Photonest

Requirements Specification and Analysis

1.0

08.03.2020

Dilara Ünbay

Nursena Karakulah

Feyzullah Berkay Danış

Özay Ezerceli

Prepared for

SOFT3102 Software Engineering



Table of Contents

1. Introduction 1

1.1. Purpose of the System 1

1.2. Scope of the System 1

1.3. Objectives and Success Criteria of the Project 1

1.4. Definitions, Acronyms, and Abbreviations 1

1.5. Overview 1

2. Current System 2

3. Proposed System 2

3.1. Overview 2

3.2. Functional Requirements 2

3.3. Nonfunctional Requirements 3

Usability 3

Reliability 3

Performance 3

Supportability 3

Implementation 3

Interface 3

Packaging 4

Legal 4

3.4. System Models 4

Scenarios 4

Use case descriptions 12

Use case model 25

Object model 25

4. Glossary 26

5. References 26

REQUIREMENTS ANALYSIS DOCUMENT

# Introduction

## Purpose of the System

There are several social media applications that include photo sharing. Purpose of the Photonest is to set photo sharing as the main focus through a simplified user interface (UI) and an enhanced like system. The simplified user interface (UI) aims to make navigation easier throughout the application to provide a shorter and enjoyable learning phase. The enhanced like system intends to increase the interaction between users by offering a more positively competitive environment.

Overall, Photonest targets to be a quick, easy and enjoyable option for users of all ages.

## Scope of the System

Photonest is a social media application in which users share photos as posts. Each user can have a profile, follow users, share a photo, tag other users, use hashtags, add locations, leave comments and like photos. Users are able to delete their profiles, posts and comments and take back their follows and likes. If users wish to change something rather than deleting, they can edit their profiles and posts. Also, users have the ability to block other users.

## Objectives and Success Criteria of the Project

* Project meets the functional requirements.
* Project meets the non-functional requirements.
* Project provides the items described on the scope.
* Project is completed on time.
* 20% of the bugs found in the project are fixed.
* 50% of the feedbacks are positive.
* Project documentation is delivered on time.

## Definitions, Acronyms, and Abbreviations

UI – User Interface.

RAD – Requirements Analysis Document.

Firebase Realtime Database – A cloud-hosted database provided by Google which allows to store and sync data.

## Overview

Rest of the RAD explains the differences from existing systems, high-level and user-level functional overview, supported features, usage of processes and functional specification of the project.

# Current System

Photonest is a photo sharing based social media app similar to Instagram. Photonest, uses Firebase Realtime Database to store data unlike Instagram. It aspires to have a cleaner and simpler UI to avoid overwhelming and losing users. It uses a number based like system to allow users to show the degree of their liking and to present a quick feedback while creating a positively competitive environment. Photonest, aims to sustain a persistent user group through its simple, fun and competitive nature.

# Proposed System

## Overview

This section provides further information regarding the supported features by describing the requirements.

## Functional Requirements

* Users should be able to have profiles.
* Users should be able to add photos.
* Users should be able to tag other users.
* Users should be able to post photos.
* Users should be able to add locations.
* Users should be able to add comments.
* Users should be able to logout.
* Users should be able to delete their profiles.
* Users should be able to delete their comments.
* Users should be able to use hashtags.
* Users should be able to like posts.
* Users should be able to follow other users.
* Users should be able to unlike posts.
* Users should be able to unfollow other users.
* Users should be able to block other users.
* Users should be able to have profile photos.
* Users should be able to login by using their emails and passwords.
* The system should be able to send emails to the users.
* Users should be able to change their passwords.
* Users should be able to change their usernames.
* Users should be able to delete their posts.
* Users should be able to add descriptions.

## Nonfunctional Requirements

### Usability

* The system should be easy to use by adult users.
* The users should be able to use the system without any training.
* The system should be easy to navigate and understand such that a user should be able to successfully complete their profile within 10 minutes on their first time.
* Users with poor English skills should be able to use the system.
* The users should be able to reach their profiles with one click.

### Reliability

* The system should be able to prevent restarting on failures 85% of the time.
* When the system fails to share a post, the user should be able to retry sharing it without creating a new post.
* The system should be able to back-up all data automatically every day.

