

**Software Development Proposal**  
for  
LogBlock — programming social media service

Prepared by Hung K. Ngu

*Viet Nam National University Ho Chi Minh City, University of Science*  
*Introduction to Software Engineering*  
**Group 01**

17. October 2024

# Contents

<b>Revision History</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Purpose . . . . .	2
1.2 Intended Use . . . . .	2
1.3 Development Team Information . . . . .	2
<b>2 Overall Description</b>	<b>3</b>
2.1 Service overview . . . . .	3
2.2 User Classes and Characteristics . . . . .	3
2.3 Operating Environment . . . . .	4
2.4 Constraints . . . . .	4
2.5 Abstract Functional Requirements . . . . .	4
2.5.1 User Profile Management . . . . .	4
2.5.2 Abstract User Cross Interactions . . . . .	4
2.5.3 Content Sharing Block (Post) . . . . .	4
2.5.4 Content Feed, Exploration and Interaction . . . . .	5
2.5.5 Direct Messaging . . . . .	5
2.5.6 Expert Requirements . . . . .	5
2.5.7 Administrator Requirements . . . . .	5
2.5.8 Real Time Interaction . . . . .	5
2.5.9 Maintainability . . . . .	5
<b>3 Development Team Contract</b>	<b>7</b>
3.1 Development Team Roles and Responsibilities . . . . .	7
3.2 Development Communication Plans . . . . .	7
3.2.1 Communication Stack . . . . .	8
3.2.2 Executive Decision and Conflict Resolution Protocols . . . . .	8
3.2.3 Work Schedules and Deadlines . . . . .	8
3.2.4 Code and Documentation Standards . . . . .	9

---

3.2.5	Accountability and Performance . . . . .	9
-------	--	---

# Revision History

Revision	Date	Author(s)	Description
1.0	10.12.2024	Hung K. Ngu	Basic social media services
1.1	10.16.2024	Hung K. Ngu	Programming-centralized design descriptions and development model

# Chapter 1

## Introduction

LogBlock is an online social media Service platform with a programming media sharing central view. The platform provides an easier and user-friendly means for both new and senior developers to share their open contributions without compromising too much on reachability of contents.

### 1.1 Purpose

The LogBlock Software Requirements Proposal is addressed to general team development members and business officers. The specification provides high-level abstracted views on the Service's functional operations as well as initial descriptions on the system's constraints. The specification provides a development contract applied to all personnel of the development team.

### 1.2 Intended Use

The proposal and occasional feature references in this document should be viewed as the descriptions of the software Services the system is expected to provide to users and the constraints under which it must operate.

Although some technical details may be included, this document **does not** enforces the inclusion of the defined implementation guides and should only act as a guidance to the desired outcomes of the system.

Albeit usually up-to-date, consumers of this document should note that due to the ever-changing requirements as well as the production model of the development organization, minors changes are to be introduced to accomodate with the needed alternation associated in each development cycle's retrospective. And thus, it is advise to **revise** the document in known interval

### 1.3 Development Team Information

- Team Leader, Project Manager: Hung K. Ngu (22127134) (nkhung22@clc.fitus.edu.vn)
- Executive Officer: Long T. Tran (22127249) (ttlong221@clc.fitus.edu.vn)
- Back-End Implementation Personnel: An C. Trinh (22127004) (tcan22@clc.fitus.edu.vn)
- Front-End Implementation Personnel: Cuong Ng. N. Tran (22127048) (tnncuong22@clc.fitus.edu.vn)
- Quality Assurance Personnel: Long T. T. Nguyen (22127247) (nttlong22@clc.fitus.edu.vn)

## Chapter 2

# Overall Description

### 2.1 Service overview

The LogBlock Social Media Service (hence forward will be referenced as the Service) is a multi-media platform for **programming-based** content distributions, with a focus on appealing interfaces and ease of access to code block **authoring tools** for interactive guides, code snippets. The Service aims to provide a personal profile space for content storage, while also encouraging **visibility** of knowledge via tailored exploration functionalities based on posts interaction. On a post-level, we provides a community-building scheme, allowing users to comment and react, share the content they are seeing, while also allowing **Experts** to recommend alternative solutions to the original posts.

