Software Requirements Specification for

<GigPals>

Version 1.0 approved

Prepared by <Abdelrahman Sayed Ahmed>

<Abdelrahman Hesham Mohammed>

<Abdelrahman Ashraf>

<Ahmad Magdy>

<Mohammed Yasser>

<GigPals Team>

<11/12/2022>

Table of Contents

1. Introduction

1.1 Purpose

Gigpals is a freelancing platform that aims to be the perfect community to connect both Channels the clients who need help with there every day challenges whether it is in their personal or career life, and freelancers that have great skills.

1.2 Intended Audience and Reading Suggestions

End users, whether they are clients looking for someone to hire to help them with a gig of any sort, or freelancers looking for jobs to earn extra cash or help grow their portfolio.

Administrators, who are responsible for monitoring and accepting users and gigs into the website and moderating them.

Developing Team, who are responsible for this Product development and maintenance. Overtime to work for each user type.

1.3 Product Scope

System has 3 users' roles:

Admin, client, freelancers

2. Overall Description

2.1 Product Perspective

Freelancer management system is software that makes it significantly easier to manage every aspect of working with freelancers. It automates many processes and combines all the functions of traditional management in one seamless and effective way. By using freelance management systems, companies develop a more dynamic relationship with freelancers along with up scaling their businesses scope.

2.2 Operating Environment

To access the website, users must have access to the XAMPP Control Pane (a management tool that offers to supervise the actions of individual components of XAMPP) so that they can run Apache (Apache is the most widely used webserver software and runs on 67% of all websites in the world) and MySQL (a database management system.)

2.3 Design and Implementation Constraints

We present a web service based approach to enable an evolutionary sensornet system where additional sensor nodes may be added after the initial deployment. The functionality and data provided by the new nodes is exposed in a structured manner, so that multiple applications may access them. The result is a highly interoperable system where multiple applications can share a common evolving sensor substrate. A key challenge in using web services on resource constrained sensor nodes is the energy and bandwidth overhead of the structured data formats used in web services. Our work provides a detailed evaluation of the overheads and presents an implementation on a representative sensor platform with 48k of ROM, 10k of RAM and an 802.15.4 radio. We identify design choices that optimize the web service operation on resource constrained sensor nodes, including support for low latency messaging and sleep modes, quantifying trade-offs between the design generality and resource efficiency. We also prototyped an example application, for home energy management, demonstrating how evolutionary sensor networks can be supported with our approach.

2.4 User Documentation

-how to download MySQL https://www.mysql.com/downloads/

-how to download Apache https://httpd.apache.org/download.cgi

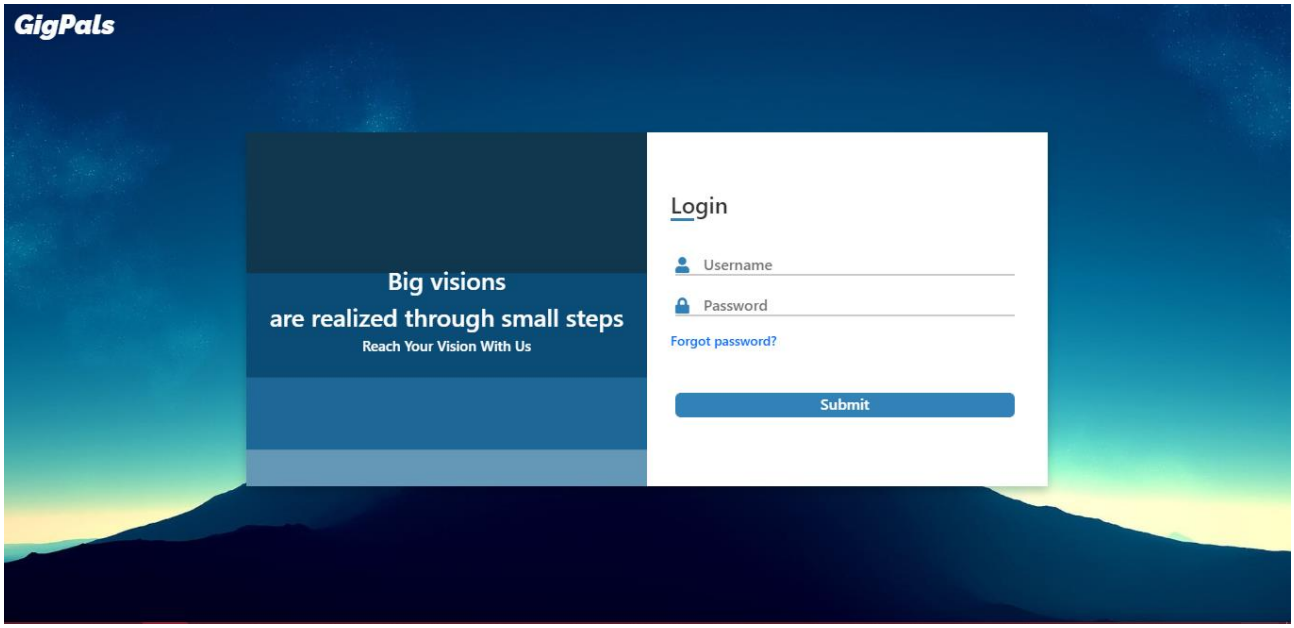
-how to download XAMPP https://www.apachefriends.org/

