Read Article;  Section 2.5.2, User Read Article |
| **Trigger** |  |
| **Precondition** | The Blog Website is open  At least one Article exists in Blog |
| **Basic Path** | 1. The Reader chooses an Article from Article list by clicking on Article’s title |
| **Alternative Paths** | 1. The Reader goes to Homepage 2. The Reader chooses an Article from Article list by clicking on Article’s title |
| **Post condition** | The selected article is open in the same browser window |
| **Exception Paths** | The Reader may leave the article/page at any time |
| **Other** |  |

### 3.2.3 Register Account

|  |  |
| --- | --- |
| **Use Case Name** | Register Account |
| **XRef** | Section 2.5.3, Visitor Register Account Section 2.5.1, Visitor Register Account |
| **Trigger** | The Visitor clicks on Register link. |
| **Precondition** | The Blog Website is open  Visitor’s credentials are valid |
| **Basic Path** | 1. Visitor populates with valid entries fields for:  * Email * Full Name * Password * Confirm password  1. Visitor clicks on Register button |
| **Alternative Paths** |  |
| **Post condition** | Visitor is registered as a User with the System |
| **Exception Paths** | The attempt may be abandoned at any time  If the Visitor enters non-unique Full Name and/or Email, should not be able to register  If the Visitor enters invalid information in any field, should not be able to register |
| **Other** |  |

### 3.2.4 User Login

|  |  |
| --- | --- |
| **Use Case Name** | User Login |
| **XRef** | Section 2.5.2, User Login |
| **Trigger** | The User clicks Login button |
| **Precondition** | The Blog Website is open  The User has valid account with the System |
| **Basic Path** | 1. The User enters valid credentials 2. The User clicks Login button |
| **Alternative Paths** |  |
| **Post condition** | The User is Logged in the System |
| **Exception Paths** | The User may leave the operation at any time  If the User enters invalid credentials, should not be able to Login |
| **Other** |  |

### 3.2.5 Create Article

|  |  |
| --- | --- |
| **Use Case Name** | Create Article |
| **XRef** | Section 2.5.2, User Create Article |
| **Trigger** | The User clicks Create Article button |
| **Precondition** | The Blog Website is open  The User is Logged in the System |
| **Basic Path** | 1. The User populates Title and Content fields 2. The User clicks Create button |
| **Alternative Paths** |  |
| **Post condition** | A new Article is created in the System with Author – User |
| **Exception Paths** | If the User is not Logged-In, he should not be permitted to Create Article. |
| **Other** |  |

### 3.2.6 Logoff

|  |  |
| --- | --- |
| **Use Case Name** | User Logoff |
| **XRef** | Section 2.5.2, User Logoff |
| **Trigger** | The User clicks the Logoff button |
| **Precondition** | The Blog Website is open  The User is Logged in the System |
| **Basic Path** | 1. The User clicks on Logoff button |
| **Alternative Paths** |  |
| **Post condition** | The User is logged off the System |
| **Exception Paths** |  |
| **Other** |  |

### 3.2.7 Edit Article

|  |  |
| --- | --- |
| **Use Case Name** | Edit Article |
| **XRef** | Section 2.5.3, Author Edit Article |
| **Trigger** | The User clicks Edit button of an Article |
| **Precondition** | The Blog Website is open  The User is Logged in the System  At least one Article exists in Blog  The Article is open |
| **Basic Path** | 1. The User clicks Edit button in the Article   If the User is the Author of this Article, then:   1. The Author is redirected to Edit Article page 2. The Author alters the fields’ content the desired way 3. The Author clicks Edit button |
| **Alternative Paths** | None |
| **Post condition** | The Article content changes according the edits and is re-saved to the System |
| **Exception Paths** | If the User is not the Author of the Article, User should not be able to Edit the Article  If the User is not Logged-In , User should be redirected to Login page upon clicking Edit button |
| **Other** |  |