### 2.2 User Classes and Characteristics

The Service is open to public usage, and partition users based on functional requirements as well as contributions to the platform. The intended user classes include four main classes: *Guests*, *Registered user*, *Experts* and *Administrators*.

**Guest** class provides a mean for public, new-come users to access shared content on the platform without explicitly register an operational account. Albeit the class is contrained to various functions of the Service, including: the ability to *post*, the ability to generate interactions with posts via commenting and reaction and limited view of other registred user profiles.

**Registered** class is considered to be the main targeted consumer of the platform. Registered users are generated through the platform's *authentication system* and should be consistent until the of such Service. Registered user can access almost all services provided by the platform as they are the targeted audience, and in turn, the platform is designed to best accomodate the changing needs and scalability of this user base.

**Expert** class inherents all functionalities of the Registered class and provides a way for the platform administration to demonstrate *verifiable* and *trusted content*, as well as providing functionalities for expert individuals to reach their targeted audience in a controllable manners by means of higher post visibility and **Expert Suggested Solution**

**Administrator** class is the gateway for the social developement team and moderators associated with the users' business organization to access the platform and uses tools that provide administration capabilities in order to enforce the Terms of Service of public usage. The administrator class also provides a mean for public users to flag and directly send reports to the moderating sections, allowing the ability to quickly identify and resolve issues of other users.

## 2.3 Operating Environment

The platform primary operating is **Web Browsers**, offering extensibilities and incorporating OS features for ease of usage of the platform. The choice of platforms reflects the needs reach the maximum number of users while offering backwards compatibilities with user on older software or hardwares.

## 2.4 Constraints

The platform should respect Privacy and Data Protection. All user data must be securely stored, adhering to relevant privacy regulations (GDPR, CCPA).

The platform should not promote hazardous contents and should not provide intervention methods to steer naratives or spread conspiracy.

The system should handle large volumes of users and content uploads with minimized performance degradation. The system should provides a mean for administrator to monitor uptime and be notified of downtime as soon as possible.

Based on the choice of Operating Environment, the design of User Interaction, User Experience should be consistent across different devices and browsers, offering Cross-Platform Compatibility.

## 2.5 Abstract Functional Requirements

### 2.5.1 User Profile Management

**FR-0:** Users can be promoted from Guest to Registered by creating an account through the service dedicated authentication system.

**FR-1:** Each Registered user is associated with a unique Handle. Each Registered user is in possession of a personal profile.

**FR-2:** Registered Users can create, update, and delete their profiles.

**FR-3:** Profile features include a profile picture, biography section (with character limit), and posts display.

**FR-4:** Follower and following counts should be displayed on the profile page.

**FR-5:** Registered Users can pin posts to their profiles.

**FR-6:** Registered Users can privately view their activity log.

### 2.5.2 Abstract User Cross Interactions

**FR-7:** Provides a Follow/Unfollowing button for external Registered users to subscribe to the targeted profile.

**FR-8:** Provides a Message button for external Registered users to directly message the recipient profile privately.

**FR-9:** Provides a Block/Report button for external Registered users to report, isolate unwanted or harmful contents and notify the moderation team.

### 2.5.3 Content Sharing Block (Post)

**FR-10:** The content block should be visually appealing, but not obstructive and should corresponds to the flexible design of the platform.

**FR-11:** The content block should be able to display original user's description, code block, as well as salient attachment through Markdown rendering.

**FR-12:** Registered Users can upload images from device storage or drag-and-drop using the running OS

functionality.

**FR-13:** Registered Users can tag other users as co-contributor.

**FR-14:** The captioning description should be limited to at most 1000 characters.

**FR-15:** External Registered users can like, share and save content blocks.

**FR-16:** External Registered users can comment on the content blocks, creating a sub-thread within the content block

**FR-17:** Sub-thread comments can be liked, commented on, creating another sub-thread depth if it does not already exists.