### Performance

* The system should be able to create a new user profile within 3 seconds.
* The feed page load time of the system should be no more than 6 seconds.
* The system should be available 95% of the time.
* The system should log in a user within 10 seconds.
* The system should not lose any user data.

### Supportability

* The system should support the English language.
* The system should be able to support 5% of concurrent usage without failing to satisfy the performance requirements.
* The database of the system should be able to support 15% of growth without losing any performance.

### Implementation

* The system should run on at least 87% of Android devices.
* The system should use Firebase Realtime Database as its database.
* The system should be written in Android Studio.

### Interface

* The buttons should change color upon clicking.
* The system complies with the software interfaces defined by Android 5.0 API Lollipop.
* The system should have a profile, search, add, feed and notification button on every page as a navigation bar.

### Packaging

* The system should have an APK as an installation package.

### Legal

* The system should be licensed as freeware.

## System Models

### Scenarios

**1)**  *Scenario name* **CuteGuy***Participating* Anna:The User *actorsFlow of events*

* + - 1. Anna is enjoying a coffee with her friend. Her friend wants to show her a cute photo of her crush. However, she is running low on her battery and Anna is not following the guy. Her friend suggests using her account instead. So, Anna activates the “Logout” function.
      2. Anna receives a text asking if she is sure about logging out. She confirms it and passes her phone to her friend in excitement.

**2)**  *Scenario name* **DrinkingGame***Participating* Boris:The Visitor *actorsFlow of events*

* + - 1. Boris is drinking with his friends. His friends challenge him in a game suggesting that if he loses, he will join Photonest like the rest of them. He accepts the terms and loses the game. So, Boris gets his phone activates the “Register User” function.
      2. Boris enters his e-mail and name. After thinking for a while, he decides on a username and enters it too. Finally, he enters a password and submits the form.
      3. Boris checks his e-mail and sees a mail from Photonest. He opens the mail and clicks on the verification link.
      4. Boris reads the text and activates his account wondering why he had to accept the game.

**3)**  *Scenario name* **NewPhone***Participating* Emily:The Visitor *actorsFlow of events*

* + - 1. Emily got a brand-new phone and finished downloading her favorite applications. She decided to start with Photonest and opens it. So, she activates the “Login” function.
      2. Emily enters her e-mail and password and submits it. After waiting for a few seconds, she sees her home page.

**4)**  *Scenario name* **SickAndBored***Participating* Laura:The Visitor *actorsFlow of events*

* + - 1. Laura is sick and she is very bored from staying in bed and doing nothing. While thinking for ideas to entertain her, she remembers her Photonest profile. She heads to application and tries to login but fails several times. So, she activates the “Forgot Password” function.
      2. Laura enters her most used e-mail and submits it hoping that it is the right one.
      3. Shortly after Laura receives a mail from Photonest and happily clicks on the link.
      4. Laura enters a new password and confirms it. She immediately sees her home page and starts to explore.

**5)**  *Scenario name* **BadBreakup***Participating* Cassandra:The User *actorsFlow of events*

* + - 1. Cassandra finally broke up with her controlling boyfriend but she thinks that her ex-boyfriend still has her Photonest password. So, she activates the “Change Password” function.
      2. Cassandra enters her current password and new password that she thinks is very hard to guess. She is relieved to see her home page after submitting the new password.

**6)**  *Scenario name* **VacationPhoto***Participating* Mike:The User *actorsFlow of events*

* + - 1. Mike went to Japan for a vacation. One morning he saw an amazing road filled with Sakura trees and took a photo of it. He really likes the photo and wants to share it with his friends. So, he activates the “Create Post” function.
      2. Mike finds the photo from his gallery and selects it. He likes the preview of the photo and proceeds.
      3. Mike adds a description stating how beautiful the road was and uses a Sakura hashtag. He also tags his best friend who couldn’t come to the vacation. He likes the look of his post and shares it.

**7)**  *Scenario name* **MisspellingMom***Participating* Adam:The User *actorsFlow of events*

* + - 1. Adam has a photo taken with his mom from his graduation day. He shared the photo thinking that it’s a nice way to announce his graduation. However, he realizes that he misspelled mom in his post. So, he activates the “Edit Post” function.
      2. Adam quickly corrects the word mom in his description and reads it again to make sure everything is correct. Then he submits the new description.

