



***Software Requirement and Design Specifications***

***Social Butterfly***

***Version: 0.1***

|  |  |
| --- | --- |
| *Course Code* |  |
| *Instructor* |  |
| *Project Team* |  |
| *Submission Date* |  |



***[Instructions]***

* *No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.*
* *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.*
* *This’ Instruction’ section should also be removed in final document.*

*Table of Contents*

[1. INTRODUCTION 5](#_Toc17152)

[1.1. Purpose of Document 5](#_Toc17153)

[1.2. Intended Audience 5](#_Toc17154)

[2. OVERALL SYSTEM DESCRIPTION 6](#_Toc17155)

[2.1. Project Background 6](#_Toc17156)

[2.2. Project Scope 6](#_Toc17157)

[2.3. Not In Scope 6](#_Toc17158)

[2.4. Project Objectives 6](#_Toc17159)

[2.5. Stakeholders 6](#_Toc17160)

[2.6. Operating Environment 6](#_Toc17161)

[2.7. System Constraints 6](#_Toc17162)

[2.8. Assumptions & Dependencies 6](#_Toc17163)

[3. EXTERNAL INTERFACE REQUIREMENTS 7](#_Toc17164)

[3.1. Hardware Interfaces 7](#_Toc17165)

[3.2. Software Interfaces 7](#_Toc17166)

[3.3. Communications Interfaces 7](#_Toc17167)

[4. FUNCTIONAL REQUIREMENTS 8](#_Toc17168)

[4.1. FUNCTIONAL HIERARCHY 8](#_Toc17169)

[4.2. Use Cases 8](#_Toc17170)

[4.2.1. [Title of use case] 8](#_Toc17171)

[5. NON-FUNCTIONAL REQUIREMENTS 9](#_Toc17172)

[5.1. Performance Requirements 9](#_Toc17173)

[5.2. Safety Requirements 9](#_Toc17174)

[5.3. Security Requirements 9](#_Toc17175)

[5.4. User Documentation 9](#_Toc17176)

[SDS 10](#_Toc17177)

[6. SYSTEM ARCHITECTURE 11](#_Toc17178)

[6.1. SYSTEM LEVEL ARCHITECTURE 11](#_Toc17179)

[6.2. SOFTWARE ARCHITECTURE 11](#_Toc17180)

[7. DESIGN STRATEGY 12](#_Toc17181)

[8. DETAILED SYSTEM DESIGN 13](#_Toc17182)

[8.1. DATABASE DESIGN 13](#_Toc17183)

[9. APPLICATION DESIGN 15](#_Toc17184)

[10. REFERENCES 15](#_Toc17185)

[11. APPENDICES 17](#_Toc17186)

## 1. Introduction

#### 1.1. Purpose of Document

This document’s purpose is to explain the working of our project, Social butterfly. And to make it easier for the reader to understand the underlying architecture and limitations of it as well.

#### 1.2. Intended Audience

This document is intended for our course instructor and peers.

#### 1.3 Definition of Terms, Acronyms and Abbreviations

Not applicable

#### 1.4 Document Convention

This document will be using Arial font at 10 font size for paragraphs and various font sizes for headings and titles. It will also use **bold** and *italics* for emphasis where required.

## 2. Overall System Description

#### 2.1. Project Background

The project aims to help the developers learn how gigantic social networking websites function. As well as how their development process goes.

#### 2.2. Project Scope

The project aims to incorporate the main basic functionalities required for a social networking website to function. Secure account management and interacting with other users is the core part of the project.

#### 2.3. Not in Scope

The project at this stage does not intend to compete with any existing social networking service. Additionally the project is not intended to be used commercially, therefore it has no monetization features.

#### 2.4. Project Objectives

The project objectives include learning how social networking websites work. How they are developed and maintained.

#### 2.5. Stakeholders

The software will be used by the general public as it is a web based social networking website.

#### 2.6. Operating Environment

The software is web based, meaning it will work on any modern desktop web browser (tested on Chrome 71) regardless of operating system. Hardware wise is required a widescreen to work properly, i.e. smartphones and handheld devices are not supported at this time. The interface required a keyboard and mouse to interact.

#### 2.7. System Constraints

* Modern web browser required (tested on Chrome 71)
* English language interface
* Currently works on localhost, once live it will require internet access to use

#### 2.8. Assumptions & Dependencies

[This section will identify:

* Any assumptions taken regarding the system or environment
* Any dependency of system on any external factor.]

## 3. External Interface Requirements

[This section is intended to specify any requirements that ensure that the new system will connect properly to external components. Place a context diagram showing the external interfaces at a high level of abstraction.]