### 3.2.8 Delete Article

|  |  |
| --- | --- |
| **Use Case Name** | Delete Article |
| **XRef** | Section 2.5.3, Author Delete Article |
| **Trigger** | The User clicks Edit button of an Article |
| **Precondition** | The Blog Website is open  The User is Logged in the System  At least one Article exists in Blog  The Article is open |
| **Basic Path** | 1. The User clicks Delete button in the Article   If the User is the Author of this Article, then:   1. The Author is redirected to Delete Article page 2. The Author clicks Delete button again to confirm deletion |
| **Alternative Paths** | None |
| **Post condition** | The Article is deleted from the System |
| **Exception Paths** | If the User is not the Author of the Article, User should not be able to Delete the Article  If the User is not Logged-In, User should be redirected to Login page upon clicking Delete button |
| **Other** |  |

### 3.2.9 View Users

|  |  |
| --- | --- |
| **Use Case Name** | View Users |
| **XRef** | Section 2.5.4, Admins View Users; |
| **Trigger** |  |
| **Precondition** | The Blog Website is open  Admins is logged-in the SoftUni Blog Website |
| **Basic Path** | 1. The Admins clicks the Admins link  2. Then the "Manage Users" page opens in the same browser window.  3. List of all SoftUni Blog users is displayed with their ID and Email.  4. Links to Edit and Delete actions are enabled, related to each user. |
| **Alternative Paths** |  |
| **Post condition** | the "Manage Users" page containing list of all users opens in the same browser window |
| **Exception Paths** | The Admins may leave the article/page at any time |
| **Other** |  |

### 3.2.10 Edit User

|  |  |
| --- | --- |
| **Use Case Name** | Edit User |
| **XRef** | Section 2.5.4, Admins Edit User |
| **Trigger** |  |
| **Precondition** | 1. Given that the Admins is logged-in the SoftUni Blog Website  2. And the "Manage Users" page is open.  3. List of all SoftUni Blog users is displayed with their ID, Email.  4. "Edit" and "Delete" links for each user account are enabled. |
| **Basic Path** | 1. Admins clicks the "Edit" link  2. "Edit" page opens in the same browser window.  3. The Admins updates the user's data and/or the User status (from user to admin and vice versa).  4. And the Amins clicks Edit link |
| **Alternative Paths** |  |
| **Post condition** | Then the User account changes according to the edits and is re-saved to the Blog System |
| **Exception Paths** |  |
| **Other** |  |

### 3.2.11 Delete User

|  |  |
| --- | --- |
| **Use Case Name** | Delete User |
| **XRef** | Section 2.5.4, Admins Delete User |
| **Trigger** |  |
| **Precondition** | 1. Given that the Admins is logged-in the SoftUni Blog Website  2. And the "Manage Users" page is open.  3. List of all SoftUni Blog users is displayed with their ID, Email.  4. "Edit" and "Delete" links for each user account are enabled. |
| **Basic Path** | 1. Admins clicks the "Delete" link  2. "Delete" page opens in the same browser window.  3. The Admins clicks on Delete link  4. And the "Delete User" page opens  5. Admins clicks on Delete button |
| **Alternative Paths** |  |
| **Post condition** | Then the User account is deleted  Admins is redirected to "Users" page. |
| **Exception Paths** |  |
| **Other** |  |

## 

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal Article Manager database is as follows:

**User Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of User |  |
| Email Address | Text | Internet address |  |
| Article | Pointer | Article entity | May be several |

**Article Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Article |  |
| Author | Pointer | Author entity | Name of Author |
| Contents | Text | Body of article |  |

The Logical Structure of the data to be stored in the Online Journal database on the server is as follows:

**Published Article Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Article |  |
| Author | Text | Name of the Author |  |
| Content | Text | Body of article |  |

### 3.3.2. Security

The server on which the Online Journal resides will have its own security to prevent unauthorized *write*/*delete* access. There is no restriction on *read* access. The use of email by an Author is on the client systems and thus is external to the system.

### 3.3.3. Availability

The system is 100% available depends on the Client Type (Visitor, User, Author or Admin) and its used 24 hours a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

### 3.3.4. Performance

The system shall respond to the member in not less than two seconds from the time of the request submittal.

### 3.3.5. Reliability

The system has to be reliable by testing each operation in the software at least once, load testing should be performed and eventually after every change regression testing should be conducted.