**8)**  *Scenario name* **GetIntoUniversity***Participating* Molly:The User *actorsFlow of events*

* + - 1. Molly found out that she got into her desired university. Seeing that her friends wrote their universities in their profile bios, she wanted to do the same. So, she activates the “Edit Profile” function.
      2. Molly adds her university to her bio and confirms the change.

**9)**  *Scenario name* **NewHaircut***Participating* Dennis:The User *actorsFlow of events*

* + - 1. Dennis got a new haircut and style inspired from Vikings. He really loved the way it looked combined with his neck tattoo hence he took a selfie. He really liked how the photo turned out. So, he activates the “Edit Profile Photo” function.
      2. Dennis chooses the library option because he already has the photo in his gallery.
      3. Dennis selects his new photo and confirms it.

**10)**  *Scenario name* **FightWithFriend***Participating* Sam:The User *actorsFlow of events*

* + - 1. Sam had a huge fight with her best friend and she doesn’t think that they’ll ever talk again. She has their photo as her profile photo but she thinks that its not okay to use it anymore. She doesn’t know what else to use. So, she activates the “Delete Profile Photo” function.
      2. Sam receives a warning asking if she is sure about deleting it. After thinking for a few seconds, she encourages herself and confirms it.

**11)**  *Scenario name* **AntiVaxxerPosts***Participating* Karen:The User *actorsFlow of events*

* + - 1. Karen recently changed her mind on being an anti-vaxxer. She wants to delete her anti-vaxxer posts because she believes that they are misguiding people like her. So, she activates the “Delete Post” function.
      2. Karen reads the warning that asks if she is sure about deleting her post and confirms it without doubt.

**12)**  *Scenario name* **LostBrother***Participating* Bran:The User *actorsFlow of events*

* + - 1. Bran recently lost his brother to cancer. Every time he sees posts about family, he is reminded of his brother and how much he misses him. He thinks that it affects his mental state and wants to delete his profile. So, he activates the “Delete Profile” function.
      2. Bran receives a warning that says he won’t be able to recover his profile. He confirms it anyway in hopes of getting better.

**13)**  *Scenario name* **OldFriend***Participating* Bob:The User *actorsFlow of events*

* + - 1. Bob remembers the full name of his childhood crush. He wonders how she is doing in life. He decides to look for her thinking maybe she is using Photonest as well. So, he activates the “View Search” function.
      2. Bob writes his childhood crush’s name and sees a few users getting listed. However, none of them seems to be the person he is looking for.

**14)**  *Scenario name* **ValentinesCake***Participating* Bella:The User *actorsFlow of events*

* + - 1. Bella sees an amazing chocolate cake while searching for valentine’s day presents. The post also has the recipe of the cake in the description. Bella is happy to find this post because it is exactly what she was looking for. So, she activates the “Like Post” function.
      2. Bella chooses 5+ to give a like because she wants to show her gratitude to the poster for being such a life saver.

**15)**  *Scenario name* **FollowSister***Participating* James:The User *actorsFlow of events*

* + - 1. James receives a message from his little sister. His sister writes that she joined Photonest and followed him there. James says that he’ll happily return the favor. So, he goes to his little sister’s profile and activates the “Follow User” function.

**16)**  *Scenario name* **BirthdayMessage***Participating* John:The User *actorsFlow of events*

* + - 1. John sees a post of his classmate holding an eighteen balloon. He reads the description and finds out that it is his classmate’s birthday. He decides to send his best wishes. So, he activates the “Add Comment” function.
      2. John writes a short text to wish a happy birthday to his classmate and sends his comment.

**17)**  *Scenario name* **NegativeComment***Participating* Nina:The User *actorsFlow of events*

* + - 1. Nina left a negative comment on one of her friend’s posts stating that the color of her purse doesn’t match at all. After receiving a few upset comments, she decided to delete her comment. So, she finds her comment activates the “Delete Comment” function.
      2. Nina deletes her comment and tells her friend in private that she didn’t mean to upset her.