3. External Interface Requirements

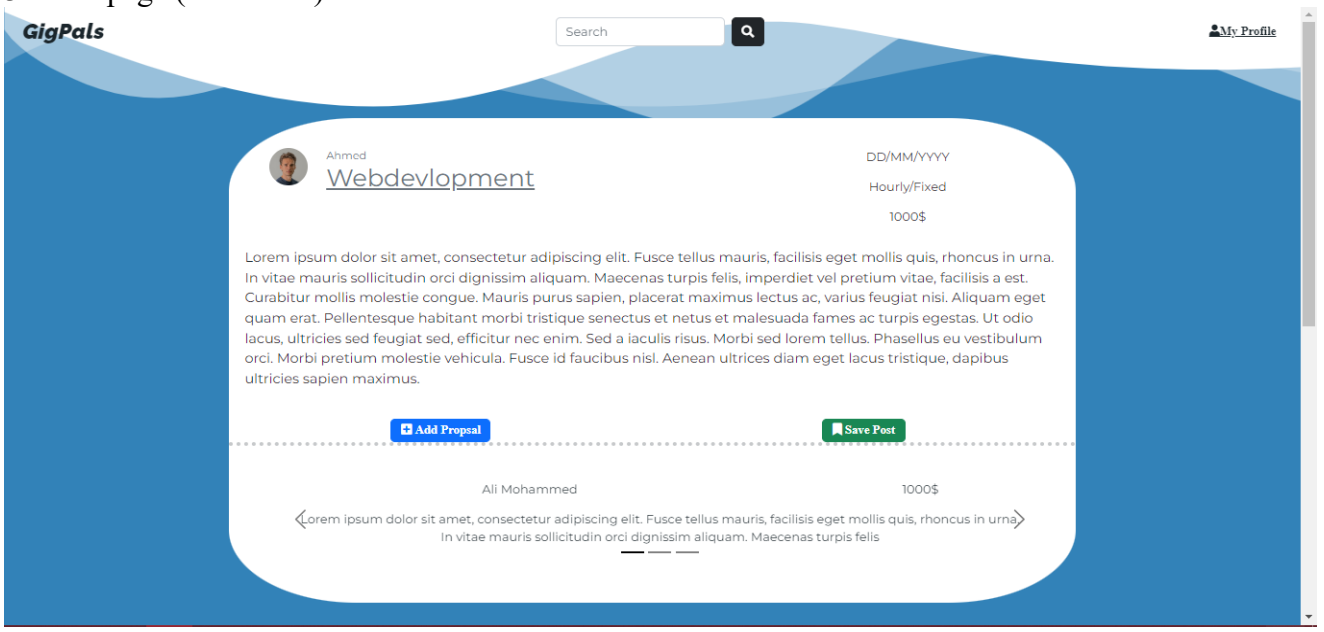
3.1 User Interfaces

1. Landing page

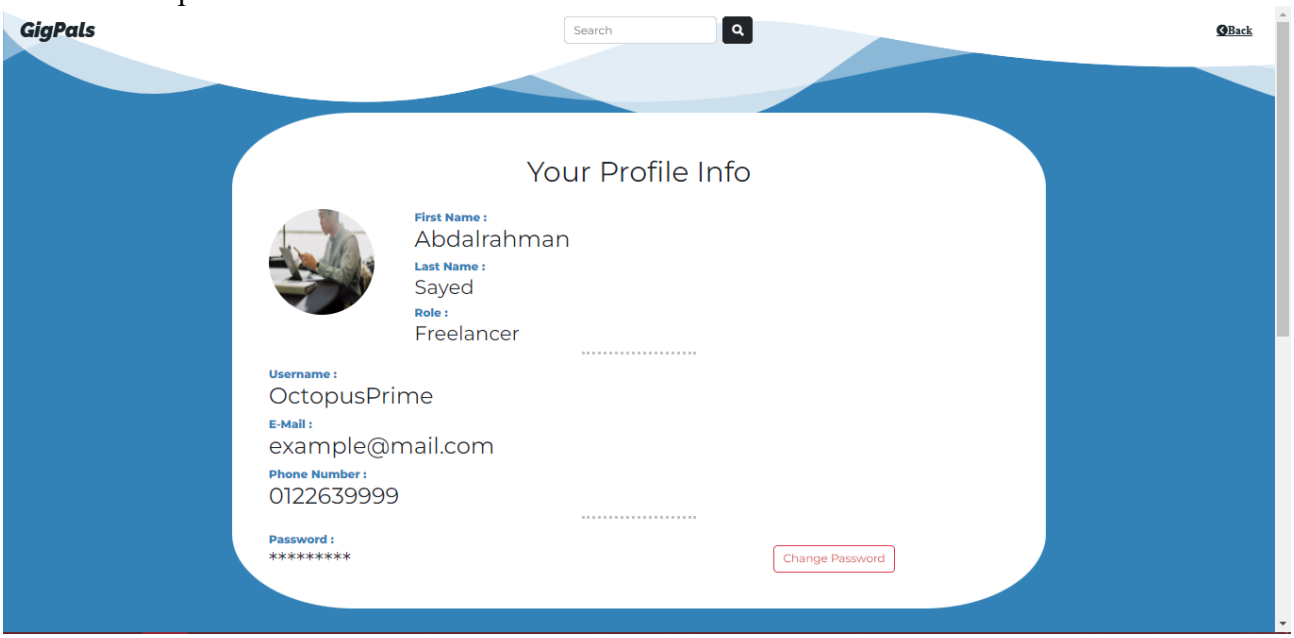
2. Login

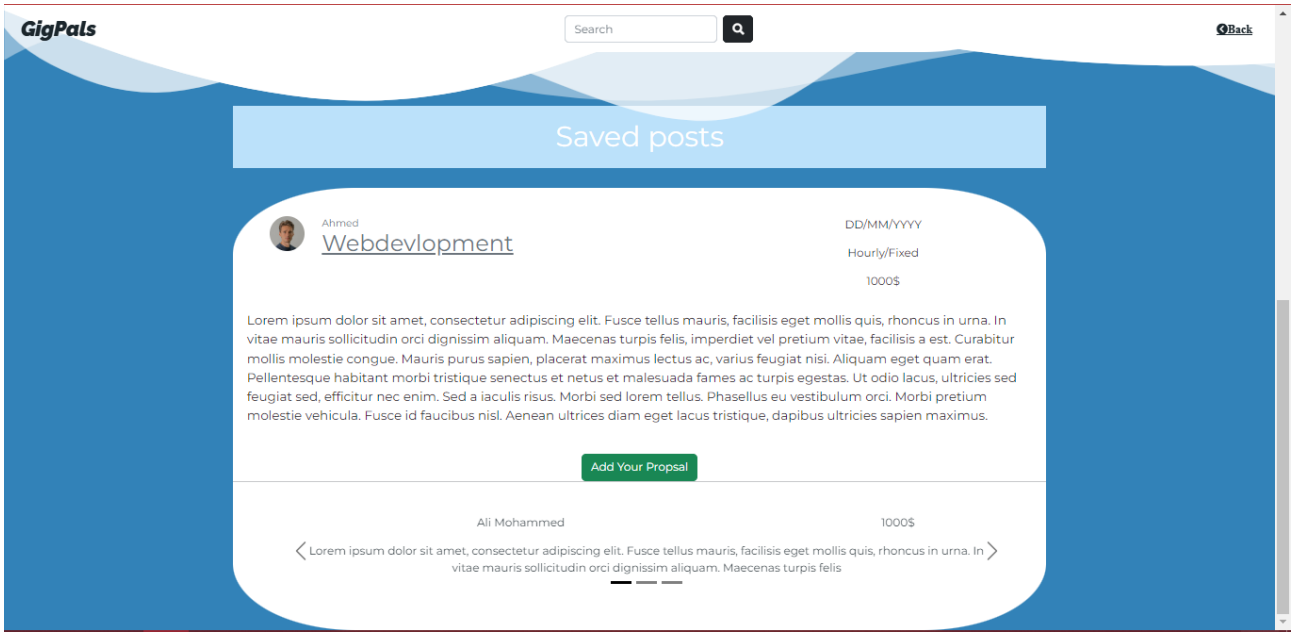


3. Wallpage (freelancer)

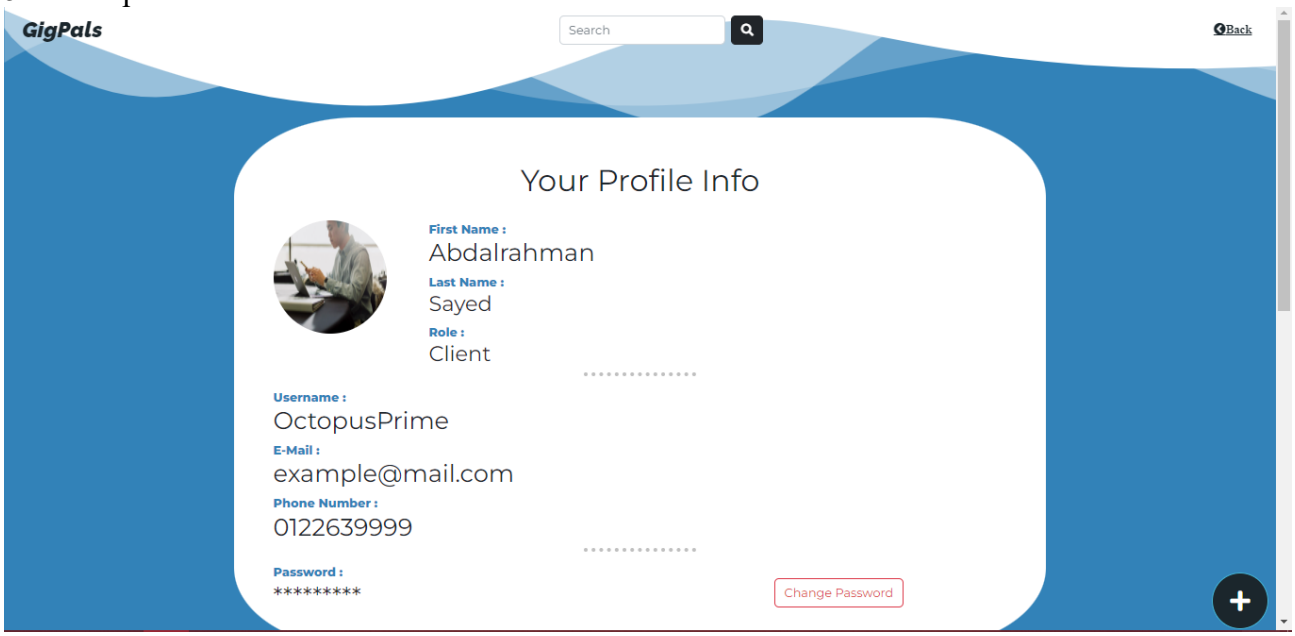