**FR-18:** Sub-thread comments depth should not exceed 3.

## 2.5.4 Content Feed, Exploration and Interaction

**FR-19:** External Registered Users content can be promoted to connected Registered users through the use of News Feed.

**FR-20:** Registered Users can explore trending posts outside of own's connection through the Explorations page.

**FR-21:** Exploration posts is sorted based on real time interaction rate, tags.

**FR-22:** Registered Users can explore or connect with known associates through the search function, the search function should find corresponding results based on display information and User Handle.

**FR-23:** Search results should be locally stored.

## 2.5.5 Direct Messaging

**FR-24:** Registered Users can send and receive direct messages privately from external Registered users.

**FR-25:** Direct messages are limited to markdown-compatible messages.

**FR-26:** Direct message is limited to 500 characters per message.

## 2.5.6 Expert Requirements

**FR-27:** **FR-27:**

## 2.5.7 Administrator Requirements

**FR-28:** User should not have the ability to access or leverage the functionalities of Administrator class without direct promotion from the moderation team.

**FR-29:** Administrators must have the ability to monitor, flag, and remove content.

**FR-30:** Administrators can issue warnings, suspend or ban accounts, and respond to flagged content.

**FR-31:** Administrators can view Service-specific users and manage ad content.

**FR-32:** Administrators can create help forum responses for users.

## 2.5.8 Real Time Interaction

**FR-33:** Real Time Interaction should provides up-to-date user-subscribed content status through real time means.

**FR-34:** Real Time Interaction should be implemented through email notifications in implementation.

**FR-35:** Real Time Interaction should be rate-limited to 3 updates per minute.

## 2.5.9 Maintainability

**FR-36:** The system should be modular and maintainable, allowing for future expansions of features, including additional editing tools or messaging capabilities.



**FR-37:** The codebase should follow industry best practices and be well-documented to enable easy updates and bug fixes.

## Chapter 3

# Development Team Contract

### 3.1 Development Team Roles and Responsibilities

Each team member should be responsible for the development of functionality descriptions as well as detailed implementation of the whole system due to the small team size. In order to provide a more reliable method of management, the team role is defined as follow:

- Team Leader, Project Manager: Hung K. Ngu (22127134) (nkhung22@clc.fitus.edu.vn)
- Executive Officer: Long T. Tran (22127249) (ttlont221@clc.fitus.edu.vn)
- Implementation Personel: An C. Trinh (22127004) (tcan22@clc.fitus.edu.vn)
- Implementation Personel: Cuong Ng. N. Tran (22127048) (tnncuong22@clc.fitus.edu.vn)
- Implementation Personel: Long T. T. Nguyen (22127247) (nttlont22@clc.fitus.edu.vn)

Project Manager should be responsible for the team coordination, performance and development documentation. Project Manager is expected to apply development schemes and follows standard developement procedures in order to accurately track and maintain production quality. Project Manager should be responsible for reporting and finalizing production results to User.

Executive Officer should be responsible for team specific production. Detailed development models is implemented based on Executive Office plan. Executice Officer is expected to finalize each modular functionalities of the product system specification.

Implementation Personel is responsible for the specific implementation of the codebase as well as testing and ensuring working, performance and competitive production results. Implementation Personel is expected to provide acceptable results in reasonable amount of time.

### 3.2 Development Communication Plans

The Development Team communication plans will be conducted primarily online, with the exception of conflict resolutions and contract changes to be decided in person. Members of developement team should be responsible for their updates on the team's status and quota. An interval of two days is recommended in order to not miss crucial informations and executive decisions.

Team members are expected to be noticed of communication updates and provides reply if necessary in manners of at-most three to five hours since the message departure.

### 3.2.1 Communication Stack

Based on the scope of the project, the communication stack should not be too complicated and based around third party Services, outlined as follow:

- Primary Communication Channel: Discord
- Secondary Communication Channel: Email
- Project and Task Management Service: Jira

### 3.2.2 Executive Decision and Conflict Resolution Protocols

The Development Team will conduct weekly retrospective survey and forward-looking between 19th hour and 22nd hour at weekend in order to evaluate team's production result as well as planning the upcoming development direction; finalizing production materials for submission at each 2-week sprints.