#### 3.1. Hardware Interfaces

Not applicable

#### 3.2. Software Interfaces

[Describe the connections between this system and other external software components (identified by name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify and describe the purpose of the data items or messages exchanged among the

Software components. Describe the services needed and the nature of the inter-component communications. Identify data that will be shared across software components. ]

#### 3.3. Communications Interfaces

The project is web based and uses php POST and GET to communicate and pass information between various web pages. The whole project runs within a web browser. It uses HTML, PHP, CSS therefore a modern updated browser is recommended. Database is accessed using standard queries.

## 4. Functional Requirements

### 4.1. Functional Hierarchy

[This section will give a big picture of overall system functionality. The main modules/features of system and their sub-functions will be described here in the form of a functional hierarchy so that, before getting into the use case, audience could grab the idea of overall system functions.]

#### 4.2. Use Cases

##### 4.2.1. Add Friend / Follow

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:**  Add Friend / Follow |  |
| **Use Case Description:** The user can browse for and add another user as a friend or follow them | |
| **Primary actor:** User | **Other actors:** Other existing users |
| **Stakeholders:** |  |
| **Relationships**:   * **Includes:** * **Excludes:** | |
| **Pre-conditions:**   * User is already logged in | |
| **Flow of Events:**   1. Actor clicks on browse users 2. Actor selects user he/she wants to add/follow 3. Actor is taken to another users timeline 4. Actor clicks on *add friend* | |
| **Alternative and exceptional flows:**  **4.1** …. | |
| **Post-conditions:** Database is updated | |

##### 4.2.2. Make Post

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:**  Make post |  |
| **Use Case Description:** The user can make new post | |
| **Primary actor:** User | **Other actors:** none |
| **Stakeholders:** |  |
| **Relationships**:   * **Includes:** * **Excludes:** | |
| **Pre-conditions:**   * User is already logged in | |
| **Flow of Events:**   1. Enter post content 2. Click on post | |
| **Alternative and exceptional flows:**   * 1. Click on image button   2. Click on browse   3. Upload image   4. Click on post | |
| **Post-conditions:** Database is updated | |

##### 4.2.3. Edit Profile

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:**  Edit Profile |  |
| **Use Case Description:** The user can edit their profile | |
| **Primary actor:** User | **Other actors:** none |
| **Stakeholders:** |  |
| **Relationships**:   * **Includes:** * **Excludes:** | |
| **Pre-conditions:**   * User is already logged in | |
| **Flow of Events:**   1. Click on Account 2. Click on Edit profile 3. Change values if required 4. Upload new profile picture if required 5. Click on save | |
| **Alternative and exceptional flows:** | |
| **Post-conditions:** Database is updated | |

##### 4.2.4. Admin Access (Panel)

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:**  Admin Access |  |
| **Use Case Description:** Admin can delete users | |
| **Primary actor:** Admin | **Other actors:** users |
| **Stakeholders:** |  |
| **Relationships**:   * **Includes:** * **Excludes:** | |
| **Pre-conditions:**   * Admin is already logged in | |
| **Flow of Events:**   1. Search User 2. Click on Delete User button 3. Post of deleted user will also be deleted 4. Friendlist of deleted user will also be deleted 5. Profile of Deleted User will be deleted | |
| **Alternative and exceptional flows:**   * If Database relations are not cascaded then anomalies will be created | |
| **Post-conditions:** Database is updated | |

##### 4.2.5. Browse Newsfeed

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** Browse Newsfeed |  |
| **Use Case Description:** The user can browse their newsfeed and view other user’s posts and also visit their timelines from here. | |
| **Primary actor:** User | **Other actors:** none |
| **Stakeholders:** |  |
| **Relationships**:   * **Includes:** * **Excludes:** | |
| **Pre-conditions:**   * User is already logged in | |
| **Flow of Events:**   1. Click on newsfeed 2. View posts | |
| **Alternative and exceptional flows:**   * 1. Click on username of another   2. Visit their timeline | |
| **Post-conditions:** none | |

## 5. Non-functional Requirements

#### 5.1. Performance Requirements

The response rate of the website depends on the server it is deployed on. The project is precise and reliable as well. The hyperlinks generated point to the correct address. However the project is currently prone to SQL injection in this version.

#### 5.2. Safety Requirements

Not applicable

#### 5.3. Security Requirements

The information entered by the user and the posts made by the user as well as the profile picture uploaded is publically visible. Users must observe caution while using the website.

