
Software Requirements Specification for Yourstruly Blog

Version 1.0.0

Prepared by:

Arceño, Hannah Bella C.

January 10, 2022

Note: This SRS document is mainly based on the current progress of the project which is still incomplete. Thus, this document is also a partial SRS for the project.

TABLE OF CONTENTS

1. INTRODUCTION	4
1.1 Purpose.....	4
1.2 Intended Audience	4
1.3 Intended Use.....	4
1.4 Scope.....	4
1.5 Definitions and Acronyms.....	4
1.6 Overview	5
2. OVERALL DESCRIPTION.....	6
2.1 User Needs.....	6
2.1.1 User characteristics.....	6
2.1.2 Product functions.....	6
2.2 Product perspective.....	6
2.2.1 System interfaces.....	6
2.2.2 User Interfaces	6
2.2.3 Hardware interfaces	7
2.2.4 Software interfaces.....	7
2.3 Assumptions and Dependencies.....	7
3. SPECIFIC REQUIREMENTS.....	8
3.1 Functional requirements	8
3.1.1 Visitor	8
3.1.2 Member.....	8
3.1.3 Reader	8
3.1.4 Author.....	9
3.1.5 Admin.....	9
3.2 External Interface Requirements	10
3.2.1 User interfaces.....	10
3.2.2 Hardware interfaces.....	10
3.2.3 Software interfaces.....	10
3.2.3 Communications Interfaces.....	11
3.3 System Features	11
3.3.1 Member Registration.....	11

3.3.2 Login/Logout System	11
3.3.3 Automatic Logout.....	12
3.3.4 Blog View.....	12
3.3.5 Blog Search.....	12
3.4 Nonfunctional requirements	12
3.4.1 Performance.....	12
3.4.2 Safety.....	12
3.4.3 Security	13
3.4.4 Usability	13
3.4.5. Quality.....	13

Revision History

Version	Date	Name(s)	Reason	Sections changed
1.0.0	Jan. 10, 2022	Hannah Bella C. Arceño	---	---

1. INTRODUCTION

1.1 Purpose

Yourstruly Blog aims to be a medium for people---particularly those interested in programming---to post blogs and reach their audience. It also aims to be an avenue for people who likes blogs of similar interest to find and read posts they like.

1.2 Intended Audience

This document is intended to be read by designers, developers, testers, and project managers as well as other stakeholders involved in developing the site.

1.3 Intended Use

This document specifies requirements for a simple blog page of a website and is mainly intended to be used as basis in creating its significant functionalities and features as well as other necessary features for requirements management of a software and system products.

1.4 Scope

Primarily, Yourstruly Blog shall allow authorized users or registered authors to share blog posts.

Moreover, the blog site shall:

- Let visitors to register as a member
- Allow members to login to or logout from the site
- Allow readers to search for blogs according to a topic or by a search term
- Allow members to send message/feedback to site owner

The system stores user information and blog contents in a database.

The system is temporarily accessible via a local machine that has an installation of the necessary frontend and backend dependencies/technologies (indicated in further sections) and has Internet connection.

1.5 Definitions and Acronyms

Users

- | | |
|-----------|--|
| ▪ Visitor | A visitor is anyone who views or visits the site. |
| ▪ Member | A member is a person who successfully registered or joined the site via the Registration page. |

- Reader A reader is a member who can read a blog in the site.
- Author An author is a member who has the privilege of posting a blog in the site. For the meantime, only admins can be authors.
- Admin “Admin” is a short term for Administrator. An administrator is a person one who manages the blog site alongside its members.

Technical terms

- HTML HTML stands for HyperText Markup Language.
- CSS CSS stands for Cascading Style Sheets.
- PHP PHP is a recursive acronym for PHP: Hypertext Preprocessor.
- DBMS DBMS is an acronym for Database Management System.
- OS OS stands for Operating System.

Other terms

- I/P I/P is an abbreviation for “input”.
- O/P O/P is an abbreviation for “output”.
- NA NA is an abbreviation for “Not applicable” (as of the moment).

1.6 Overview

The following sections shall describe in detail the functionalities, features, constraints, and dependencies of the product.

The SRS is organized as follows:

- Section 2 provides overall idea of the product.
- Section 3 details the specific requirements of the product, functional and nonfunctional.

2. OVERALL DESCRIPTION

2.1 User Needs

2.1.1 User characteristics

Any user may opt to join as a member of the blog. However, the target users of the product are people with interest in programming or computer technology who would like to share their programming experiences and other related information. Readers who are also interested in programming topics are target users of the product.

