**Photographer Matching Service**

System and Software Requirements Definition(SSRD)

Team 2

* 2014210011 고나현 rhskgus@korea.ac.kr
* 2013210020 임원준 rhimjun@korea.ac.kr
* 2013210039 이민섭 alexn0@korea.ac.kr
* 2013210081 박진호 onlylove6913@kroea.ac.kr
* 2013210086 김영훈 2013210086@korea.ac.kr
* 2014210059 정희석 poco2889@korea.ac.kr
* 2014210065 이상진 eesj@korea.ac.kr
* 2014210101 김원경 kwk2392@korea.ac.kr
* 2015410005 이정섭 james319@korea.ac.kr

Table of Contents

1 Introduction 1

1.1 Purpose of SSRD 1

1.2 References 1

2 Project Requirements 3

2.1 Budget and Schedule 3

2.2 Development Requirements 3

2.3 Packaging Requirements 4

2.4 Implementation Requirements 4

2.5 Support Requirements 4

3 Capability Requirements 5

3.1 System Definition 5

3.2 System Requirements 5

4 System Interface Requirements 11

4.1 User Interface Requirements 11

4.2 Hardware Interface Requirements 12

4.3 Communications Interface Requirements 12

4.4 Other Software Interface Requirements 13

5 Level of Service Requirements 13

6 Evolution Requirements 17

6.1 Capability Evolution Requirements 17

6.2 Interface Evolution Requirements 17

6.3 Technology Evolution Requirements 17

6.4 Workload Evolution Requirements 18

7 Common Definition Language 19

Appendix 21

A.I Standards Specifications **Error! Bookmark not defined.**

A.II Interface Specifications **Error! Bookmark not defined.**

# Introduction

The SSRD describes the requirements for the proposed Web platform for interior. This information were obtained after both the Easy WinWin negotiation (see EWW report for details) and the presentation of our prototype to the clients (OCD, section 5.0 ). The requirements were categorized into five main topics: project , capability, interface, level of service and evolution. In what follows we present and discuss those requirements within the context of the Web platform for interior.

## **Purpose of SSRD**

* Summarize the purpose and contents of this document with respect to the particular project and people involved
* Avoid generic introductions as much as possible: for instance, you can show how your particular System and Software Requirements Definition meets the completion criteria for the given phase

The main purpose of this document is to specify the functions that the system must develop and the key functions under constraints. The requirements set forth in this document arise from a compromise with all stakeholders. This includes photographers, people who want photos, customers, development teams, and many potential users of systems with varying levels of computer knowledge. This document helps to address the ambiguity of users and developers' needs.

## **References**

# 2.Project Requirements

## 2.1 Budget and Schedule

These are mandated cost and schedule constraints in terms of money and calendar months available for project completion.

|  |  |
| --- | --- |
| Project requirement: | PR-1 |
| Description: | Minimize the total cost of development |
| Measurable: | Total development cost should be zero. |
| Achievable: | Use Freeware and public domain libraries that provided by companies and Korea univ |
| Relevant: | Budget |
| Specific: | Low cost |
| Reference: |  |

|  |  |
| --- | --- |
| Project requirement: | PR-2 |
| Description: | Analyze and design the system in 8 weeks |
| Measurable: | The analysis phase of the system must be completed by December 10, 2018 |
| Achievable: | Perfectly visualize the system and provide the most important features of it |
| Relevant: | Schedule constraints |
| Specific: | This is the time available for designing the system. Time is used as an independent project variable. |
| Reference: |  |

[Consistent with Major Project Constraints OCD 2.4]

## 2.2 Development Requirements

|  |  |
| --- | --- |
| Project requirement: | PR-3 |
| Description: | Open API offers the useful tools and resources to visualize the system. |
| Measurable: | Tools and required information for development can be obtained. |
| Achievable: | Creating accounts to get into Open API. |
| Relevant: | Schedule |
| Specific: | Tools and resources provided by Open API will shorten the total work time. |
| Reference: |  |

## 2.3 Packaging Requirements

None identified at this stage

## 2.4 Implementation Requirements

None identified at this stage

## 2.5 Support Requirements

None identified at this stage

**3. Capability Requirements**

This section describes the proposed system and analyzes its requirements.

**3.1 System Definition**

We plan to make matching platform. Main customers of our system are everyone who wants to be taken photo and everyone who can take photo. To make our platform more elaborate, there exists several requirements from photographers, users and developers. More detailed requirements are explained below.

**3.2 System Requirements**

The requirements for the system defined in the previous section are described below in terms of nominal and off-nominal requirements.

