**Software Requirement Specification**

**Document**

**For**

**Blog Application**

**Prepared by:**

Hasib Ahmad Khaliqi

System Analyst – Intern

30 April 2022

1. **Introduction**
   1. **Purpose**

The blog application is to provide content on the website that answers Developers prospective questions and connect the relevant audience to each other. This document enlists the details and specification of the blog post application.

It is intended for developers, designers and testers working on “Appointment App for Doctors/Patients” as well as project investors.

* 1. **Intended Audience**

The users of this system are software engineers and developers who intend to share their ideas to the community and those who want to learn new technologies and updates related to their field.

* 1. **Overview**

Bloggers can log into this system and post their content so that the followers can see and like the post. Each post requires to be validated by the administrator of the blog and then be accessible to public. Followers of the application are able to react to the posts.

* 1. **Scope**

This document helps the developers, Quality assurance, mangers and clients to understand the development process and plan of the application better. it also estimates the cost and time required for the development of blog.

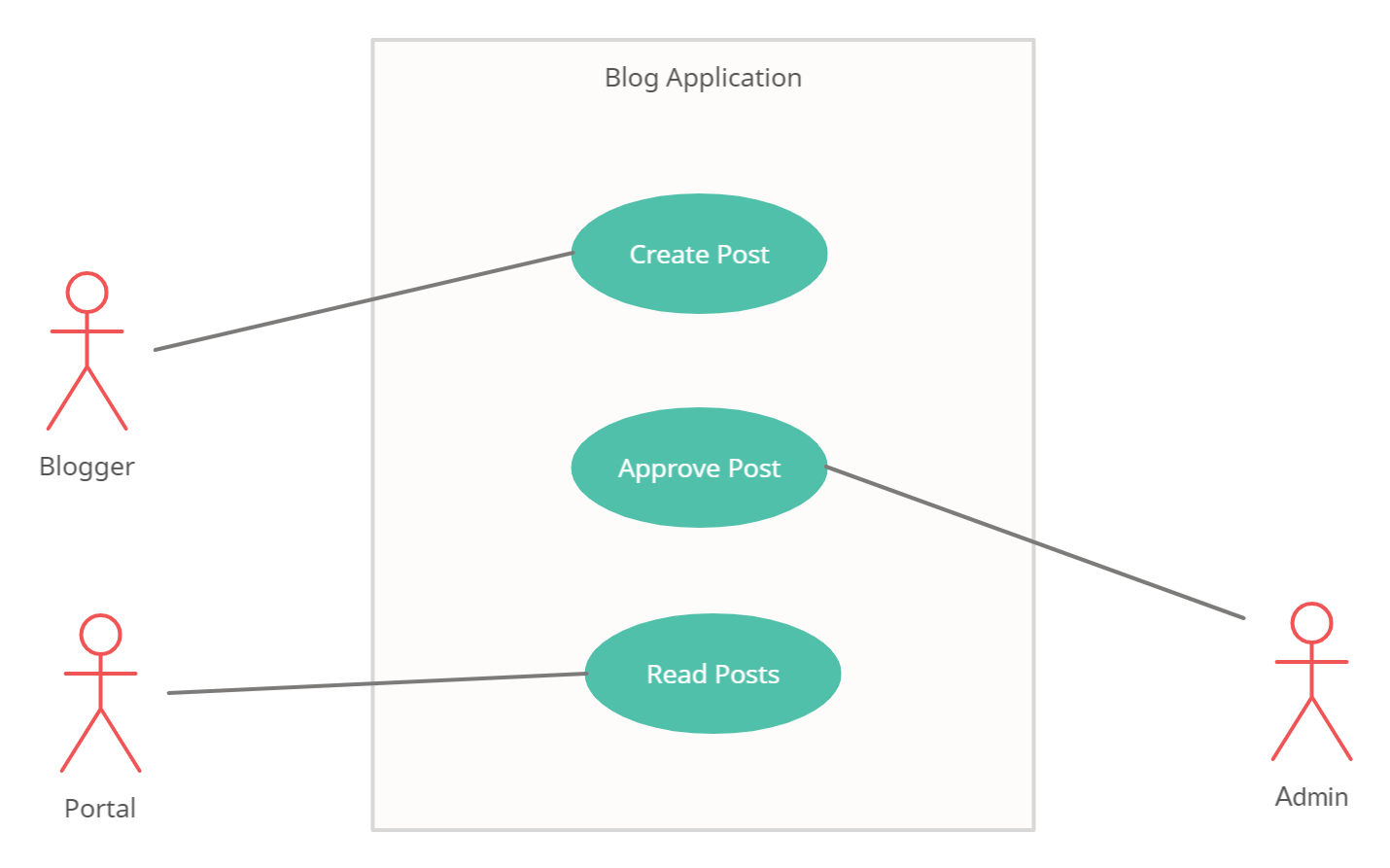
1. **Overall Description**
   1. **User Needs**

