Photonest

Requirements Specification and Analysis

1.0

08.03.2020

Dilara Ünbay

Nursena Karakulah

Feyzullah Berkay Danış

Özay Ezerceli

Prepared for

SE301 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 2

Usability 2

Reliability 2

Performance 2

Supportability 2

Implementation 2

Interface 2

Packaging 2

Legal 2

3.4. System Models 2

Scenarios 2

Use case model 2

Object model 16

3.5. Project Schedule 16

4. Glossary 16

5. References 16

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

Describes user-level requirements that are not directly related to functionality. This includes usability, reliability, performance, supportability, implementation, and interface, operational, packaging, and legal requirements.

### Usability

### Reliability

### Performance

### Supportability

### Implementation

### Interface

### Packaging

### Legal

## System Models

Describes the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.

### Scenarios

A scenario is an instance of a use case.

### 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 “Ok” 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 immediately re-directed to the login page.  *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 twice 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 updating the followers of the selected profile.

*Entry condition* The **User** has clicked on the Follow button.*Exit conditions* The profile’s followers has been updated.

The **User** has clicked on the Follow button twice.   
*Quality* The follower button changes color upon following.  *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

The analysis object model, depicted with UML class diagrams, includes classes, attributes, and operations. The analysis object model is a visual dictionary of the main concepts visible to the user.

## Project Schedule

Prepare Gannt Chart, and add it to this section.

# Glossary

To establish a clear terminology, developers **identify the participating objects** for each use case. Developers should **identify, name, and describe them** unambiguously and collate them into a glossary.

# References

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.