**3.2.2 Win Agreements**

|  |  |
| --- | --- |
| *Agreement* | *AGRE-1* |
| Desired Results | Want to know others’ preferences, photos, as easy as possible. |
| Guidelines: | User inputs keyword which present their desires. The results should consist of the proper output. |
| Resources: | Number of photos, trend of the group of photos. |
| Accountability: | As fast as possible and high relevant results come up first. |
| Consequences: | User can see the relevant result that they want. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-2* |
| Desired Results | Want to prove photographers’ qualification. |
| Guidelines: | Voting/Rating system for qualifying their capability. |
| Resources: | Like, reviews, comments and the quality of his/her photos. |
| Accountability: | There should be no abusing on the system. |
| Consequences: | Users can trust the photographers in the platform and photographers satisfy what they achieve. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-3* |
| Desired Results | Want various spot’s and people’s photos. |
| Guidelines: | Must can be searched if the photos are uploaded. |
| Resources: | Uploaded photos. |
| Accountability: | Photos’ information might be included. |
| Consequences: | It is very comfortable and easy to see a lot of photos. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-4* |
| Desired Results | Want to make photo perfect by retouching. |
| Guidelines: | Give time to photographer to retouch photos. |
| Resources: |  |
| Accountability: | Need enough time. |
| Consequences: | Photo will be beautiful or perfect. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-5* |
| Desired Results | Want to take a photo in this place. |
| Guidelines: | Communicate via our application in real-time for taking a photo. |
| Resources: |  |
| Accountability: | Messaging function is needed. It has to be done in real-time. |
| Consequences: | Photographer can find new spot or earn money. User can get a photo in that place which he/she wants. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-6* |
| Desired Results | Satisfying when see my photos. |
| Guidelines: | After finishing all process of service, photographer has to send photos to the photographer and photographee should present his/her grading. |
| Resources: |  |
| Accountability: | Need easy transmitting method and grading system. |
| Consequences: | Both of users (photographee, photographer) satisfy of the result. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-7* |
| Desired Results | Taking pretty photos. |
| Guidelines: | Using good equipment with nice place and model. |
| Resources: | References of pretty photos. |
| Accountability: |  |
| Consequences: | Getting pretty photos. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-8* |
| Desired Results | Need to check a photographer’s information, reliability, etc. |
| Guidelines: | Rank photographer |
| Resources: | References of pretty photos. |
| Accountability: |  |
| Consequences: | Getting pretty photos. |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-9* |
| Desired Results | Need profile |
| Guidelines: | profile including information about a photographer and photos which the photographee already permitted. |
| Resources: | information of a photographer |
| Accountability: |  |
| Consequences: | User can check someone’s profile who is a photographer |

|  |  |
| --- | --- |
| *Agreement* | *AGRE-10* |
| Desired Results | Need to communicate between a photographer and a photographee |
| Guidelines: | Simple message including each other’s desire of a requirement(s). |
| Resources: |  |
| Accountability: |  |
| Consequences: | Meeting and taking a photo(s) in somewhere. |

