Software Requirements Specification

SOFTUNI BLOG

Team FAFNIR :

**Gergana Guncheva**

**Gergana Bozova**

**Georgi Lukov**

(Project Title)

(Team Name and Number)

(Team Members)

**Software Requirements Specification**

**Document**

**Version: (n)** **Date: (mm/dd/yyyy)**

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| [1. Introduction ...............................................................................................................................................](#page3) | | | [1](#page3) |
| [1.1](#page3) | [Purpose ................................................................................................................................................](#page3) | | [1](#page3) |
| [1.2](#page3) | [Scope ...................................................................................................................................................](#page3) | | [1](#page3) |
| [1.3](#page3) | [Definitions, acronyms, and abbreviations ...........................................................................................](#page3) | | [1](#page3) |
| [1.4](#page4) | [References ...........................................................................................................................................](#page4) | | [2](#page4) |
| [2. Overall description ....................................................................................................................................](#page6) | | | [4](#page6) |
| [2.1](#page6) | [Product perspective .............................................................................................................................](#page6) | | [4](#page6) |
| [2.2](#page6) | [Product functions ................................................................................................................................](#page6) | | [4](#page6) |
| [2.3](#page7) | [User characteristics .............................................................................................................................](#page7) | | [5](#page7) |
| [2.4](#page7) | [Constraints ..........................................................................................................................................](#page7) | | [5](#page7) |
| [2.5](#page7) | [Assumptions and dependencies ..........................................................................................................](#page7) | | [5](#page7) |
| [2.6](#page8) | [Apportioning of requirements .............................................................................................................](#page8) | | [6](#page8) |
| [3. Specific requirements ................................................................................................................................](#page9) | | | [7](#page9) |
| [3.1.1](#page9) | | [User interfaces .............................................................................................................................](#page9) | [7](#page9) |
| [3.1.2](#page10) | | [Hardware interfaces .....................................................................................................................](#page10) | [8](#page10) |
| [3.1.3](#page10) | | [Software interfaces .......................................................................................................................](#page10) | [8](#page10) |
| [3.1.4](#page11) | | [Communications interfaces ..........................................................................................................](#page11) | [9](#page11) |
| [3.2](#page11) | [Functional requirements ......................................................................................................................](#page11) | | [9](#page11) |
| [3.2.1](#page11) | | [User Class 1 - The User ...............................................................................................................](#page11) | [9](#page11) |
| [3.2.2](#page16) | | [User Class 2 - Restaurant Owner ...............................................................................................](#page16) | [14](#page16) |
| [3.2.3](#page20) | | [User Class 3 - Administrator ......................................................................................................](#page20) | [18](#page20) |
| [3.3](#page23) | [Performance requirements ................................................................................................................](#page23) | | [21](#page23) |
| [3.4](#page25) | [Design constraints .............................................................................................................................](#page25) | | [23](#page25) |
| [3.5](#page25) | [Software system attributes ................................................................................................................](#page25) | | [23](#page25) |
| [4. Prioritization and Release Plan ...............................................................................................................](#page29) | | | [27](#page29) |
| [4.1](#page29) | [Choice of prioritization method ........................................................................................................](#page29) | | [27](#page29) |
| [Appendix I: Selection for Cost-Value Approach ........................................................................................](#page31) | | | [29](#page31) |
| [Appendix II: Prioritization Result of 10 selected Requirements Using Cost-Value Approach ..................](#page34) | | | [32](#page34) |
| [Appendix III: Five-Way Priority Scheme ...................................................................................................](#page38) | | | [36](#page38) |
| [Appendix IV: Release Plan .........................................................................................................................](#page49) | | | [47](#page49) |
| [Appendix V: I-star ......................................................................................................................................](#page57) | | | [55](#page57) |

# Introduction

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “SoftUni Blog” system It will explain the purpose and features of the system, the interfaces of the system and what the system will do. This document is intended for both the developers of the system and the quality assurance team.

## Scope

The system is a basic online blog, which provides functionality, where the users can post their own thoughts, their visions on exact problems or things that are in their interests. The system also lets the users to edit or delete their posts.

The users have different permissions as ordinary users and as administrator. The registered users can make posts and edit or delete their posts and the administrator monitors for the correctness of the posts in the blog.

The system works on different browsers and needs internet connection.

## 1.3 Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
|  | **Table 1 - Definitions** |
|  |  |
| **Term** | **Definition** |
| Blog | .  A regularly updated web-based system, which is displaying the posts of individuals or small group with same interests. |
| Admin/Administrator | A person who is given specific permission for managing and |
|  | controlling the system |
| User | A person who uses the functionalities, provided by the system and have some limited permissions |







Stakeholder



Any person who has interaction with the system who is not a developer.