This system is used primarily, by the bloggers who post content to the system and share their knowledge. However, people who only wants to learn can also access this system and view its contents. These people are referenced as portal users in this document.

To manage and control the content of the application an administrator is added to the system to organize these tasks.

* 1. **Use cases**

This section outlines the use cases for each of the users separately.



* Blogger use case
  + - Create post
    1. Blogger signs in to the system
    2. Blogger navigates to new post page
    3. Adds post title, description, content and a photo if required
    4. Blogger uploads the post
* Admin use case
  + - Approve post

1. Admin signs in to the system
2. Admin navigates to the approval page where all posts who need approval are stored
3. Admin views each post and confirms or rejects a post by clicking approve post

* Portal use case
  + - Read a post

1. Portal user signs in to the system
2. Opens a post in the system
3. Clicks on the like button to react to the post
4. Can take back the like by clicking again on the like button
   1. **Assumptions and Dependencies**

This system is assumed to be accessed view all mobile devices and web browsers. Also, the application is developed from scratch and there have not been used any previous projects as a base for this. The system should be maintained well so that there is no down time for the end users.

We assume that the bloggers are valid content creators and provide them blogger accounts automatically without verification.

1. **System requirements**
   1. **Functional Requirements**

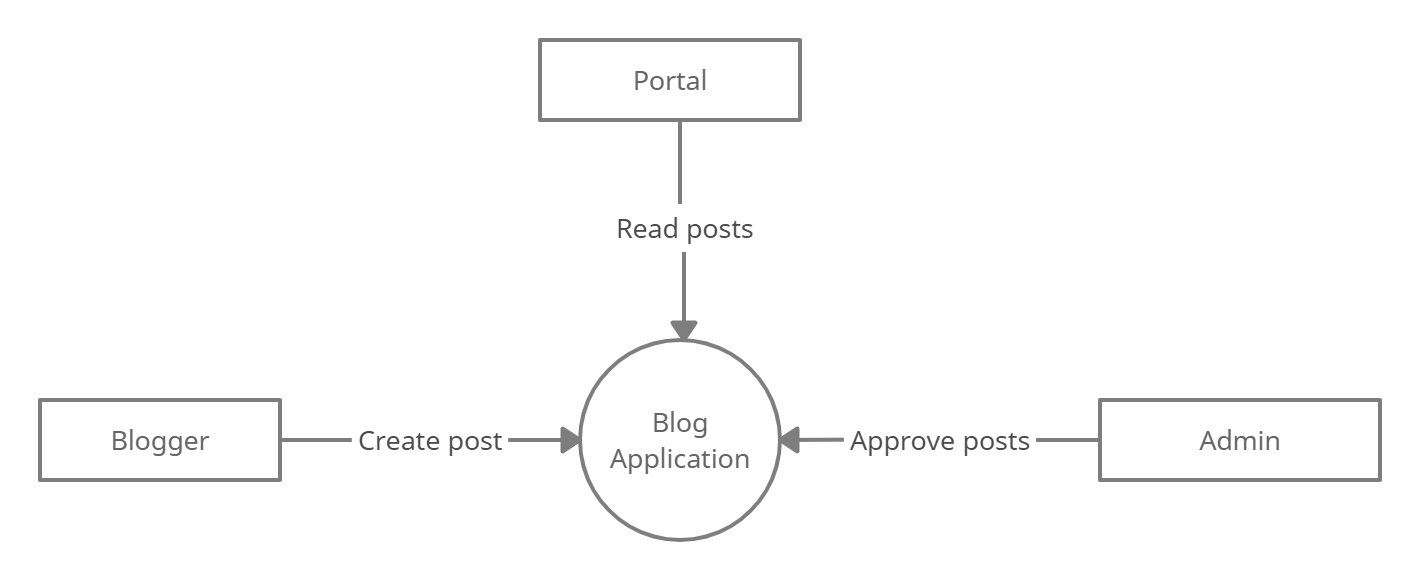
* Bloggers should be able to create accounts on the system without verification need
* Bloggers should be able create, modify and delete their posts on the system
* Bloggers can view posts from other users
* Bloggers can like and dislike all the posts in the system including their own posts
* Bloggers post should be approved by the administrator before being published on the system
* Portal users can log in to the system and manage their accounts
* Portal users can read the posts and like or dislike the posts only
* Admin users can only approve the posts after viewing
* Admin users can delete their own posts but cannot delete other users’ posts
* Posts can have photos but they are optional and not required
* All users can react to posts
* Each post must show total number of likes
  1. **Non-functional Requirements**
* Performance requirements
  + the application should be able to handle large amount of users
  + the application should upload the posts within acceptable amount of time
  + the application should manage the storage to manage posts photos and remove photos once posts are deleted
  + the application should update the interface regularly to show like counts on real-time
* availability: the system must be functional all the time and the downtime should be reduced to the least possible amount within a year if needed.
* Maintainability: the application should be checked continuously for quick bug fixes.
* Usability: the application interface should be easy to learn within the need for tutorials or trainings