**3.2.1 Nominal Requirements**

|  |  |
| --- | --- |
| *Requirement:* | *RQ-1* |
| Title: | Registration |
| Priorirty: | Very High |
| Description: | Anyone can register and use our service. |
| Input(s): | * E-mail account, password, and nickname |
| Source(s): | E-mail account |
| Output(s): | * User Interface Main Homepage |
| Destination(s): | User Interface |
| Precondition(s): | People who want to register must have a valid e-mail account. |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-8 |
| Mainstream Scenario: | People who want to be a member in our platform for taking a photo or being taken, they click a registration button. It shows up a registration page which has simple input forms including E-mail address, password, and nickname. By clicking Log-in button, Terms of Service page is pop-up and after agreement of this, the registration is done. |
| Exception Handling Scenario: | If he/she entered a wrong account information, an alert message will be popped up. |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-2* |
| Title: | Log-in |
| Priority: | Very High |
| Description: | Login function for app access.  There are two types of ID: a photographer, a photographee |
| Input(s): | * ID, Password, 2 ID Types (a photographer or a photographee) |
| Source(s): | User input |
| Output(s): | * User Interface Main homepage |
| Destination(s): |  |
| Precondition(s): | Must be a membership ID |
| Postcondition(s): | Automatically log-in when reconnecting once logged in |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-1 |
| Mainstream Scenario: | When the application runs, a login window appears. Every user (an amateur photographer and ordinary people) enters his or her ID and password to access the application. Besides, he/she should check whether he/she is a photographer or a photographee to identify whether he/she wants to take a photo or to be taken. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-3* |
| Title: | Profile View |
| Priorirty: | High |
| Description: | A photographee(or a photographer) can see specific photographer profile to get his/her information about the photographer. To view photographer’s personal information, he/she should be logged-in for security. |
| Input(s): | * specific photographer’s profile. |
| Source(s): | Photographer’s profile |
| Output(s): | * Profile Page |
| Destination(s): | User Interface |
| Precondition(s): | The photographer’s profile must be written before. |
| Postcondition(s): | A photographer(or a photographee) can see photographer’s profile. |
| Proposed Activity: |  |
| WinWin Agreement(s): |  |
| Mainstream Scenario: | If someone who wants to check photographer’s profile, he/she can click the profile of him/her and see that. After checking profile and thinking that he/she is a proper photographer for oneself at the time, a photographee can contact him/her to do a further process. A further process contains making appointments between a photographer and a photographee. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-4* |
| Title: | Profile Management |
| Priorirty: | Very High |
| Description: | A photographer can manage his/her profile. A photographer can manage his/her own profile which includes his/her information. (ex. location, rating, name, sex, introduction, photos, etc.) Each profile will be shown to photographees, and they will refer a photographer’s profile to make a deal. |
| Input(s): | * Contents of a photographer himself/herself |
| Source(s): | User input |
| Output(s): | * Profile page |
| Destination(s): | User Interface |
| Precondition(s): | Only authorized photographer (already logged-in) can manage his/her profile. |
| Postcondition(s): | Photographees(or photographers if they want to) can see a photographer’s profile. |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-1, AGRE-9 |
| Mainstream Scenario: | After the registration, a photographer should write down his/her profile for showing his/her information to appeal to photographees. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-5* |
| Title: | Searching a photographer |
| Priority: | Very High |
| Description: | A photographee can search by a photographer’s rates or his/her location. In the app, all photographers are rated by photographees’ evaluation. There are 2 types of evaluation; a photographee can click ‘like’ if he/she likes photographer’s photo, and the photographee who is taken by the photographer only rates the photographer from 1 to 5.  On top of that, a photographee can see photographer’s location so that he/she can find out where the photographer usually works at. The photographee could take an advantage of the photographer’s location to pick a nearby photographer. |
| Input(s): | * rating, location |
| Source(s): | Photographer's information |
| Output(s): | * searching results |
| Destination(s): | User Interface |
| Precondition(s): | . Information of photographers and locations should be stored in their profile. |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-2 |
| Mainstream Scenario: | Photographees can line up photographers by their rates when searching. The total number of ‘like’ given to each photographer will be shown. Also, the average of rates will be displayed so that a photographee can find out the photographer he/she wants.  Also, photographees can search photographers by the distance between them. In case photographees want to find out nearby one or they travel to a different city, a very proper photographer will be shown up. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-6* |
| Title: | Searching a photo |
| Priorirty: | High |
| Description: | A photographee can search for uploaded photos by location and a keyword. Also, he/she sorts photos by the number of ‘like’. |
| Input(s): | Keyword(s), location |
| Source(s): | Photo(s) |
| Output(s): | Photos related to keyword(s) or a location |
| Destination(s): | User Interface |
| Precondition(s): | When a photographer uploads a photo, he/she can include a photo-taken location information and keyword(s). |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): |  |
| Mainstream Scenario: | When a photographer uploads a photo, he/she can include a location information such as where a picture is taken at. Key words selected by photographer(s) are also contained in photos. For instance, if a photographee wants to go to Busan for trip, he/she could search ‘Busan’ to look over some photos related to Busan so that he/she can pick the most proper photographer. By searching, you can view pictures of your desired locations, find a photographer, and send a request for a photographer. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-7* |