DESC



Description



RAT



Rational



DEP



Dependency



TAG



A unique, persistent identifier contained in a PLanguage statement [2]



GIST



A short, simple description of the concept contained in a PLanguage statement [2]



SCALE



The scale of measure used by the requirement contained in a PLanguage statement [2]



METER



The process or device used to establish location on a SCALE contained in a PLanguage statement [2]



MUST



The minimum level required to avoid failure contained in a PLanguage statement [2]



PLAN



The level at which good success can be claimed contained in a PLanguage statement [2]



WISH



A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement [2]



DEFINED



The official definition of a term contained in a PLanguage statement [2]



## 1.4.References

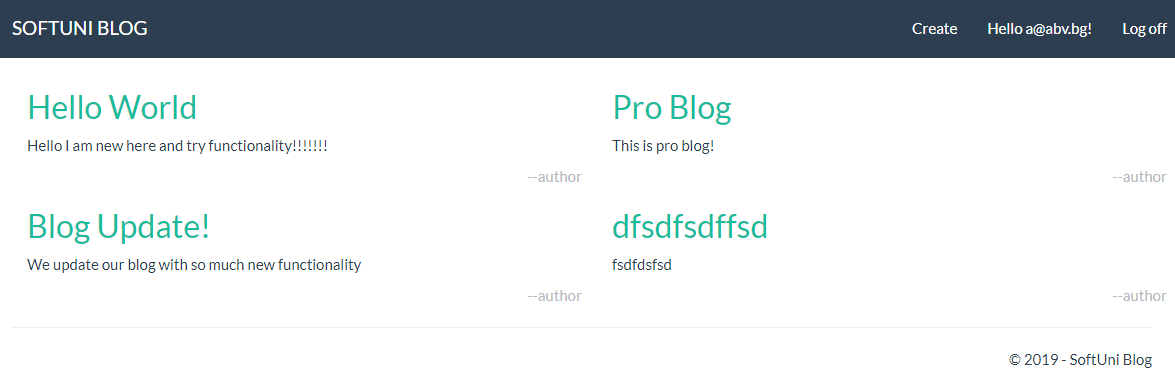
<http://www.softwaretestingstandard.org/> - International standards for software testing