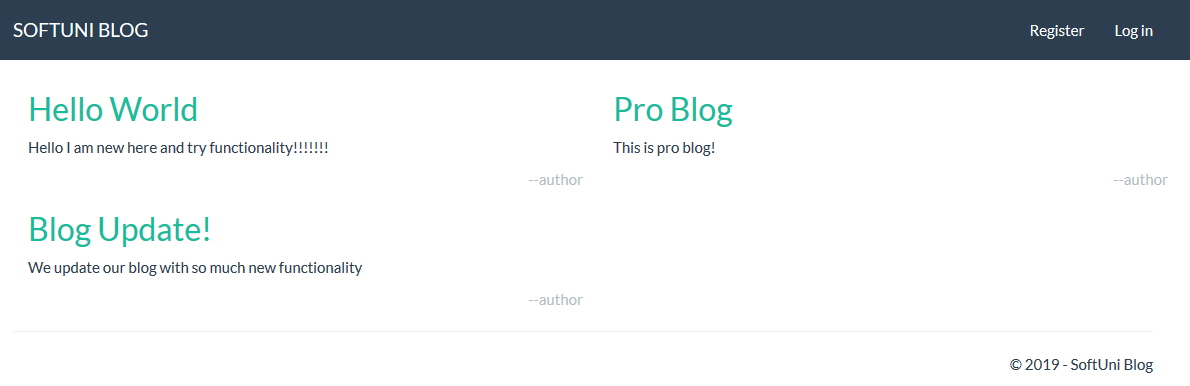
### 3.3.6. Usability

The system has to be user friendly in order for the User to learn and operate with it without any problems.