**Operating Environment**

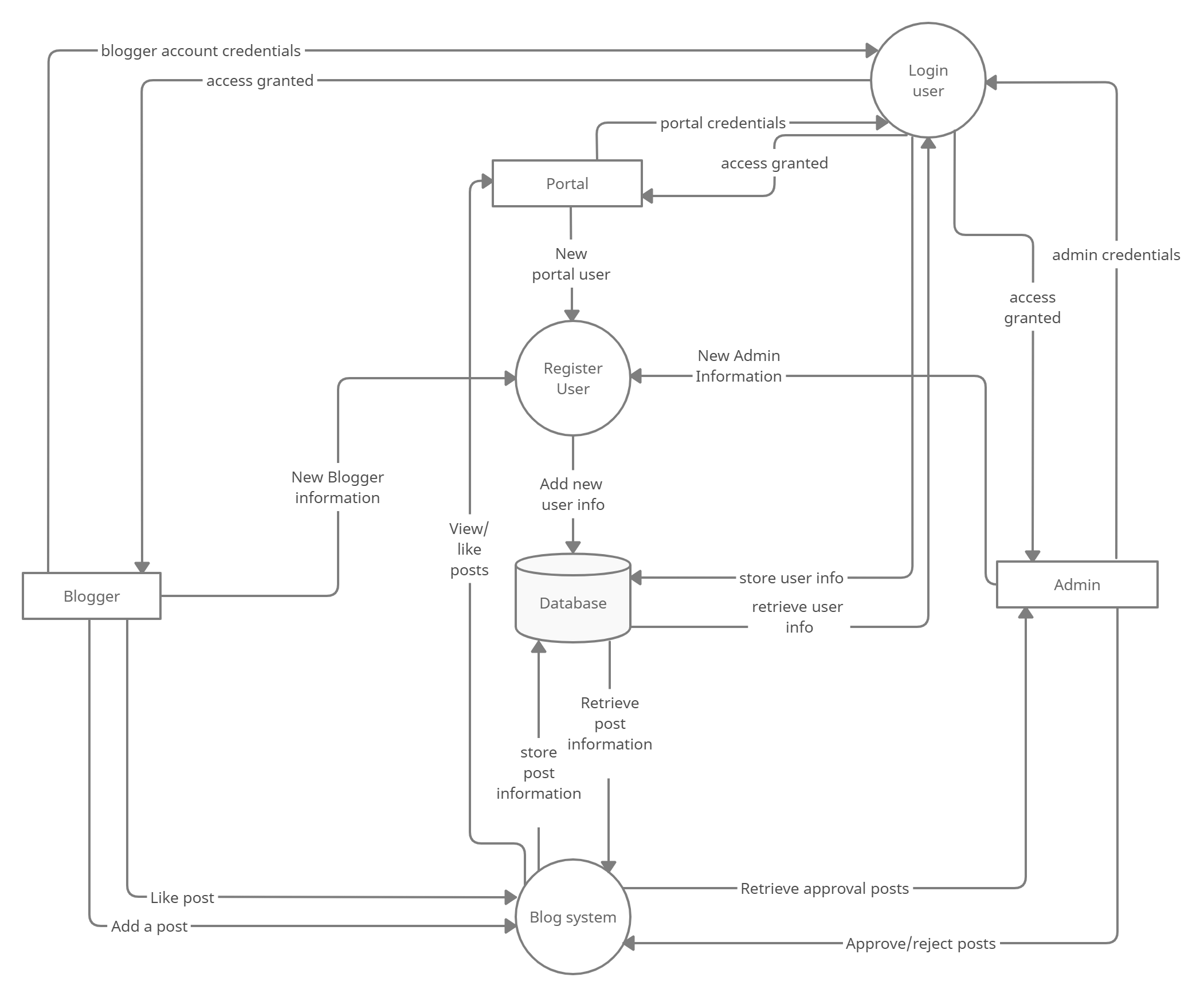
* Operating system: Windows, Linux, Mac OS, Android, iOS
* Platform: python Flask
* Database: SQLite database

