# Software Requirements Specification

**for**

# Pangaea

**Version 1.0 approved**

**Prepared by Tony Mogoa**

**unknown**

**March 5th 2021**

# Table of Contents

**Introduction** 1
- **Purpose** 1
- **Document Conventions** 1
- **Intended Audience and Reading Suggestions** 1
- **Product Scope** 1
- **References** 1

**Overall Description** 1
- **Product Perspective** 1
- **Product Functions** 1
- **User Classes and Characteristics** 2
- **Operating Environment** 2
- **Design and Implementation Constraints** 2
- **User Documentation** 2
- **Assumptions and Dependencies** 2
  - **Assumptions** 2
  - **Dependencies** 3

**External Interface Requirements** 3
- **User Interfaces** 3
- **Hardware Interfaces** 3
- **Software Interfaces** 3
- **Communications Interfaces** 3

**System Features** 3

**Other Nonfunctional Requirements** 3
- **Performance Requirements** 3
- **Safety Requirements** 3
- **Security Requirements** 3
- **Software Quality Attributes** 4
- **Business Rules** 4

**Other Requirements** 4

*Page intentionally left blank.*

# 1. Introduction

## 1.1 Purpose

The purpose of this proposal is the development of a locale-agnostic paywall based blogging platform called Pangaea. Pangaea should allow writers from all over the world to write articles and blog posts and be paid for it regardless of the region they are resident. Readers will all have to pay a subscription fee to read articles.

## 1.2 Document Conventions

No document conventions.

## 1.3 Intended Audience and Reading Suggestions

Read this slowly one sentence at a time.

## 1.4 Product Scope

The scope of Pangaea is modest. What is required is a simple blogging site with a paywall and which pays writers depending on the amount of user reading time.

## 1.5 References

To understand more about the problem being solved read this and this.

# 2. Overall Description

## 2.1 Product Perspective

This project is an improvement on an existing system namely Medium.

## 2.2 Product Functions

- Allow users to create accounts and provide their M-PESA details. Of course a reader provides the number so that they pay to the paywall while the writer provides in order to get paid through it. Notice that a single user can be botha reader and a writer.
- Additionally, for the writer, the system should allow the writer to create a custom subdomain preferably from their username like so: tonymogoa.pangaea.com
- User accounts should have full name, profile pic, preferred article topics, email and password, The existence of user registration requirement presupposes the need for other CRUD functionalities actionable on the said user account details.

- Allow users to write articles which include a title, subtitle, rich-text body which can contain images and or videos, featured image, content tags and curation consent. This requirement presupposes the existence the need for the other CRUD functionalities actionable on each of the said article constituents,
- Allow users to publish articles and save drafts not yet ready for publication.
- Allow users to browse articles on the home page. Articles may be recommended based on topic preference specified by the user. Additionally, The system should mail users daily with top trending articles most relevant to their said taste. The home page should have sections such as “Editor’s Picks”, “Trending”, “Best in <insert topic preference>” and so forth.
- The system should allow curators to promote certain articles in the home page.
- The system should allow users to applaud articles.
- The system should log every read indicating the total read time for each user for each article. Such stats will be used in calculating the payment.
- The system should disburse payment to writers on a monthly basis. For development purposes the payment interval need not be monthly.
- The system should allow users to view both their payment history and billing history.
- The user should be able to report articles with inappropriate content. This requirement presupposes the existence of curators(administrators) to handle the reports.

## 2.3 User Classes and Characteristics

User classes comprise of:

- Readers
- Writers
- Writer readers or the other way round depending on which of either is more dominant
- Curators

## 2.4 Operating Environment

The system will operate on the XAMPP stack that is Apache Server, MySQL DBMS on both Windows 10 and Ubuntu(Tony Mogoa).

## 2.5 Design and Implementation Constraints

- The system should be developed in English.
- The system codebase should use camelCase for scripting and snake case for database columns.
- The system should be using PHP, HTML, CSS frameworks and JS.

## 2.6 User Documentation

No user documentation will be provided. The system should be very intuitive to the user.

## 2.7 Assumptions and Dependencies

### Assumptions

- Tax issues are not considered in the development of this system.

**Dependencies**

- The system depends on Safaricom's M-Pesa API presupposing its dependence on Composer dependency manager.

# 3. External Interface Requirements

## 3.1 User Interfaces

To be worked on.

## 3.2 Hardware Interfaces

The system will be supported on any internet-capable device with a browser. But, depending on the project's schedule feasibility i.e time available, the system may not be device-dimensional responsive.

## 3.3 Software Interfaces

MySQL, Apache

## 3.4 Communications Interfaces

HTTP will be involved.

# 4. System Features

*Please derive the system features when you need to do so for the aforementioned system requirements.*

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

None.

## 5.2 Safety Requirements

None.

## 5.3 Security Requirements

None.

### 5.4 Software Quality Attributes

None.

### 5.5 Business Rules

None.

# 6. Other Requirements

*None.*

# Appendix A: Glossary

*None.*

# Appendix B: Analysis Models

*None.*

# Appendix C: To Be Determined List

*None.*