**18)**  *Scenario name* **ToxicFriend***Participating* Lilly:The User *actorsFlow of events*

* + - 1. Lilly keeps receiving sexual jokes from a friend under every single one of her posts. His behavior didn’t change despite her warnings. She decides to end it for good. So, she goes to his profile and activates the “Block User” function.
      2. Lilly receives a text which informs her that this user won’t be able to see her or interact with her. She confirms it and blocks him happily.

### Use case descriptions

**1)**  *Use case name* **Logout***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Logout” function from the profile page of the application by clicking the “Logout” option.  
   * 1. **SYSTEM** presents a warning text to the **User**.
2. The **User** reads the message and clicks the “Logout” button.
   * 1. **SYSTEM** responds by re-directing the **Visitor** to the login page.

*Entry condition* The **User** has clicked the Logout option.*Exit conditions* The **Visitor** has been re-directed to the login page.

The **User** has clicked on the Cancel button.   
*Quality* The **Visitor** is immediately re-directed to the login page.  *Requirements*

**2)**  *Use case name* **RegisterUser***Participating* Initiated by **Visitor**  *actorsFlow of events*

1. The **Visitor** activates the “Register User” function from the login page of the application by clicking the “Sign Up” text.  
   * 1. **SYSTEM** presents a sign-up form to the **Visitor**.
2. The **Visitor** fills the form by entering an e-mail, a full name, a username and a password. The **Visitor** submits the form by clicking the “Sign Up”.
   * 1. **SYSTEM** receives the sign-up form, checks the given e-mail and sends a verification mail to the given e-mail.

5.  The **Visitor** receives the verification mail and clicks the link to activate his/her account.

6. **SYSTEM** presents an informing message to the **Visitor**. 

7.  The **Visitor** reads the message and ends the sign-up process by clicking the “Activate” button. 

8. **SYSTEM** presents a success message to the **User**.

*Entry condition* The **Visitor** has clicked the Sign-Up button.*Exit conditions* The **Visitor** has received a success message.   
*Quality* The success message is shown immediately after the **Visitor** clicks the *requirements* Activate button.

**3)**  *Use case name* **Login***Participating* Initiated by **Visitor**  *actorsFlow of events*

1. The **Visitor** opens the application.  
   * 1. **SYSTEM** presents a login form to the **Visitor**.
2. The **Visitor** fills the form by entering an e-mail and a password. The **Visitor** submits the form by clicking the “Login” button and activates “Login” function.
   * 1. **SYSTEM** receives the login form, checks the given information and re-directs the **User** to the home page.

*Entry condition* The **Visitor** has opened the application.*Exit conditions* The **User** has been re-directed to the home page.   
*Quality* The **Visitor** is re-directed to the home page within 10 seconds.  *Requirements*

**4)**  *Use case name* **ForgotPassword***Participating* Initiated by **Visitor**  *actorsFlow of events*

1. The **Visitor** activates the “Forgot Password” function from the login page of the application by clicking the “Forgot Password?” text.  
   * 1. **SYSTEM** responds by presenting an informing text and asking for an e-mail from the **Visitor**.
2. The **Visitor** enters an e-mail and submits it by clicking the “Reset Password” button.
   * 1. **SYSTEM** receives the given e-mail, checks it and sends a reset mail to the given e-mail.

5.  The **Visitor** receives the reset mail and clicks the link to change his/her password.

6. **SYSTEM** presents a password reset form to the **Visitor**. 

7.  The **Visitor** fills the form by entering a new password and submits the form by clicking the “Reset Password” button. 

8. **SYSTEM** receives the form, checks the given information and re-directs the **User** to the home page.

*Entry condition* The **Visitor** has clicked the Forgot Password text.*Exit conditions* The **User** has been re-directed to the home page.   
*Quality* The **User** is immediately re-directed to the home page.  *Requirements*

**5)**  *Use case name* **ChangePassword***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Change Password” function from the profile page of the application by clicking the “Change Password” option.  
   * 1. **SYSTEM** responds by presenting a password form to the **User**.
2. The **User** fills the form by entering the current password and a new password twice. The **User** submits it by clicking the “Change” button.
   * 1. **SYSTEM** receives the form, checks the given information and re-directs the **User** to the home page.