1. **Design and Implementation**

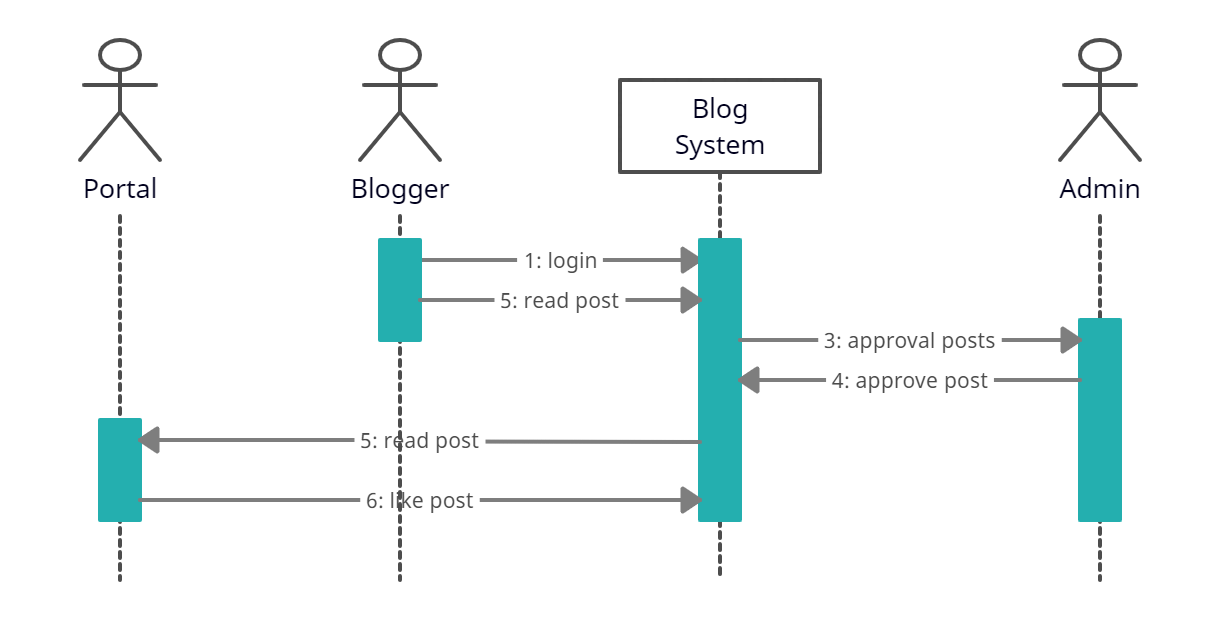
* **Data Flow Diagram level 0: Context Diagram**



* **Data Flow Diagram level 1**



* **Sequence Diagram**



* **Entity-Relationship Diagrams**