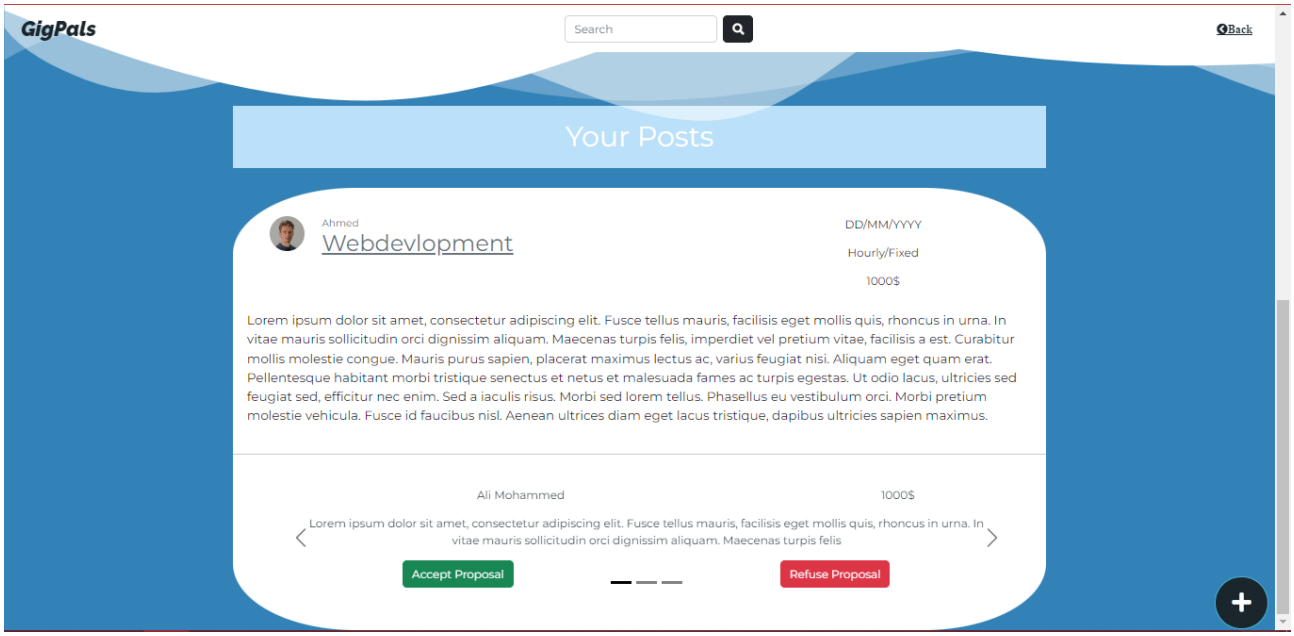
4. Freelancer profile



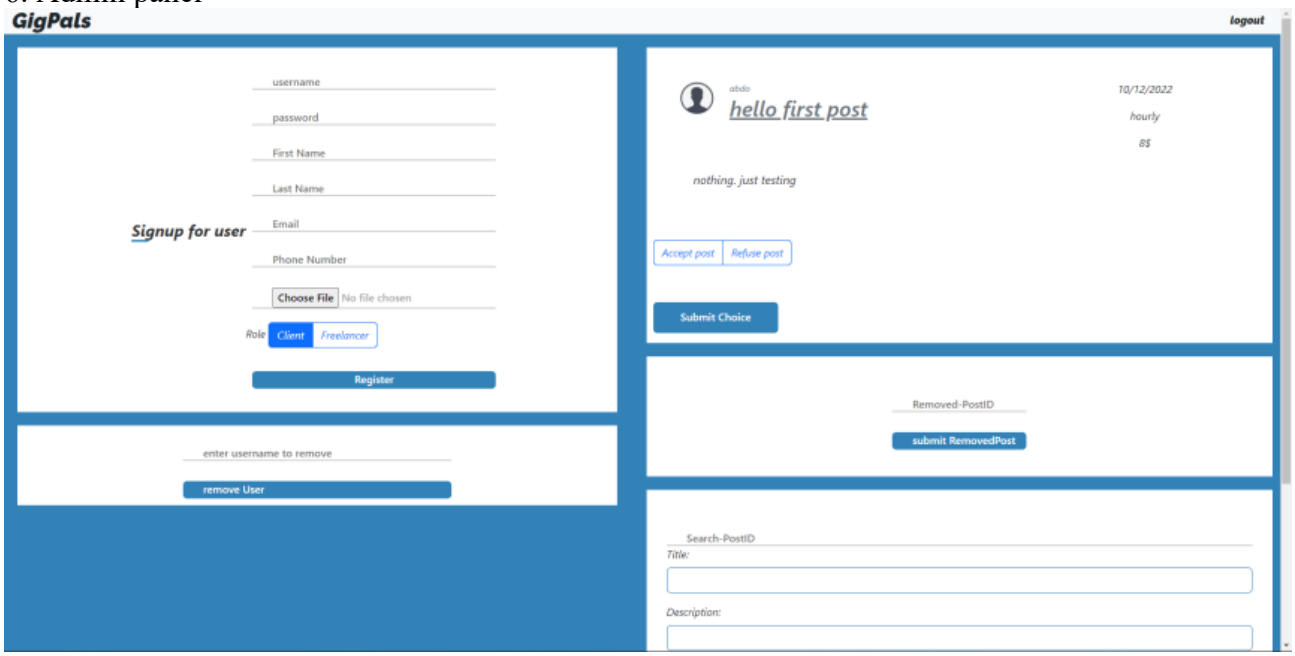


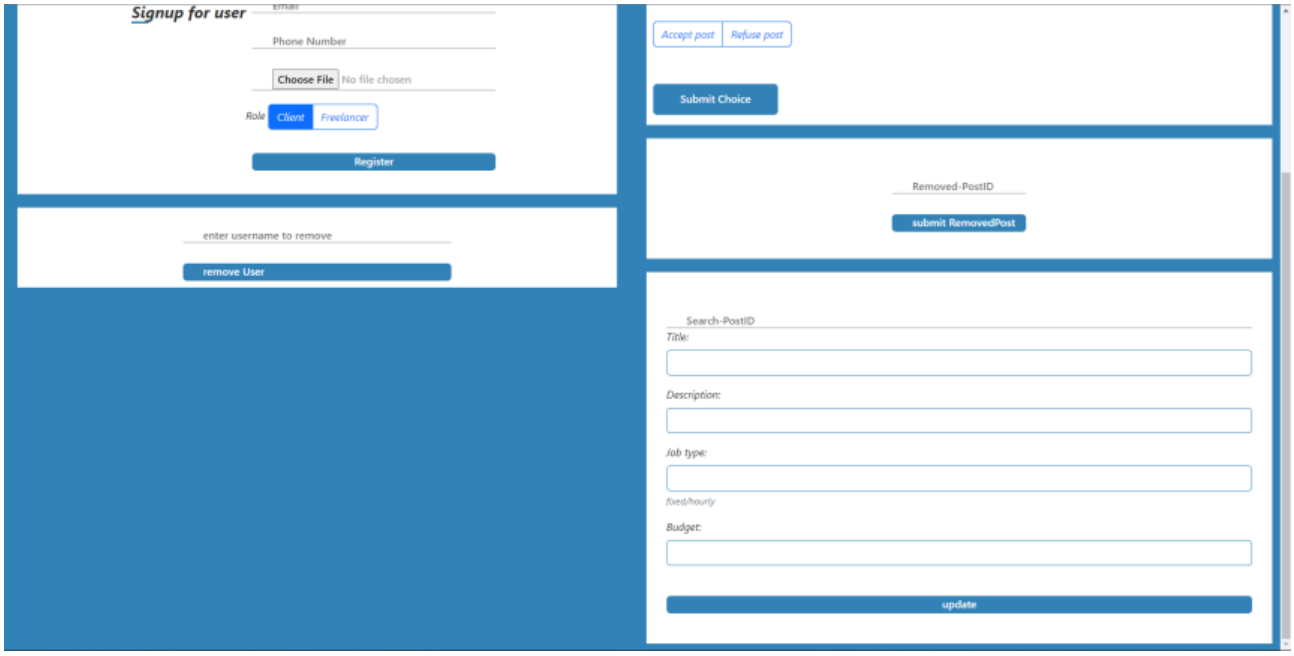
5. Client profile



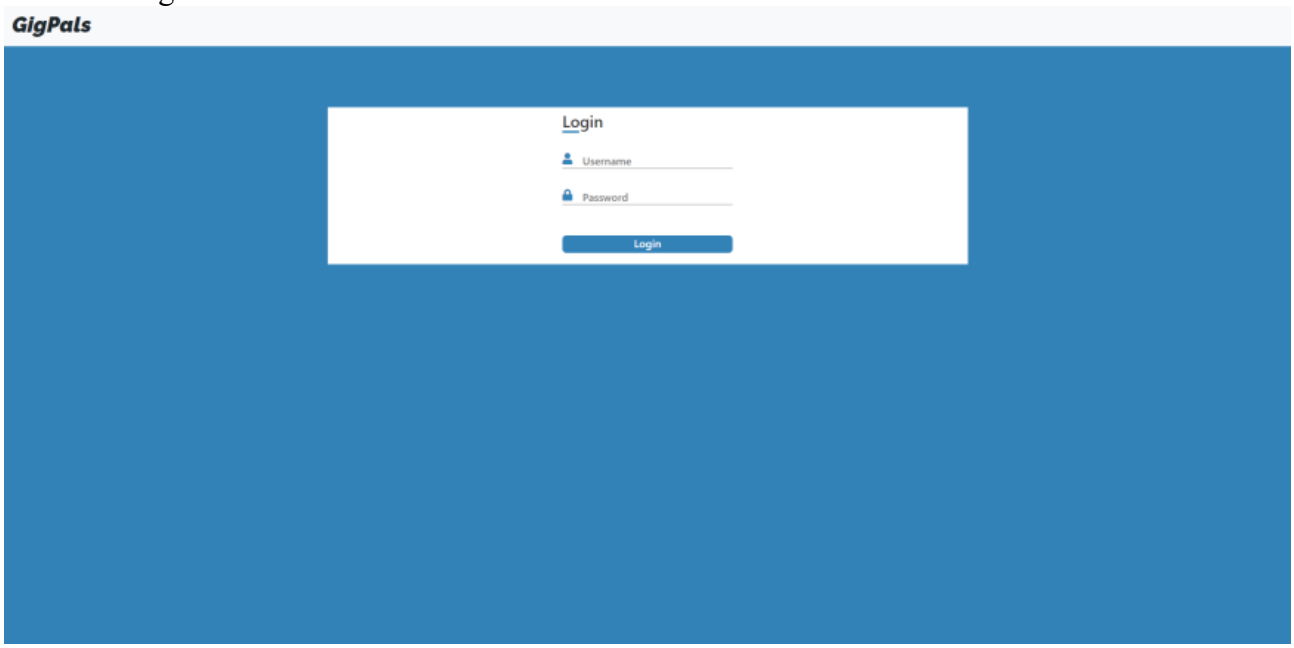


6. Admin panel





7. Admin login