Moreover, the target users are users belonging to younger age groups such as high school students and college students, who enjoy reading about the trends and updates on Technology.

The target users may have beginner-level or intermediate knowledge in working with web-based applications.

2.1.2 Product functions

The blog lets its visitor to register as a member of the site. Members may then login and use basic privileges like reading and searching blogs.

A member can be registered as an author, and they will have the privilege of posting blogs. If the user is only a member, they do not have this additional privilege.

Furthermore, members are provided a way to contact the administrator of the blog to give a message or feedback via a contact form.

2.2 Product perspective

2.2.1 System interfaces

The blog will run in latest version of Internet browsers like Chrome, Firefox, and Microsoft Edge on a Windows OS as long as it has established connection with a PHP web server and the database.

2.2.2 User Interfaces

The blog groups its content via boxes or containers and provides menus and buttons for users to interact with using their keyboard and/or mouse.

Furthermore, the blog uses external sources available online for the icons and typefaces.

Note: In the meantime, the system will display contents as expected in browsers viewed in standard desktop sizes (i.e. full screen, viewed 100%).

2.2.3 Hardware interfaces

NA

2.2.4 Software interfaces

The frontend of the blog relies on latest technologies of the following languages: HTML, CSS, and JavaScript.

For the backend mechanism, JavaScript and PHP are used. Moreover, a connection to a locally installed Apache distribution software called XAMPP is required. The software connects the blog to a PHP web server, and to the accurate MySQL database.

Below are versions of the languages and DBMS in the early developments of the blog:

- HTML5
- CSS 2.1
- JavaScript ES2015
- PHP 8.0
- MySQL 8.0

2.3 Assumptions and Dependencies

The blog is assumed to run in any Internet browser in a Windows machine which has a file copy of the scripts and MySQL database, a connection to a PHP server, and the Internet.

If Internet connection is not sustained, the interface of the site may not look like what is expected, especially with regards to the typefaces and icons.

3. SPECIFIC REQUIREMENTS

3.1 Functional requirements¹

3.1.1 Visitor

Func. Req. ID #	Functional name	Functional Requirement Description
FR 1.1	Member registration to blog site	The blog system must let visitors to register as a member.

3.1.2 Member

Func. Req. ID #	Functional name	Functional Requirement Description
FR 2.1	Member login	The blog system shall allow its members to login to the system once login requirements are met.
FR 2.2	Member logout	The blog system shall allow its members to logout immediately when prompted.
FR 2.3	Logout due to inactivity	The system shall logout any user who is inactive for 5 minutes.
FR 2.4	Contact admin	Members should be able to send message/feedback to admin.

3.1.3 Reader

Func. Req. ID #	Functional name	Functional Requirement Description
FR 3.1	Blog search via search term	Readers shall be able to search for blogs of specific interest
FR 3.2	Blog search via topic chosen	Readers shall be able to see blogs about certain topic.

¹ Note: Most functionalities written in this section are the functionalities/features already made for the product. But some undeveloped functionalities have been added to give a reader of this document a broader idea of the product.

FR 3.3	Comment/reply under a blog	Readers shall be able to comment or start discussion under a blog and reply to other comments.
--------	----------------------------	--

3.1.4 Author

Func. Req. ID #	Functional name	Functional Requirement Description
FR 4.1	Publish a blog	A registered author should be able to publish their blog.
FR 4.2	Manage blog	An author of the blog should be able to modify or edit their blog after posting. They should also be able to delete their blog(s).
FR 4.3	Attach a topic to their blog	An author should be allowed to attach a topic to their blog(s).
FR 4.4	Blog comments notification	An author should be notified if their published blogs get a comment.
FR 4.5	Reply to comments with label "Author"	An author shall be able to reply to the comments in their blogs and shall have a distinguished label as "Author".

3.1.5 Admin

Func. Req. ID #	Functional name	Functional Requirement Description
FR 5.1	Manage members	An admin of the site shall be able to manage the users. They should be able to see the users, the total numbers, and modify authorization or privileges of members.
FR 5.2	Manage topics	An admin should be able to modify the topics in the site.
FR 5.3	View site traffic	An author should be able to delete a blog they have published.

3.2 External Interface Requirements

3.2.1 User interfaces

The user interacts with the system via a browser. The frontend shall be developed with HTML, CSS, and JavaScript; HTML shall be used primarily for the basic structure, CSS for the layout, style, or overall design and JavaScript shall be used as aid in generating outputs for some of user interactions.