### 2. System Interfaces

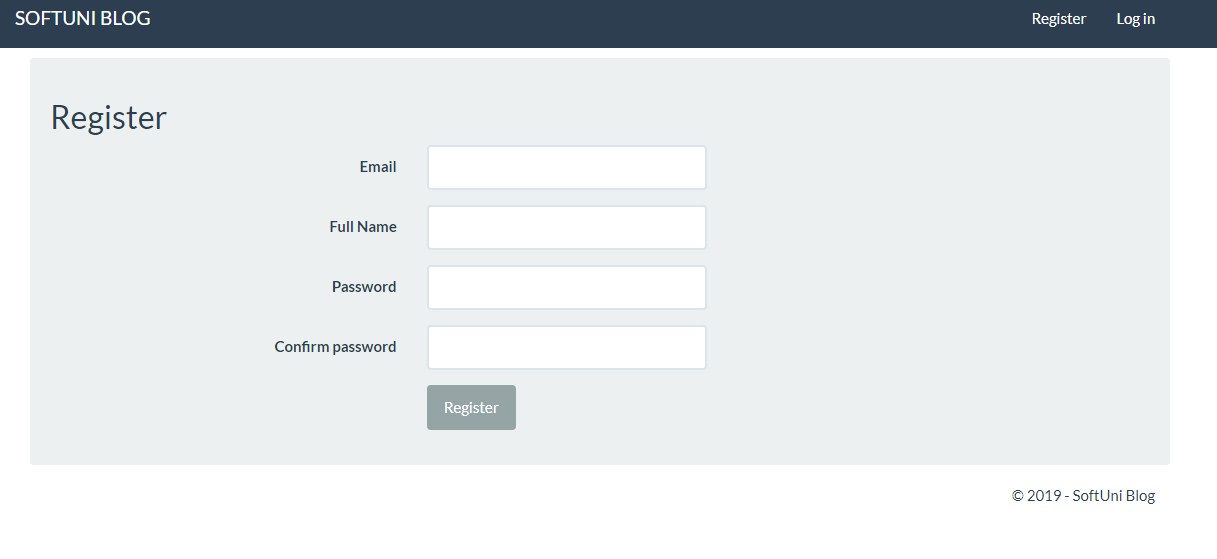
### 2.1 Interfaces

## 2.1.1 User interfaces

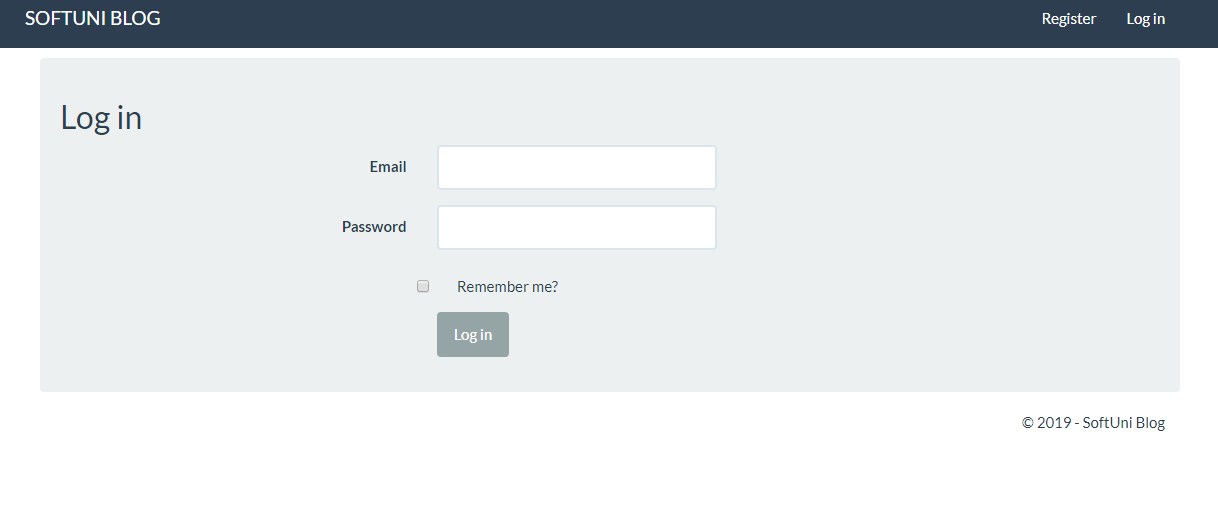
When an user loads the internet address of the system, he/she sees the home page of the web site:



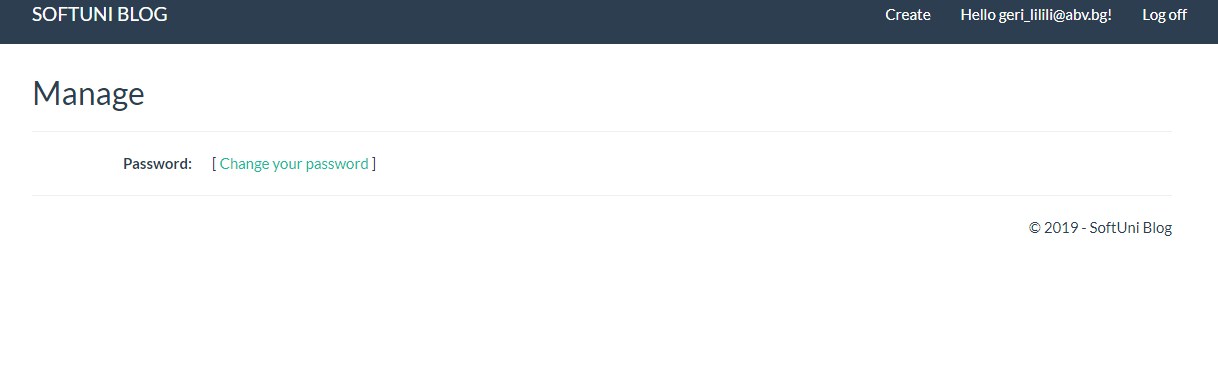
If the user is not registered, he must register his own profile in the system from the Register menu:



If the user is already registered, he must enter his username and password form the Log in menu:



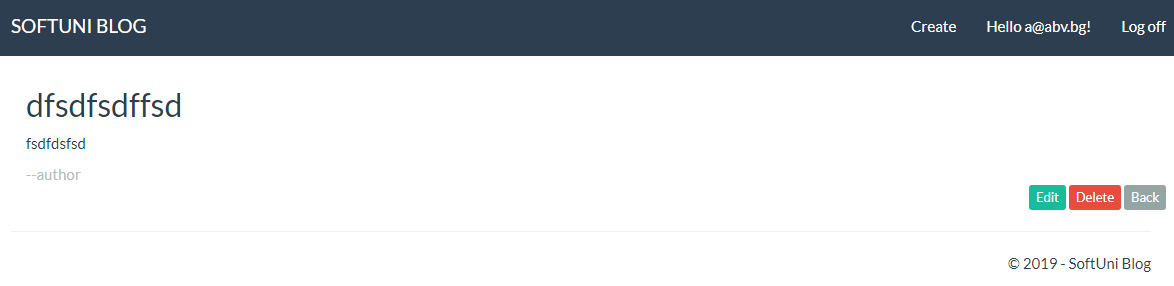
Registered user can manage his profile from special menu, with functionality for changing the password:

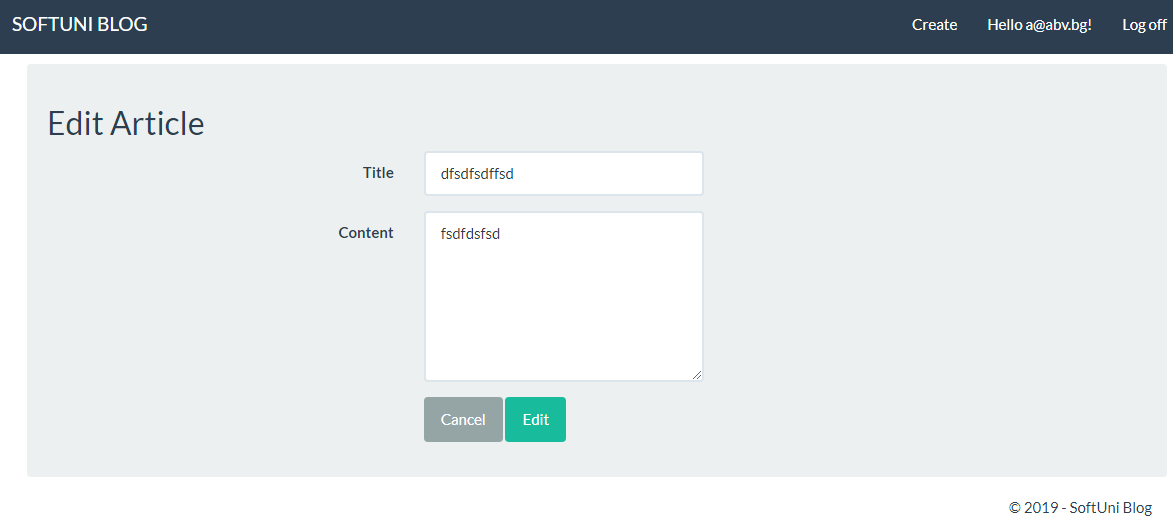


Registered and logged user can create his own article and publish it to the main page of the blog. He must provide Title and Content for his post.

### Screenshot_4

The user can edit or delete his post by clicking on it from the main page:





## 3 Functional requirements

### 3.1 Use Cases

#### 3.1.1 Use Case 1

**ID: UC1**

TITLE: User registration

DESC: When the user loads the internet address of the system he should be able to register his own profile from the menu for registration. The user must provide e-mail, full name, password and password confirmation.

Guest user:

1. Loads the internet address of the system;
2. Goes to the Registration menu;
3. Enters e-mail, full name, password, password confirmation;
4. Submits the form

System:

1. Verifies the information for e-mail. If the e-mail format is not standard the system provides error message;
2. If the password and password confirmation are not the same the system provides error message;
3. Verifies general information and displays an error message if all mandatory fields are not filled;
4. Sends registered details to the database and gives acknowledgement

RAT: In order for a user to register to the system.

DEP: Guest user

#### 3.1.2 Use Case 2

**ID: UC2**

TITLE: User Log in

DESC: When the user is registered, he/she should be able to log in to the system from the Log in menu. The user must be redirected to the main page of the system.

Registered user:

1. Loads the internet address of the system;
2. Goes to the Log in menu;
3. Enters valid e-mail and password;
4. Submits the Log in form

System:

1. Verifies login information and displays an error message if user name and/or password are incorrect;
2. Redirects the user to the main page.

RAT: In order for a user to register on the blog.

DEP: Registered user

#### 3.1.3 Use Case 3

**ID: UC3**

TITLE: Retrieve password

DESC: When registered user try to log in, but don`t remember his password, he must have the option to retrieve his password by providing the valid e-mail address.

Registered user:

1. Loads the internet address of the system;
2. Goes to the Log in menu;
3. Goes to the Forgot password menu;
4. Provides valid e-mail address
5. Submits the information

System:

1. Redirects the user to the Forgot password menu;
2. Provides e-mail address submission option;
3. Submits the e-mail id and provides the reset password link through mail.