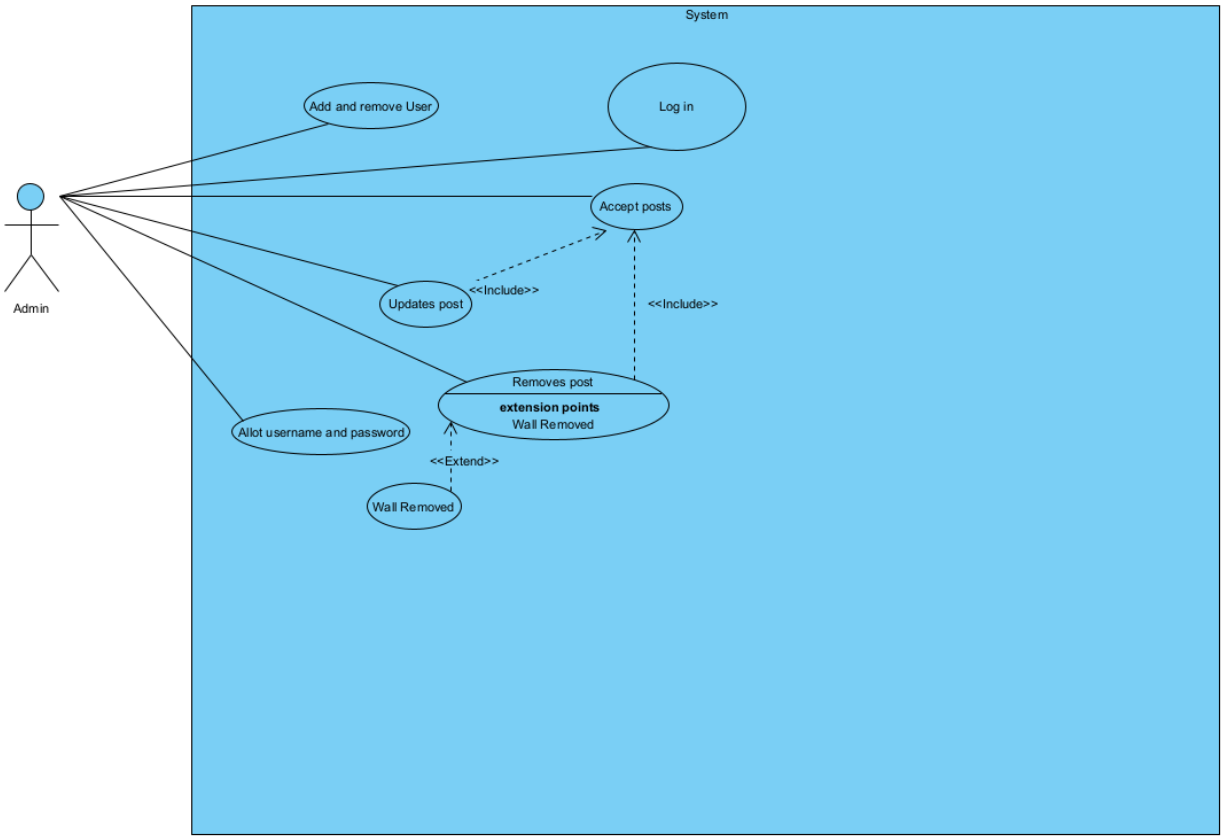
3.2 Hardware Interfaces

-GigPals works on all devices that have internet browser like chrome or safari or others…

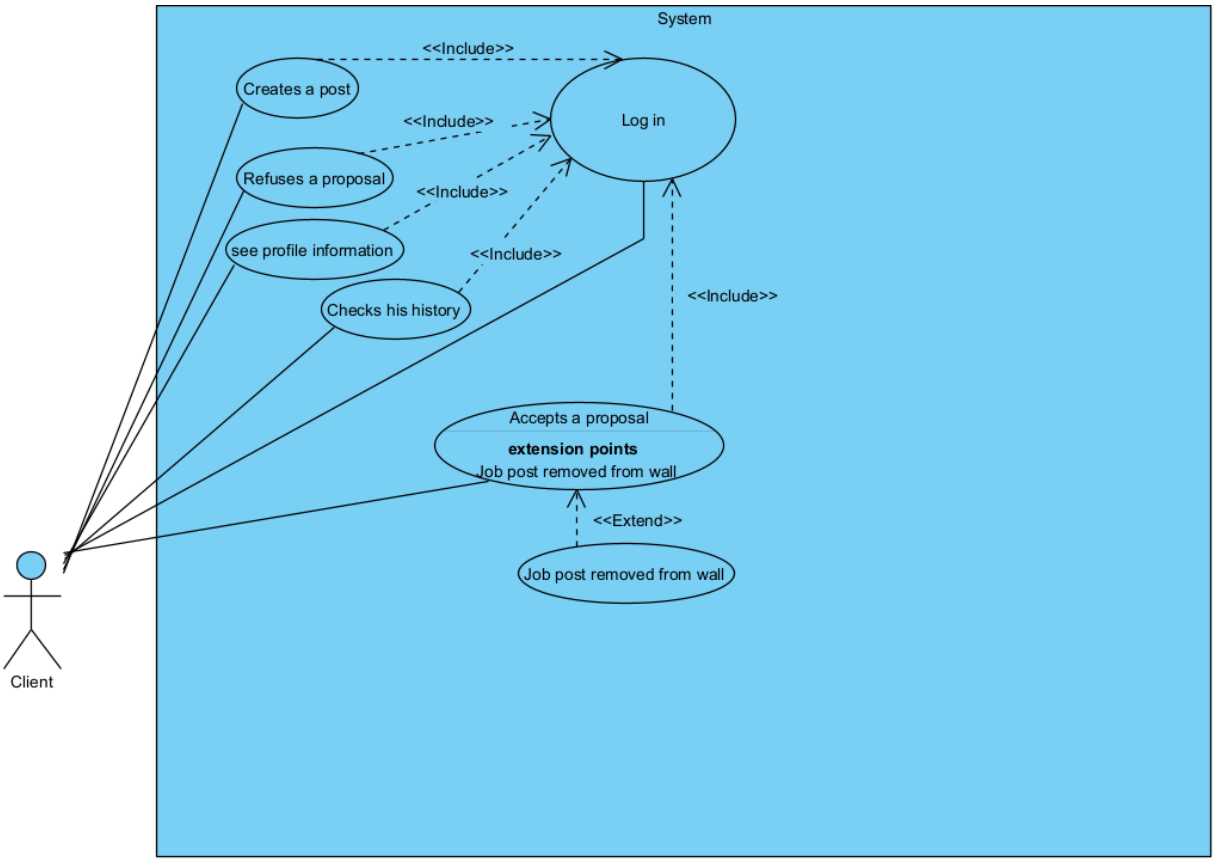
-Database is the concrete that the communication between users are recorded on.