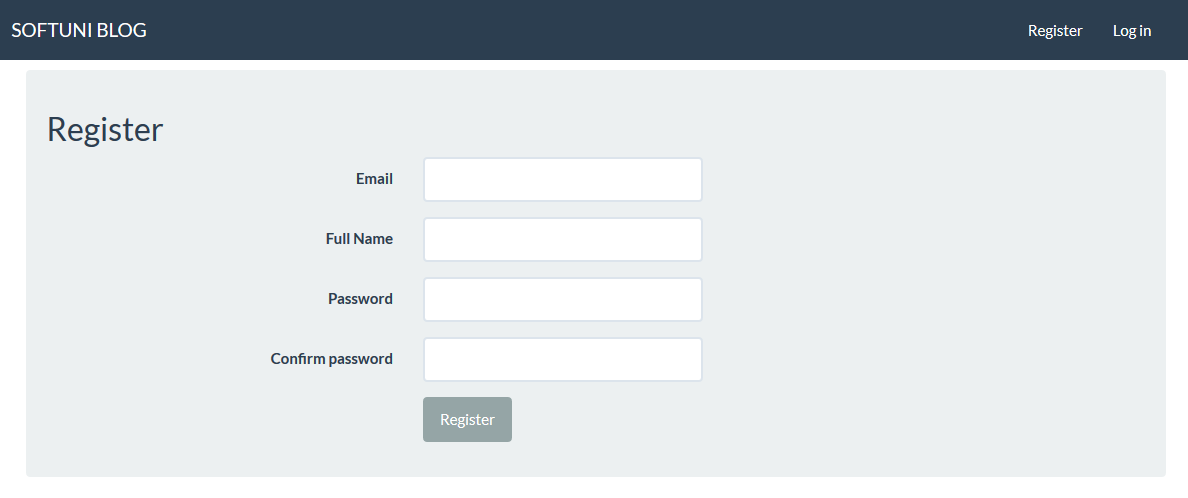
# 4. Other Requirements

## Appendix A SoftUni Blog system views of pages

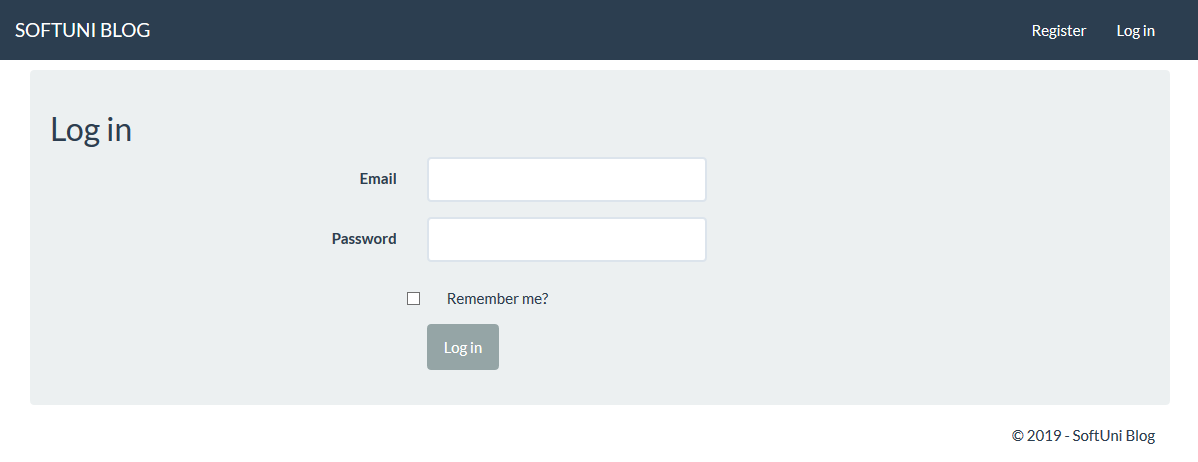
## 4.1 Home Page



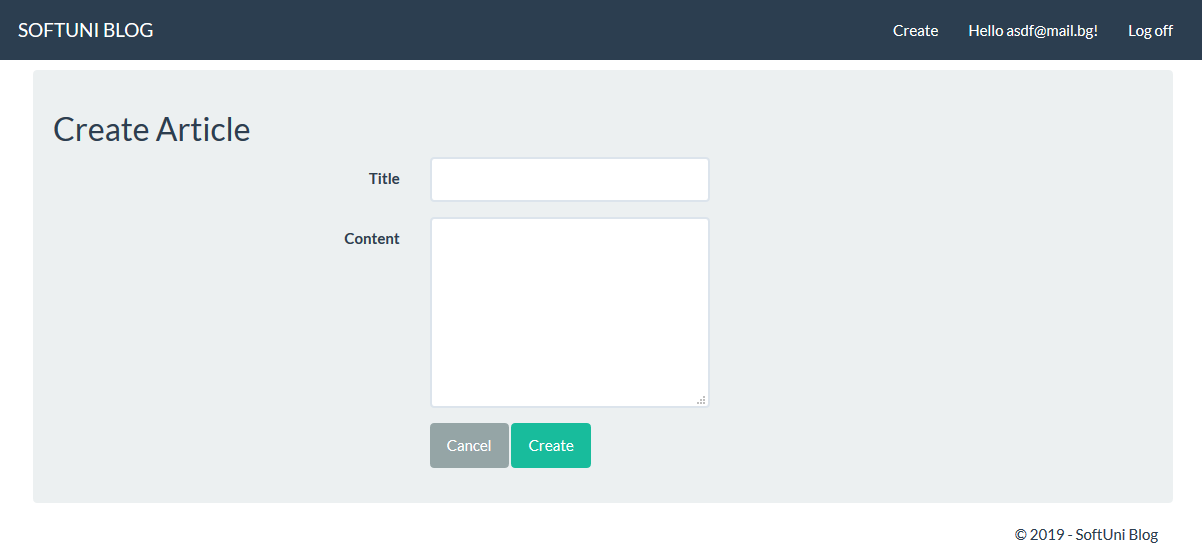
## 4.2 Register Page



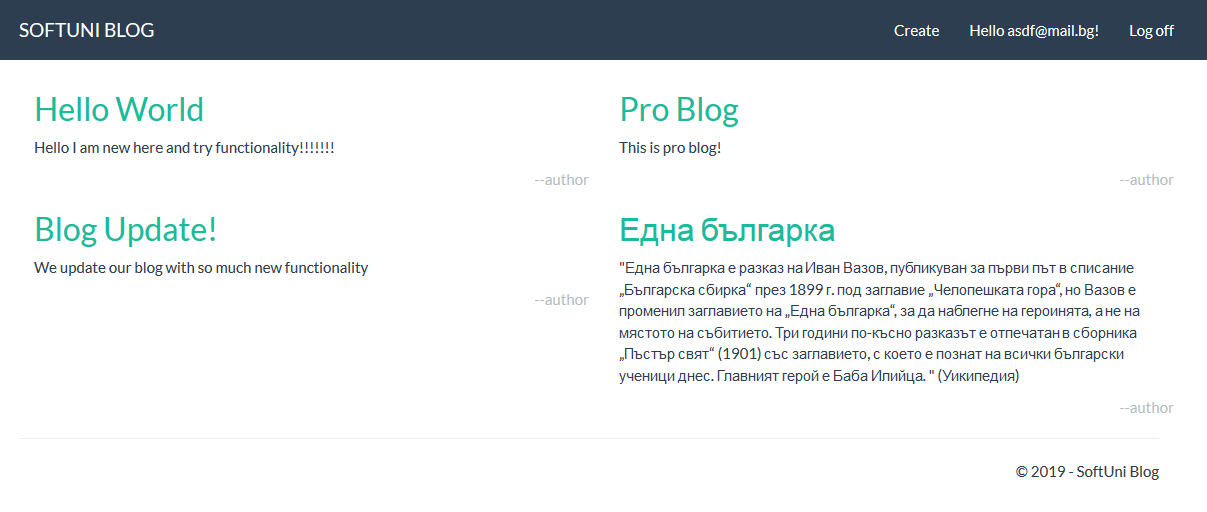