In order to promote innovative ideas and quick iterative development, Scrum model should be used as the engineering process. The team should be able to quickly go through with one's work without fear of failure. After a trial and error process, the finalized ideas that inlined with the project core concept are considered to be accepted into main development branch.

The process of deciding the integration of changes should be held at all weekends, during the retrospective and forward-looking process, in order to reflects changes induced by technical blocks or by other constraints introduced in the development sprint. The process is based on the constrast majority vote system, enabling a quick and and thorough exchanges of ideas through continuous feedback and demonstration. After the demonstration section, final executive decisions are relied on Executice Officer for acceptance.

Conflicts are also part of the process, and undecisive actions of these aspects will cause long-term harm to the project development. Conflicts are to be reported directly to Project Manager as per this contract, and violations of contracts terms are also subjected to incident loggings. The Conflicts Resolution Process determines if it is necessary to escalate the actions needed.

If the Conflicts are affecting less than three of the team's development member, disputes will be handled internally by direct communication and case-specific actions.

Otherwise, if the Conflicts affect the majority of the team's developement member, an in person meeting will be held to account for the related incidents and consideration to contract changes if it is a major blocking in the team's performance.

### 3.2.3 Work Schedules and Deadlines

The Development Team primarily operates online with flexible schedule, allowing other members to cover each's works while not compromising product quality and timing constraints.

The Development Process is conducted every week, spanning from Monday to Sunday. The ending of each week should be a milestone for product reanalysis and performance assertion. Tasks of each Process iteration will be declared and is expected to be completed in a timely manners based on the given periods.

The Central Development Process Work Session will be conducted from the 20th hour to 22nd hour on Sunday, where the team will review production assets in order to apply Quality Controls and maintaining consistency between different codebases.

In the case of missing deadlines, the entire development team is responsible to commit mandatory overtime of combined work in order to resolve missing deadlines. The responsible team member of the original task will be noticed of the incidents; an unreasonable amount of missing deadlines incidents will be reported to Project Manager for further accessment.

### 3.2.4 Code and Documentation Standards

Each individual development team member is responsible for following the Code of Conducting while practicing implementation. The specific coding convention is yet to be decided, but a good proportion will be based on C++ Core Guidelines.

Individual development team member should prioritize readability and maintainability while implementing functionalities, obfuscation of codebase is yet to be necessary and is considered a bad practice in this project.

Based on the scope of the project, the development operations stack should not be too complicated and based around third party Services, outlined as follow:

- Development Operations and Source Control Service: Github
- Project Documents and Assets Storage: Google Drive

Code testings software is yet to be determined. However, development team member should be responsible for quality working production results and self-penetration testing before deliveries.

Before the finalized decision on core implementation technologies, documentation standards is expected to follow IEEE Standard for Software User Documentation. Based on chosen technology, a more flexible and open, autonomous solutions can be used to automate the technical documentation process. Outside of implementation, the documentation is to follow the aforementioned standard.

### 3.2.5 Accountability and Performance

Development team members are expected to be on time with assigned tasks, while also maintaining quality in production results. In rare cases when a task period is misaligned, further set-back can be applied to accommodate the necessary tasks.

Development team members performance will be measured based on their communication participation, their assessment on project responsibilities and their ability to provide exceptional results.

Though not necessary, development team member is also encouraged to propose changes to the development pipeline if see fits; but the final decision still relies on both Executive Officer and Project Manager.

Lack of development participation will results in competency reassessment. The Development Team will try to establish common grounds and provides necessary supports for each individuals during the handling of underperforming members.

However, punishment of course partition degradation will be considered in some cases. If incidents are repeated in an unreasonable amount, or conflicts causing blockage in the development process, notification to course instructor will be considered if applicable.

Lack of team contract adhering will also inherently be processed based on the previously mentioned procedures. Though the Project Manager may consider alternative methods such as contract validation in order to determine if the incident is the result of the incomplete nature of this contract or by the subjective reasoning of individual.