3.3 Software Interfaces

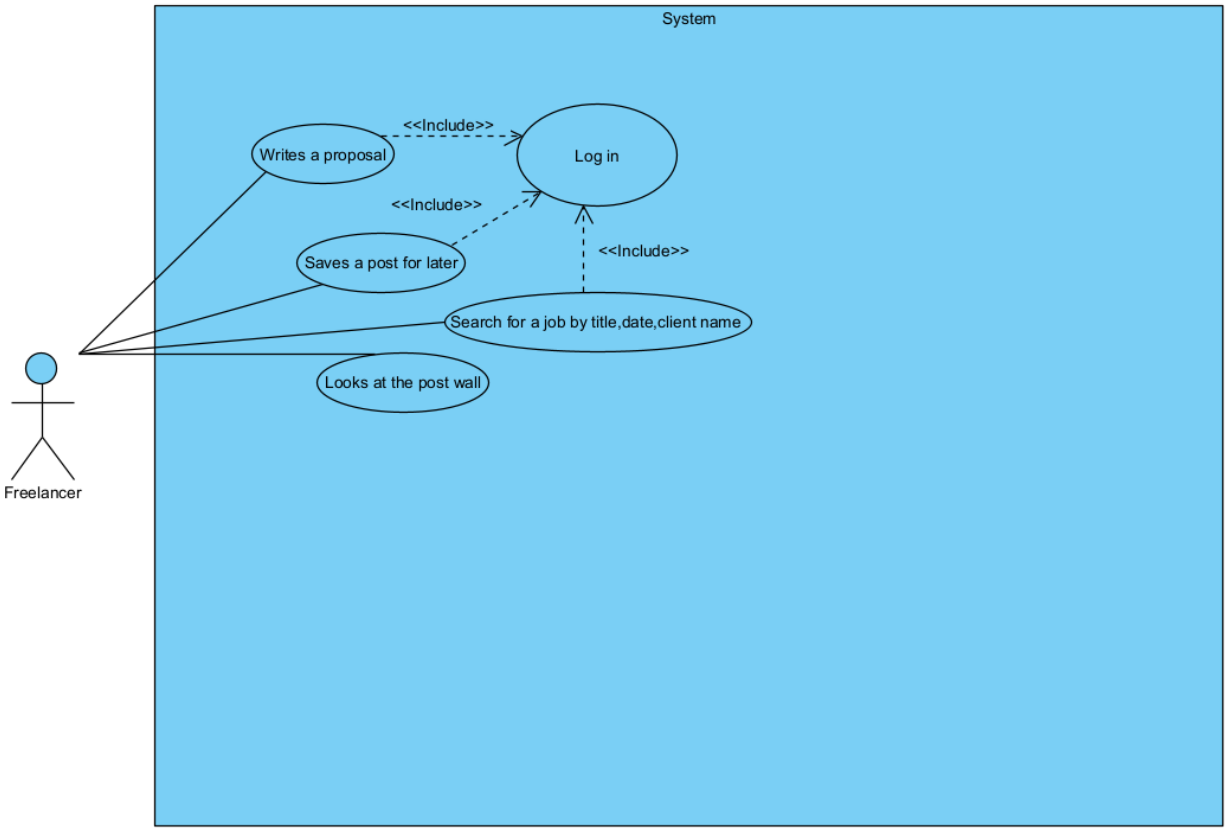
- Admin Use Case



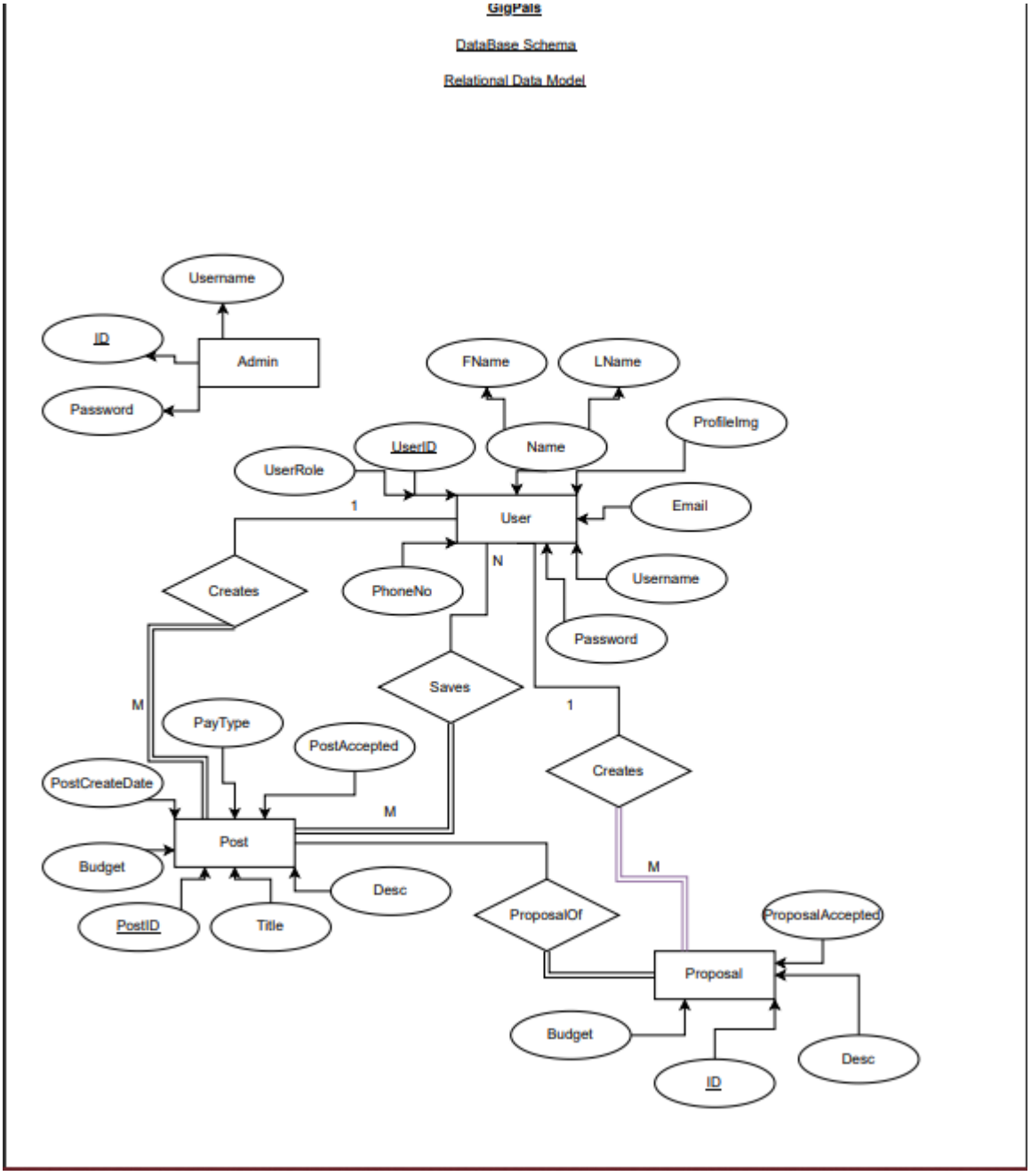
- Client Use Case



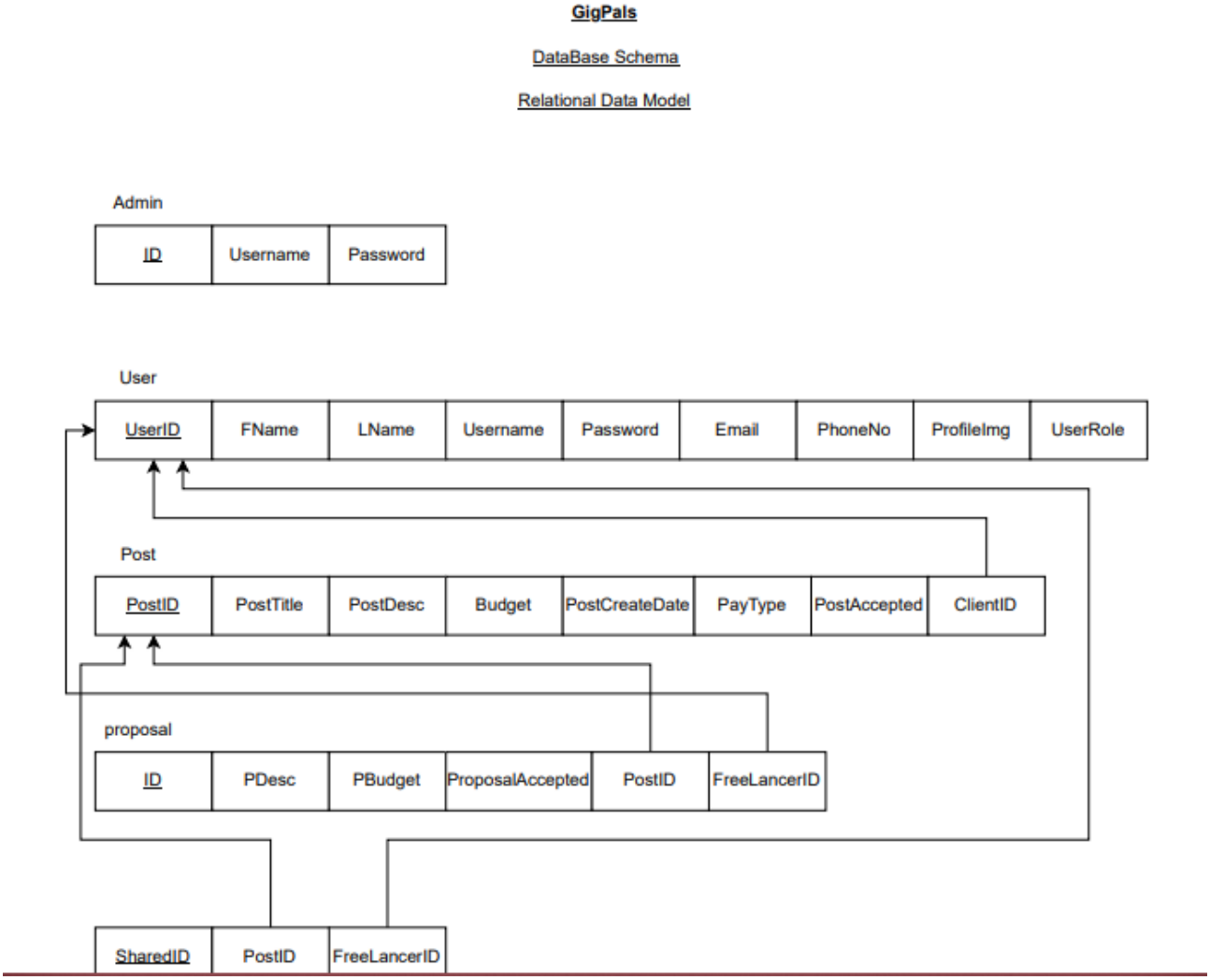
-Freelancer Use Case



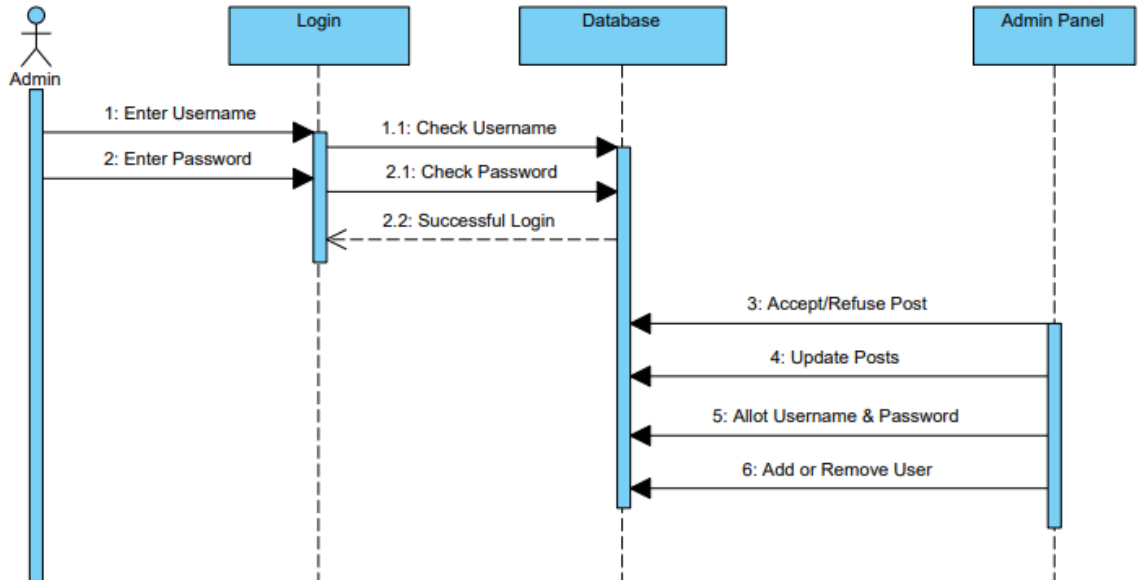
- Database Schema ERD



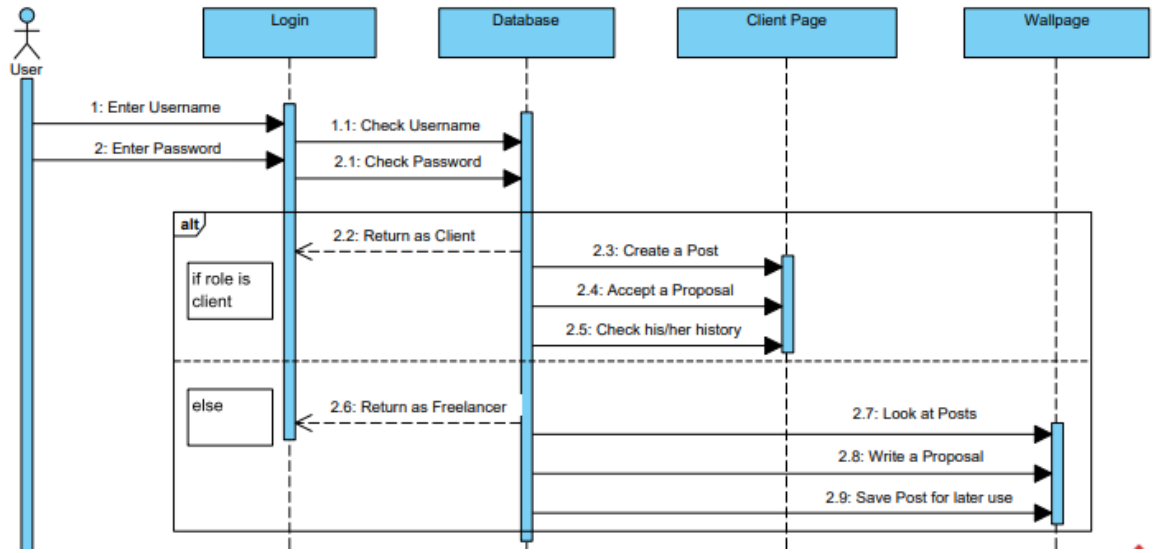
-Database Relational Data Model



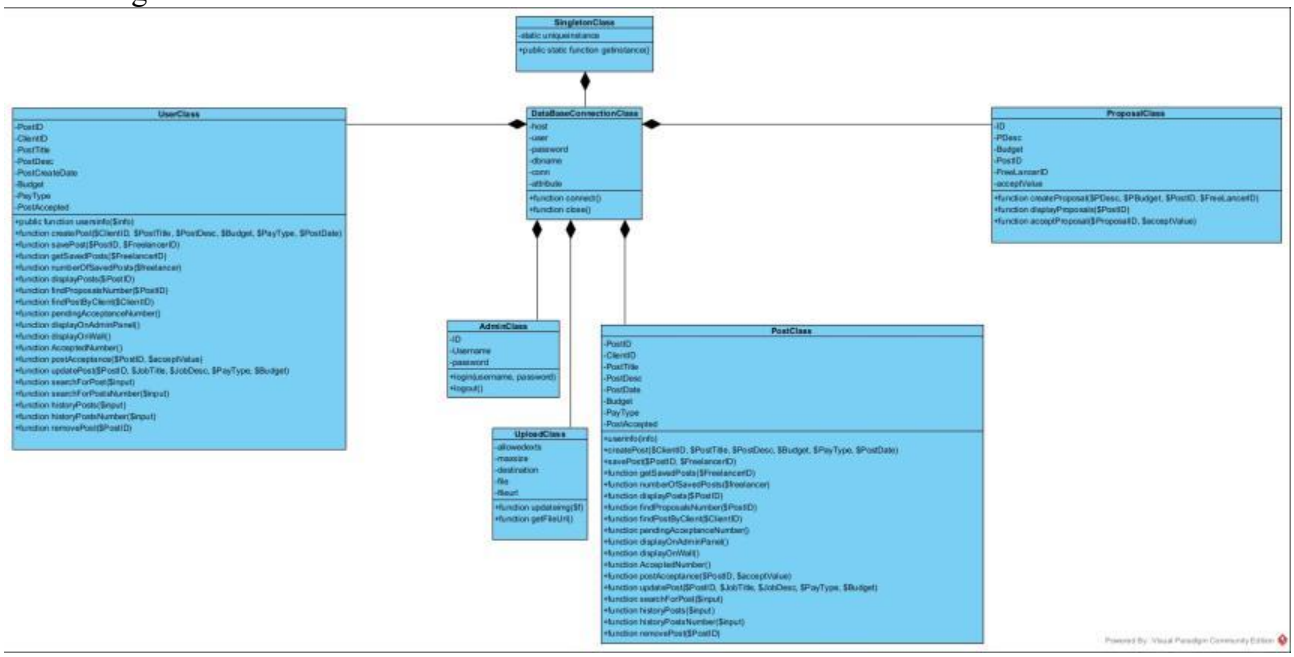
- Admin Sequence Diagram



-User Sequence Diagram

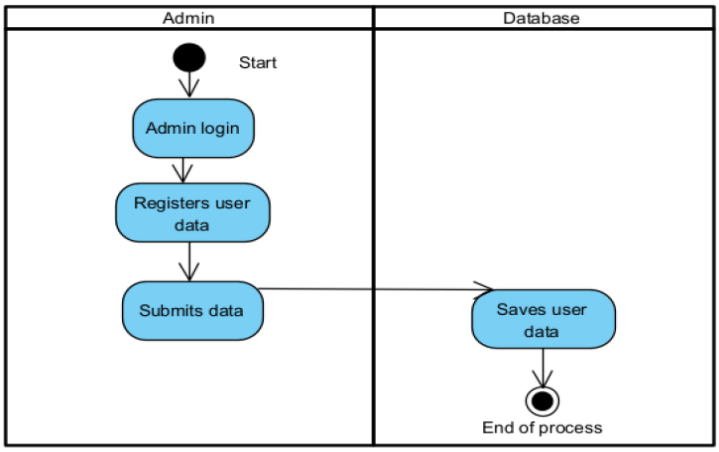


-Class diagram

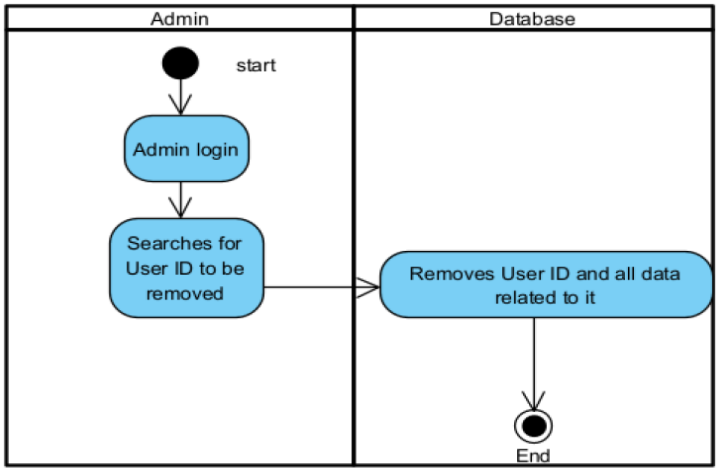


- Activity Diagram

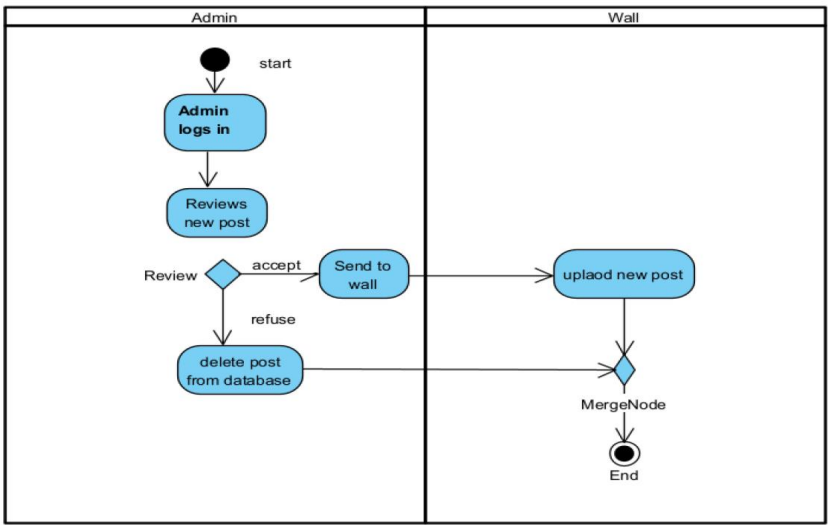
1) Add User



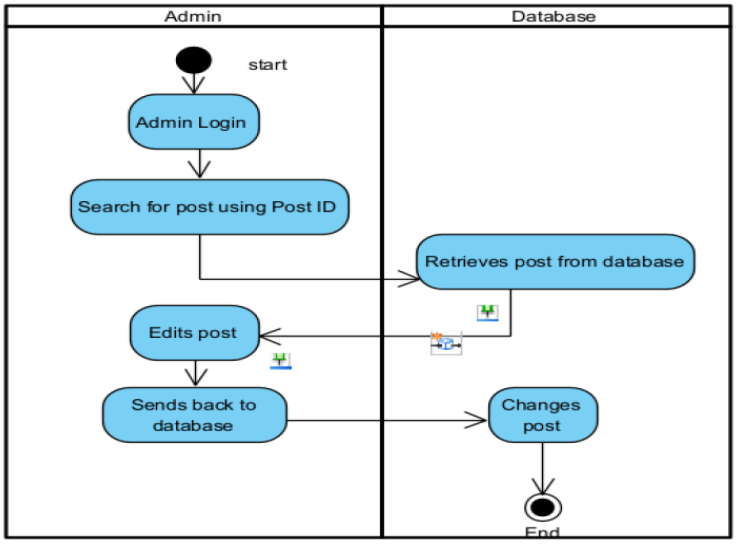
2) Remove User



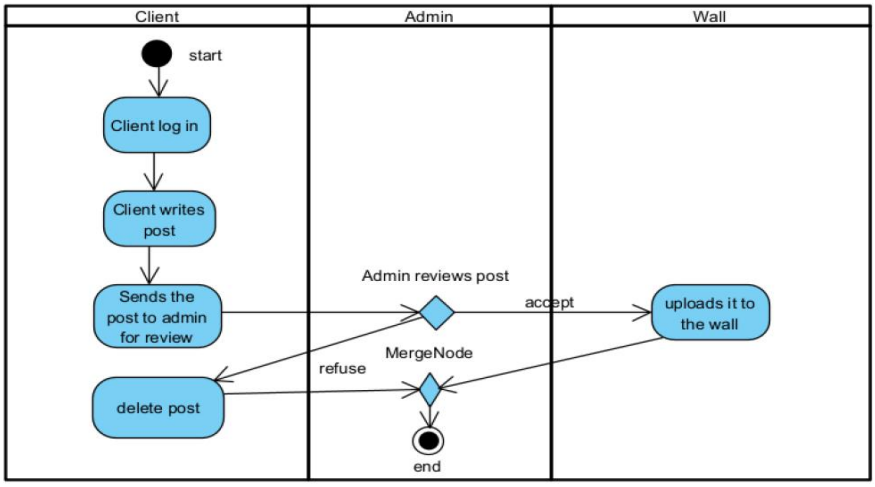
3) Post Review



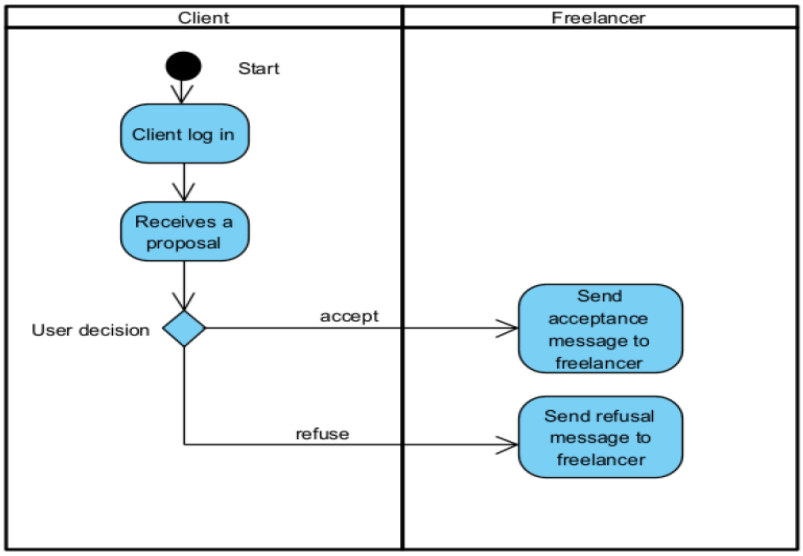
4) Post Update



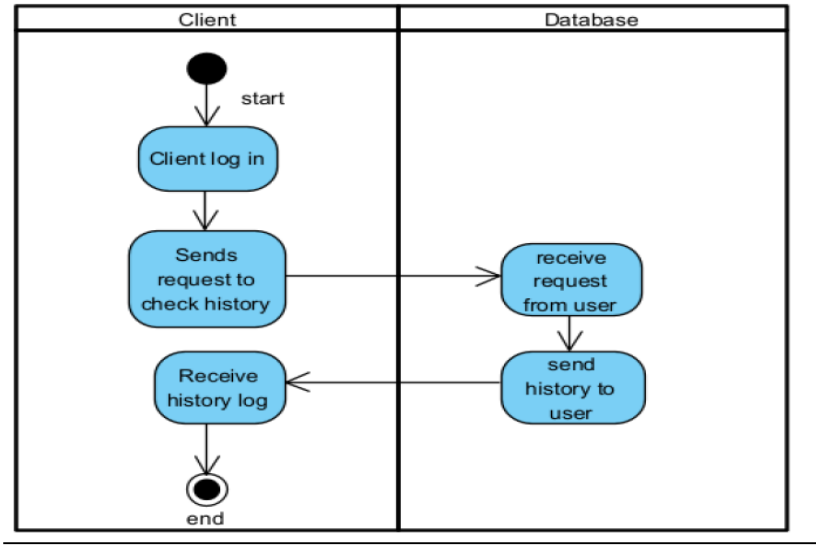
5) Client creates a post



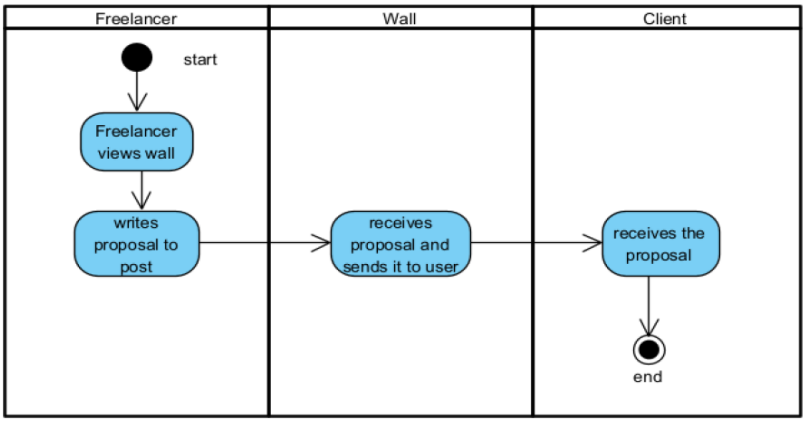
6) Proposal Review



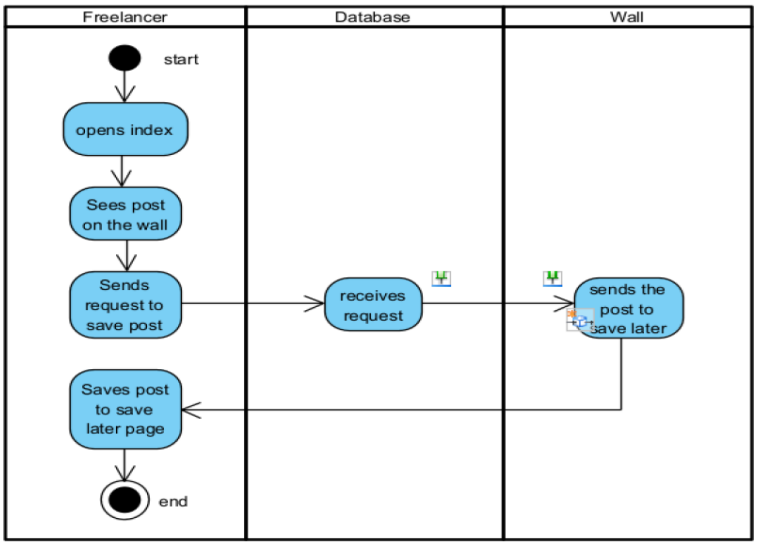