RAT: In order for a user to retrieve his/her password.

DEP: The user must be registered and not logged in

#### 3.1.4 Use Case 4

**ID: UC4**

TITLE: Profile page

DESC: When registered user wants to change the information for his profile, he can do this from the profile menu.

Logged user:

1. Goes to the profile menu;
2. Changes the personal information

System:

1. Redirects the user to the profile page
2. Actualize the database with new added personal information

RAT: In order for a user to change his/her profile information.

DEP: The user must be logged in

#### 3.1.5 Use Case 5

**ID: UC5**

TITLE: Profile page

DESC: When registered user wants to change the password, he can do this from the profile menu.

Logged user:

1. Loads the internet address of the system;
2. Goes to the Log in menu;
3. Provides correct username and password;
4. Submits the information;
5. Goes to the profile menu;
6. Add the old password;
7. Add new password
8. Confirm new password
9. Save changes

System:

1. Verifies the username and password;
2. Redirects the user to the main page;
3. Redirects the user to the profile page;
4. Verifies that the new password and the password confirmation are equal;
5. Actualize the database with new password;

RAT: In order for a user to change his/her password.

DEP: The user must be logged in

#### 3.1.6 Use Case 1.6

**ID: UC6**

TITLE: Create Post

DESC: When the user is logged, from the main page he can choose Create menu, where he can create his post to the system.

Logged user:

1. Goes to the Create menu;
2. Add the title for the post;
3. Add the content of the post;
4. Click Create to publish the post;

System:

1. Redirects the user to the creation of post page;
2. Actualize the database with added title;
3. Actualize the database with the added content;
4. Visualize the post on the main page.

RAT: In order for a user to create post

DEP: The user must be logged in

#### 3.1.7 Use Case 1.7

**ID: UC7**

TITLE: Edit/Delete Post

DESC: When the user click on the text of his/her published post, the system redirects him/her to page, where the user can edit or delete his/her post.

Logged user:

1. Click on the text of the post;
2. Changes the title of the post;
3. Changes the content of the post;
4. Click on the button Edit/Click on the button Delete.

System:

1. Redirects the user to the creation of post page;

2. Actualize the database with new added title;

3. Actualize the database with the new added content;

4. Visualize the post on the main page.

RAT: In order for a user to edit or delete his/her post

DEP: The user must be logged in

#### 3.1.8 Use Case 1.8

**ID: UC8**

TITLE: Review blog posts

DESC: The user can review the post in the blog made by another user, but he/she must not have the permissions to edit or delete reviewed post.

Logged user:

1. Click on the text of the post;
2. Click on the button Edit/Delete.

System:

1. Redirects the user to the creation of post page;
2. The system returns error message with information,that the user don`t have permissions for this action.

RAT: In order for a user to review the others posts.

DEP: The user must be logged in

#### 3.1.9 Use Case 1.9

**ID: UC9**

TITLE: Log out

DESC: The user can log out from the system by clicking the Log off button.

Logged user:

1. Click on the Logoff button;

System:

1. Terminates the session of the user;
2. Redirects the user to the main page.

RAT: In order for a user to log out from the system

DEP: The user must be logged in

**4 Non functional requirements**

**4.1 Usability Requirements**   
  
•    Colors what we use in this web system design must be attractive.  
•    Easy navigations are preferable to do any task.  
•    Multiple flows (ways) are preferable to do any task.  
•    Home page should be centralized system (Screen/Window) to go to any feature and to get any result.  
•    The facility to return to Home page from any page should available.  
  
**4.2    Performance Requirements**  
  
•    It has to load, with in the industry standard time.  
•    It has to support up to 1 000 concurrent users.  
•    Response time should be 4 seconds

**4.3    Compatibility Requirements**

•    It should support all types of hardware versions, operating systems and browsers  
  
**4.4    Operating Systems**   
  
•    Operating systems -Wndows98 and upper Versions (EX: Win98, windows 2000 prof, XP Vista and win NT Server,windows 2000 server, 2003 server and windows 2008 server)  
   
**4.5    Browsers**  
  
•    IE 6.0 and higher versions , Latest versions of Mozilla Firefox, Opera and Google Chrome  
  
4.6    Security Requirements  
  
•    Data should be encrypted during Login and while performing publishing.  
•    Server on which database resides must have its own security to prevent unauthorized access. It should be accessed only to the privileged users  
•    Server on which application resides must have its own security. Authorization and Authentications checks should be made.