Typefaces or fonts, and icons in the site are sourced externally from Google Font² and Fontawesome³ respectively.

For the backend, JavaScript shall be used for verifying user interactions. To facilitate and verify the contents of the page, and in interacting with the database, PHP shall be used. Moreover, to verify user, PHP shall also be used. Lastly, a MySQL database shall be used to store data of the blog.

3.2.2 Hardware interfaces

NA

3.2.3 Software interfaces

Inputs and outputs related to backend connections with database using XAMPP software and MySQL:

<i>Name of item</i>	<i>Purpose</i>	<i>Source</i>	<i>Destination</i>	<i>Data format</i>
I/P #1: – Register username, email, password (hashed)	Register new member	System - Registration page	Database	Texts (unhashed and hashed)
O/P #1.1: Register successful	Notify user when registration is success	System - Verify	Blog interface (browser)	Graphical text
O/P #1.2: Register unsuccessful	Notify user when registration unsuccessful	System - Verify	Blog interface (browser)	Graphical text
I/P #2: Login username/email, password (hashed)	Login member	System - Login page	Verifying system	Texts (unhashed and hashed)
O/P #2.1: Login successful	Notify user when login is success	System - Verify	Direct to Blog interface	Graphical text

² <https://fonts.google.com/>

³ <https://fontawesome.com/>

O/P #2.2: Login unsuccessful	Notify user when login unsuccessful	System - Verify	Blog interface	Graphical text
I/P #3: Search term	Search blogs related to search term	Blog interface (browser)	Database (query)	Text
I/P #4: Search topic	Search blogs related to topic	Blog interface (browser)	Database (query)	Text
O/P #3.1, 4.1: Related blogs found	Show blogs according to interest of user	System - Verify	Blog interface (browser)	Graphical text
O/P #3.1, 4.1: Related blogs not found	Show blogs according to interest of user	System - Verify	Blog interface (browser)	Graphical text

3.2.3 Communications Interfaces

NA

3.3 System Features

3.3.1 Member Registration

Visitors should be able to register as a member via Registration page.

A visitor should use a valid email and username to register. A valid email and username are those who are not used by other users; therefore, they do not yet exist in the database.

A visitor should also use a valid password to successfully register. A valid password must have the ff: (a) a minimum length of 8 characters, (b) containing at least one capital letter, (c) one number, (d) one symbol.

Once the visitor provided valid details, they should be registered to the database. And they should be able to login into the system via Login page.

3.3.2 Login/Logout System

Already registered members should be able to login to the site via Login page.

A member should be able to login into the site if they provide correct email or username with corresponding matching password.

Once a member successfully login, they should be able to view recent blogs in the home or main page as well as search for blogs through a search term or by choosing a topic from a menu.

If a member wishes to logout of the site, they should be able to do so via Logout button.

3.3.3 Automatic Logout

Logged in members who have been inactive for five (5) minutes should be automatically logged out of the site.

A notification or page that notifies them that they have been logged out due to inactivity must be shown after they are logged out due to inactivity.

3.3.4 Blog View

Any logged in member should be able to view ten (10) recent blogs immediately after they logged in.

The recent blogs should be ordered by their 'Published date' in a descending order.

3.3.5 Blog Search

Any logged in member may search for a blog by specifying a search term in a search box or by choosing a blog topic from a topic menu.

Blogs related to the search term either by: 'Author name' or 'Title' should be shown after the member clicks a search button. On the other hand, blogs related to the chosen topic should be shown once the user clicks the topic in a menu.

The blogs should be listed by their 'Published date' in a descending order.

Should there be no blogs related either to the search term or topic, the user should be notified that no related blogs were found.

3.4 Nonfunctional requirements

3.4.1 Performance

- The blog site should load and be usable within 5-7 seconds.
- The blog site should update the interface upon an interaction within 5 seconds.
- The database should be normalized to prevent redundant data and improve performance.
- Queries to the database should be one at a time to prevent crashes.

3.4.2 Safety

- The database should be backed up once a day and should be kept for a week.

3.4.3 Security

- Only registered members should be able to login to the site.
- Only logged in or verified users should be able to access the home page of the site.
- Only logged in or verified users should be able to interact within the site.

3.4.4 Usability

- User interface should be easy to learn without any assistance.
- User interface should boost confidence of users and allow them to complete noncritical tasks without errors.
- Texts should be readable by default.

3.4.5. Quality

- User interface should reflect a modern, minimalist design.
- Images should be displayed with a resolution of 85-100%.