## 4.3 Log in Page



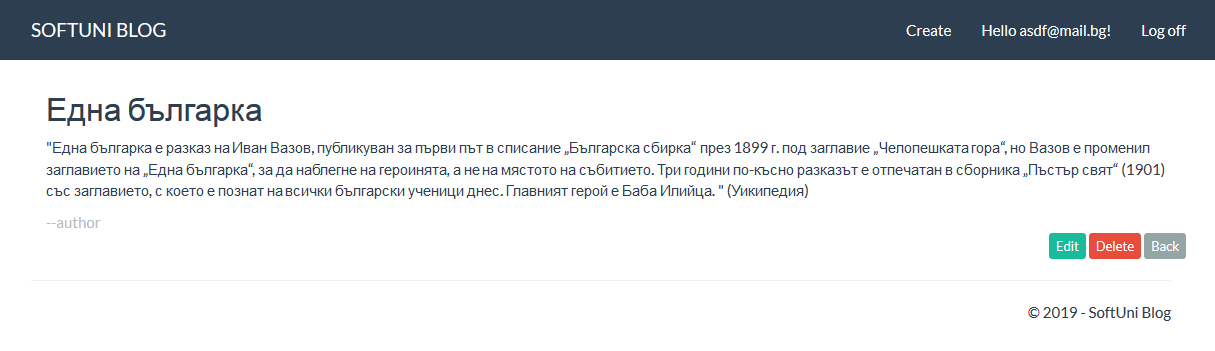
## 4.4 Create Article Page



## 4.5 Read Article Page

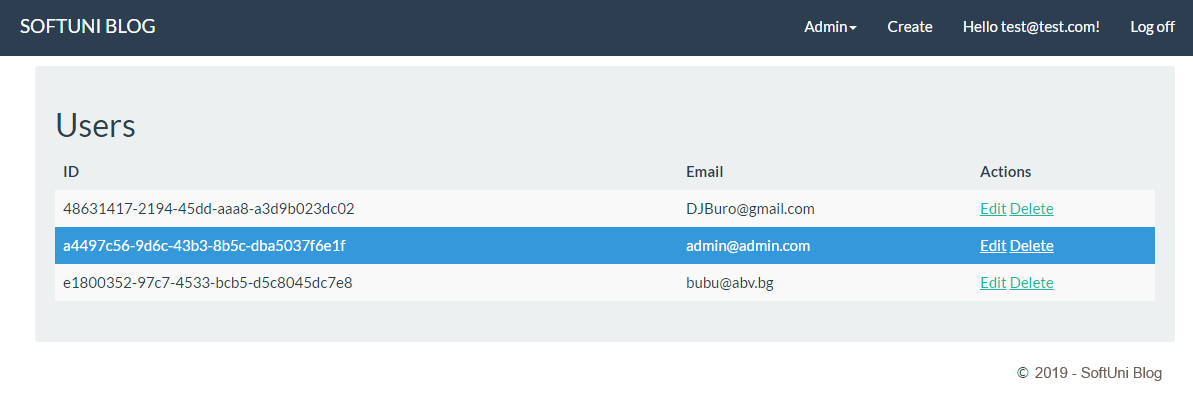


## 4.5 Edit Article Page



## 4.7 Admin pages

### 4.7.1 Users Page



### 4.7.2 Edit user Page