7) History Check



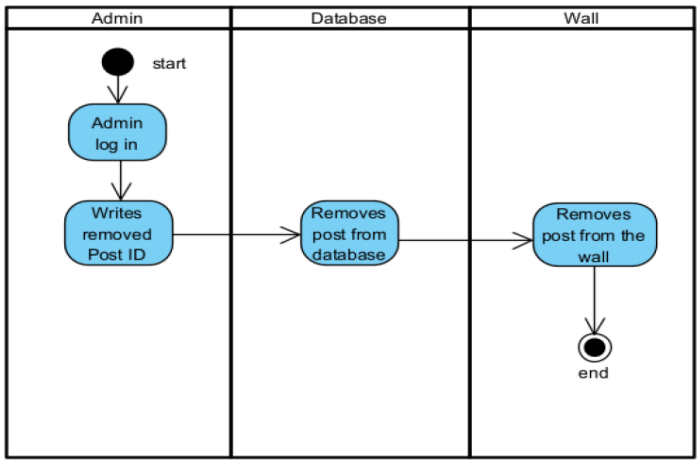
8) Proposal Creation



9) Save post for later



10) Remove post



4. System Features

4.1 Admin Interface

4.1.1 Description and Priority

The Admin has a very high priority as he is the one responsible for adding and removing users from the site and he gives them the username and password so they can log in.

4.1.2 Stimulus/Response Sequences

When the clients write their job post the admin decides first whether to accept or refuse it and if he accepts it he adds the post to the wall, after accepting the job post he can update or remove it and if the Admin removes the post, it reflects directly to the wall and be removed.

4.1.3 Functional Requirements

REQ-1: the admin must have username and password.
REQ-2: the admin need to log in to do his job.

4.2 Client interface

4.2.1 Description and Priority