*Entry condition* The **User** has clicked the Change Password option.*Exit conditions* The **User** has been re-directed to the home page.   
*Quality* The **User** is immediately re-directed to the home page.  *Requirements*

**6)**  *Use case name* **CreatePost***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Create Post” function from the home page of the application by clicking the plus icon.  
   * 1. **SYSTEM** responds by presenting a selection screen and including the gallery of the **User**.
2. The **User** selects a photo from the included gallery and clicks on the “Next” text.
   * 1. **SYSTEM** presents a preview of the photo and a form to the **User**.

5.  The **User** fills the form by adding a description, hashtags and tags. The **User** submits the form by clicking “Share” text.

6. **SYSTEM** receives the form and re-directs the **User** to the home page.

*Entry condition* The **User** has clicked the Plus icon.*Exit conditions* The **User** has been re-directed to the home page.

The **User** has clicked on the Back button and reached the home page.  
*Quality* The **User** is immediately re-directed to the home page.  *Requirements*

**7)**  *Use case name* **EditPost***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Edit Post” function from the post review page by clicking the “Edit” option.  
   * 1. **SYSTEM** responds by presenting a filled-out form with the photo preview to the **User**.
2. The **User** makes changes to the description, hashtags and tags. The **User** submits the changed form by clicking the “Done” button.

4. **SYSTEM** applies changes and re-directs the **User** to the post preview page.

*Entry condition* The **User** has clicked the Edit option.*Exit conditions* The **User** has been re-directed to the post preview page.

The **User** has clicked on the Cancel button.   
*Quality* The **User** is immediately re-directed to the post preview page.  *Requirements*

**8)**  *Use case name* **EditProfile***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Edit Profile” function from the profile page by clicking the Edit Profile button.  
   * 1. **SYSTEM** responds by presenting a filled-out profile form to the **User**.
2. The **User** makes changes to the full name, username, website link and description. The **User** submits the changed form by clicking the “Done” button.

4. **SYSTEM** applies changesand re-directs the **User** to the profile page.

*Entry condition* The **User** has clicked the Edit Profile button.*Exit conditions* The **User** has been re-directed to the profile page.

The **User** has clicked on the Cancel button.   
*Quality* The **User** is immediately re-directed to the profile page.  *Requirements*

**9)**  *Use case name* **EditProfilePhoto***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Edit Profile Photo” function from the profile page by clicking the profile photo.  
   * 1. **SYSTEM** responds by presenting a selection to the **User**.
2. The **User** clicks on the Choose from Library option.
   * 1. **SYSTEM** presents a preview page and includes the gallery of the **User**.

5.  The **User** selects a photo and submits the photo by clicking the “Done” button.

6. **SYSTEM** applies changesand re-directs the **User** to the profile page.

*Entry condition* The **User** has clicked the profile photo.*Exit conditions* The **User** has been re-directed to the profile page.

The **User** has clicked on the Cancel button.

The **User** has clicked on the Cancel option of the selection.   
*Quality* The **User** is immediately re-directed to the profile page.  *Requirements*

**10)**  *Use case name* **DeleteProfilePhoto***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** clicks on the profile photo on the profile page.  
   * 1. **SYSTEM** responds by presenting a selection to the **User**.
2. The **User** activates the “Delete Profile Photo” function by clicking on the “Delete Current Photo” option.
   * 1. **SYSTEM** presents a warning text and a selection to the **User**.

5.  The **User** reads the warning and clicks on the “Yes” button.

6. **SYSTEM** receives the selection, applies the change and re-directs the **User** to the profile page.

*Entry condition* The **User** has clicked on the Delete Current Photo option.*Exit conditions* The **User** has been re-directed to the profile page.

The **User** has clicked on the Cancel button.

The **User** has clicked on the No option of the selection.   
*Quality* The **User** is immediately re-directed to the profile page.  *Requirements*

**11)**  *Use case name* **DeletePost***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Delete Post” function from the post review page by clicking the Delete option.  
   * 1. **SYSTEM** responds by presenting a warning text and a selection to the **User**.
2. The **User** reads the warning text and clicks on the “Yes” button.
   * 1. **SYSTEM** receives the selection, deletes the post and re-directs the **User** to the initial page.

*Entry condition* The **User** has clicked the Delete option.*Exit conditions* The **User** has been re-directed to the initial page.