#### 5.4. User Documentation

Not applicable

## SDS

# 6. System Architecture

The project is a web based service, as such, it is divided into various web pages. There are php scripts that are included into various pages such as the database connection script and the header.php file. They are located in the root directory and in the scripts sub folder. The main web pages are accessible by visiting the correct url (localhost/social-butterfly/index-register.php or example) and then the others are available using the user interface such as the header or various hyperlinks throughout the website.

### 6.1. System Level Architecture

[The architecture should decompose the system at a top level in a way that provides a foundation for more detailed design work. The architecture discusses relationships and roles of system elements (subsystems, components, modules, etc.), but does not provide internal details. Areas for consideration are:

* System decomposition into elements
* The relationship between the elements
* Interfaces to external systems
* Major physical design issues such as where elements will execute
* Global design strategies such as error handling

-

NOTE: You will use UML diagrams (Deployment and Component diagrams) to document the overall system architecture. ]

### 6.2. Software Architecture

[The software architecture should include how User level Layer will interact with Database layer. Use diagram for showing the interaction between the layers.

* User Interface Layer - Middle Tier
* Data Access Layer.
* Or other

You can give any other architecture also]

# 7. Design Strategy

[Describe the design strategies or decisions that impact the overall organization of the system and its high-level structures. This information should provide the reader with insights into the key abstractions and mechanisms used in the system architecture.

For the strategy, discuss the reasoning employed (possibly referring to previously stated design goals and principles) and any trade-offs. Areas for consideration include:

* Future system extension or enhancement
* System reuse
* User interface paradigms
* Data management (storage, distribution, persistence)
* Concurrency and synchronization]

# 8. Detailed System Design

[A detailed design should include the following:

* Detailed class diagram along with a detailed description of all attributes, functions or methods specifying interactions between different classes/modules.
* Logical data model (E/R model)
* Detailed GUIs]

### 8.1. Database Design

[A detailed Database design should include the following:

* Logical data model (E/R model)
* Data dictionary]

**8.1.1. ER Diagram**

[Entity Relationship Diagram of the system with description]

**8.1.2. Data Dictionary**

[The convention recommended for writing the data dictionary is as follows.]

**8.1.2.1 Data 1**

[Description (Refer to Template on next page). ]

8.1.2.2. Data 2

[Description (Refer to Template on next page). ]

.

.

8.1.2.3 Data n

[Description (Refer to Template on next page). ]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **< Data 1>** | | | | | | | |
| **Name** | | Give primary name of the data or control item, the data store or an external entity. | | | | | |
| **Alias** | | Write other names used for the first entry. | | | | | |
| **Where-used/howused** | | List all processes that use the data or control item and how it is used (e.g., input to process, output from the process, as a store, as n external entity) | | | | | |
| **Content description** | | Notation for representing content. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| [Column1  Name] | [Description of the column] | | [Type of column] | [Length of column] | [Is  Column  Null able] | [Default  Value] | [If Primary  Key than write  PK, if Foreign Key then FK, if not a key leave it blank] |
| [Column2  Name] | [Description of the column] | | [Type of column] | [Length of column] | [Is  Column  Null able] | [Default  Value] | [If Primary  Key than write  PK, if Foreign Key then FK, if not a key leave it blank] |
|  |  | |  |  |  |  |  |

## 9. Application Design

[A detailed application design should include the following:

* Detailed Sequence diagram and Collaboration diagram with parameter list
* State Transition Diagram
* Activity Diagram

**9.1.2. Sequence Diagram**

**9.1.2.1 <Sequence Diagram 1>**

[Diagram & Explanation of diagram]

9.1.2.2 <Sequence Diagram 2>

[Diagram & Explanation of diagram]

.

.

**9.1.2.3 <Sequence Diagram n>**

[Diagram & Explanation of diagram]

**9.1.3. State Diagram**

**9.1.3.1 <State Diagram 1>**

[Diagram & Explanation of diagram]

9.1.3.2 <State Diagram 2>

[Diagram & Explanation of diagram]

.

.

.

**9.1.3.3 <State Diagram n>** [Diagram & Explanation of diagram]

**9.1.4. Activity Diagram**

**9.1.4.1 <Activity Diagram 1>**

[Diagram & Explanation of diagram]

**9.1.4.2 <Activity Diagram 2>**

[Diagram & Explanation of diagram]

## 10. References

[This section should provide a complete list of all documents referenced at specific point in time. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. (This section is like the bibliography in a published book).]

## 11. Appendices

[This section should include supporting detail that would be too distracting to include in the main body of the document.]