The client is the person who goes to the site and want someone to do a certain job for him so he writes a job post and this post will be added to the wall after being accepted by the admin.

4.2.2 Stimulus/Response Sequences

When the clients log in to the site and the admin gives him the username and password he writes his job post and after accepting it the freelancer sees the post and send a proposal to the client then the client decides whether to accept or refuse it.

4.2.3 Functional Requirements

REQ-1: The Client need to log in to do his job.
REQ-2: The post must be accepted by the admin to be added to the wall.

4.3 Freelancer interface

4.3.1 Description and Priority

The freelancer is the person who is searching for a job in the site and want to make money so he sends proposal to the client.

4.3.2 Stimulus/Response Sequences

The freelancer can see all the job posts which is written by the client without logging in, when he log in he can search for a job by Title, Date or Client name and save a particular post in saved page to read later and apply to a job by sending a proposal to the client.

4.3.3 Functional Requirements

REQ-1: The Freelancer need to log in to do his job.
REQ-2: The proposal sent to the client for the job must be accepted to start working on it.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

GigPals is designed to work with minimum requirements, although in the current state it will require to on a PC that Has XAMPP and MySQL and Apache.

5.2 Safety Requirements

Currently in this version there are no Health issues that may be of scare to some users, but the user's data have a chance of being lost due to the use of unstable servers.

5.3 Security Requirements

Currently we have no protection on the data base that have the user data, other than the admin login stage.

5.4 Software Quality Attributes

-Responsive and easy to use Interface benefits both the users and admins.

- Most of the code files have documentation which benefits the Developers.