The **User** has clicked on the No option of the selection.   
*Quality* The **User** is immediately re-directed to the initial page.  *Requirements*

**12)**  *Use case name* **DeleteProfile***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Delete Profile” function from the profile page by clicking the “Delete Profile” option.  
   * 1. **SYSTEM** responds by presenting a warning text and a selection to the **User**.
2. The **User** reads the warning text and clicks on the “Yes” button.
   * 1. **SYSTEM** receives the selection, deletes the profile and re-directs the **Visitor** to the login page.

*Entry condition* The **User** has clicked the Delete Profile option.*Exit conditions* The **Visitor** has been re-directed to the login page.

The **User** has clicked on the No option of the selection.   
*Quality* The **Visitor** is re-directed to the login page within 5 seconds.  *Requirements*

**13)**  *Use case name* **ViewSearch***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** clicks on the “Search” button.  
   * 1. **SYSTEM** responds by presenting a search page to the **User**.
2. The **User** enters a word or a sentence about what he/she wants to find and activates the “View Search” function.
   * 1. **SYSTEM** presents a list of related results that aredivided into two sections of profiles and hashtags to the **User**.

*Entry condition* The **User** has clicked on the Search button.*Exit conditions* The **User** has received the list of results.

*Quality* The **User** receives the list of results within 5 seconds.  *Requirements*

**14)**  *Use case name* **LikePost***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Like Post” function by clicking on the Like button of the selected post.  
   * 1. **SYSTEM** responds by presenting different icons to choose from to the **User**.
2. The **User** clicks on his/her desired icon and submits the like.
   * 1. **SYSTEM** receives the submission and updates the likes of the selected post.

*Entry condition* The **User** has clicked on the Like button.*Exit conditions* The post’s likes has been updated.

The **User** has clicked the like button twice.   
*Quality* The like button changes color upon liking.  *Requirements*

**15)**  *Use case name* **FollowUser***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Follow User” function from the profile page by clicking the “Follow” button.  
   * 1. **SYSTEM** responds by sending a follow request to the selected **User**.

*Entry condition* The **User** has clicked on the Follow button.*Exit conditions* The other **User** has received a follow request.

The **User** has clicked on the Follow button twice.   
*Quality* The follower button changes color upon sending a follow request.  *Requirements*

**16)**  *Use case name* **AddComment***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** clicks on the “Comment” button.  
   * 1. **SYSTEM** responds by presenting a form with the comments of the selected post to the **User**.
2. The **User** fills the form by writing his/her desired comment and submits it by clicking the “Send” button.
   * 1. **SYSTEM** receives the submission and updates the comments of the selected post.

*Entry condition* The **User** has clicked on the Comment button.*Exit conditions* The post’s comments has been updated.   
*Quality* The **User** immediately sees his/her comment.  *Requirements*

**17)**  *Use case name* **DeleteComment***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** clicks on the comment he/she wants to delete.  
   * 1. **SYSTEM** responds by selecting the comment and presenting two buttons to the **User**.
2. The **User** clicks on the “Delete” button and activates the “Delete Comment” function.
   * 1. **SYSTEM** deletes the comment and presents a success message to the **User**.

*Entry condition* The **User** has clicked on the Comment.*Exit conditions* The **User** has seen the success message.

The **User** has clicked on the Cancel button.   
*Quality* The comment is highlighted upon selection.  *Requirements*

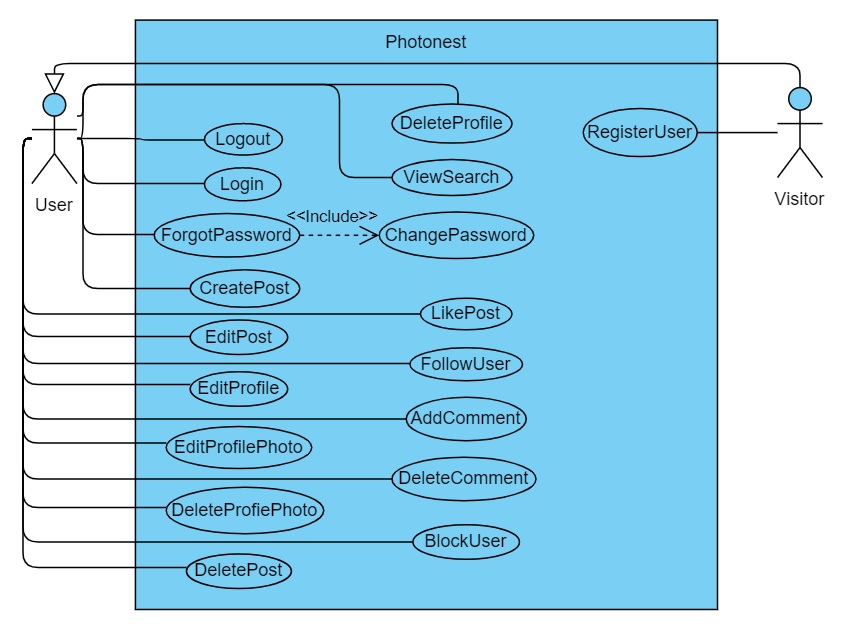
**18)**  *Use case name* **BlockUser***Participating* Initiated by **User**  *actorsFlow of events*

1. The **User** activates the “Block User” function from the profile page by clicking the “Block” option.  
   * 1. **SYSTEM** responds by presenting an informing text and a selection to the **User**.
2. The **User** reads the message and clicks the “Block” option.
   * 1. **SYSTEM** receives the selection, performs the task and presents a success message to the **User**.

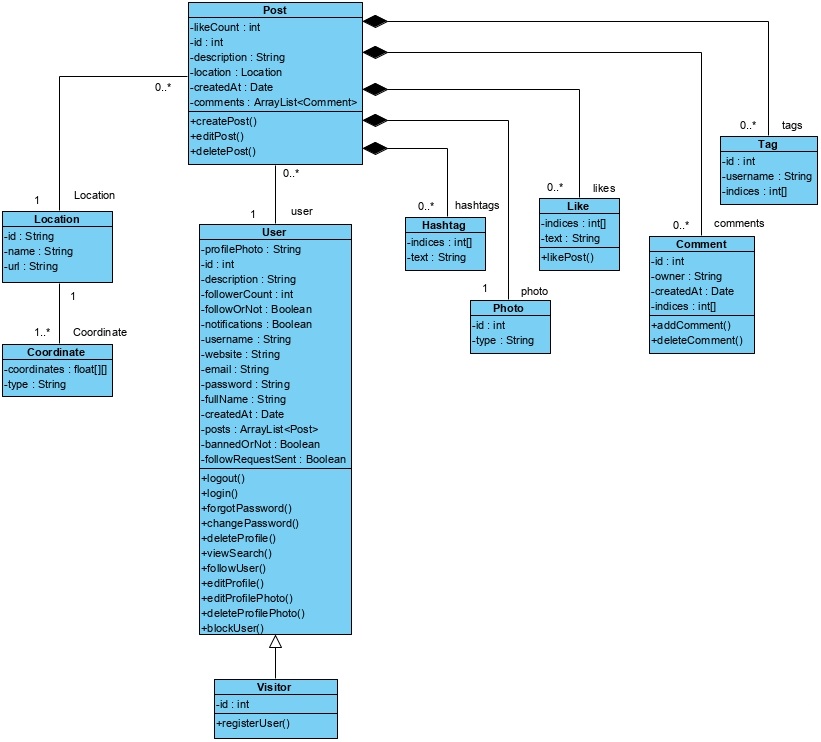
*Entry condition* The **User** has clicked the Block option.*Exit conditions* The **User** has seen a success message.

The **User** has clicked on the Cancel option.   
*Quality* The success message is shown to the **User** within 3 seconds.  *Requirements*

### Use case model



### Object model



# Glossary

**User:** A user type that is already registered to the system. This user type has access to all user functions.  
**Visitor:** A user type that is not registered to the system. This user type can only see the login page and access the Sign-Up function.

**Comment:** A message in the form of text submitted by a user under a post.

**Form:** An object that consists of fields. It is presented when an information is expected from user types.

**Post:** An object that consists of an image, description in the form of text and a location selected from map.

# References

1. Bruegge B. & Dutoit A.H. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. “Instagram” *Instagram*, Facebook Inc, www.instagram.com.
3. “Firebase”, *Firebase*, Google, firebase.google.com.