| Title: | Exchange messages between photographers and photographees |
| Priority: | Very high |
| Description: | One of the key functions of the application. A photographee should be able to contact a photographer (and vice versa) by sending and receiving a message. Each message limits 100 letters and is encrypted for the security. |
| Input(s): | What to say to a photographer or a photographee |
| Source(s): | Messages |
| Output(s): | A confirmation message received from you that the transfer between a photographee and a photographer has been completed. |
| Destination(s): | User Interface |
| Precondition(s): | Service users who only justify their real names are given an authority to send and receive messages. |
| Postcondition(s): | The Matching between photographees and photographers |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-4, AGRE-5, AGRE-10 |
| Mainstream Scenario: | Photographers post photos that they take in the same format as their portfolio, and a photographee finds the photographer they want (RQ-6), then contacts him/her directly with a message to communicate with him/ her. Each message should be encrypted to keep it from outsiders but the photographer and the photographee. SHA-256 will be used. |
| Exception Handling Scenario: | *-* |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-8* |
| Title: | Upload photos |
| Priorirty: | High |
| Description: | A photographer can select his/her own photos to show and appeal to photographees. When uploading photos, a photographer adds the description and can choose specific categories on given categories (e.g. food, scenery, people, animals, wedding, graduation, couples, friendship, etc.) A photographer can specify a location tag where one’s photo is taken. |
| Input(s): | * A photo(s) |
| Source(s): | Photos which a photographer wants to upload |
| Output(s): | * A photo(s) with description |
| Destination(s): | User Interface Main page |
| Precondition(s): | A register, a self-possessed photo(s) |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-1, AGRE-2, AGRE-3 |
| Mainstream Scenario: | A photographer can upload his/her own photos with a brief description in his/her profile. Location and any content can be included. Content contains an explanation of photos. He/she clicks the button to upload icon. Since photos should be his/her own photos, a warning message saying “You only upload your photos” for ownership. When the process of upload is done, he/she can check his/her photos on the User Interface page. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-9* |
| Title: | Edit a Description of Photos |
| Priorirty: | High |
| Description: | A photographer can edit the description of photos which he/she already uploaded before. Location and any content can be edited. Content contains an explanation of photos. However, if photographer wants to change his/her photos uploaded already, he/she cannot change them but should delete and re-upload a new photo. |
| Input(s): | * Edited location or any content |
| Source(s): | user input |
| Output(s): | * Result Page (edited photo(s)) |
| Destination(s): | User Interface |
| Precondition(s): | A photographer can edit a description of his/her own uploaded photos. |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-4 |
| Mainstream Scenario: | If someone who wants to edit his/her uploaded photo, he/she clicks edit button to edit it. He/she can only edit location or content. After that, he/she can save it. If he/she wanted to delete it, he/she would use delete function. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-10* |
| Title: | Delete a Uploaded Photo |
| Priorirty: | High |
| Description: | A photographer can delete photos which he/she wants. |
| Input(s): | * Uploaded Photos |
| Source(s): | Uploaded Photos |
| Output(s): | * Result Page (photo list of a photographer) |
| Destination(s): | User Interface |
| Precondition(s): | A photographer can delete his/her own uploaded photo only. |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): | AGRE-9 |
| Mainstream Scenario: | If a photographer who wants to delete his/her uploaded photo, he/she clicks delete button to delete it. Since this process cannot be undone, the pop-up ‘Do you really want to delete the photo?’ shows up to the photographer. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-11* |
| Title: | Photographer grading |
| Priorirty: | High |
| Description: | Only a photographee taken by the photographer grades the photographer from 1 to 5. Each score from photographees is summed up to get the average, and it will be showed up to other photographees so that they can find a photographer easier and faster. |
| Input(s): | Grades |
| Source(s): | Photo(s) |
| Output(s): | Grading result |
| Destination(s): | User Interface |
| Precondition(s): | Only a photographee taken by the photographer can grade him/her from 1 to 5. |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): |  |
| Mainstream Scenario: | A photographee taken by a desired photographer receives photo(s) and then grades him/her. The total sum of grades that a photographer gets will be shown; how many deals a photographer and a photographee have will be shown. Also, the brief average of the sum will be also provided. This system is cumulative that grades will be constantly changed when a new grade by a photographee comes in. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-12* |
| Title: | Press 'Like' button on a photographer's photo |
| Priorirty: | High |
| Description: | Any photographee can click ‘like’ if he/she likes a photographer’s photo to show his/her fondness; it reflects photographee’s fondness and shows which photographer is favored by a photographee. |
| Input(s): | * 'Like' button |
| Source(s): | uploaded photos |
| Output(s): | * When clicking the 'Like' button, ‘Like’ +1 / click |
| Destination(s): | User Interface |
| Precondition(s): | Any photographee(or a photographer) can click ‘like’ if he/she wishes. |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): |  |
| Mainstream Scenario: | A photographee who wants to use photographer's service can see the photographer's photos in his/her profile. When clicking the ‘like’ button, ‘like’ goes up as much 1. It finally shows how many 'like' for each photo receives. It is related to rating system of the platform in that the number of ‘like’ reflects personal fondness on a photographer. |
| Exception Handling Scenario: |  |

|  |  |
| --- | --- |
| *Requirement:* | *RQ-13* |
| Title: | Comments of photos |
| Priorirty: | Low |
| Description: | You can write comments on photos uploaded. |
| Input(s): | * Comments |
| Source(s): | Photo(s) |
| Output(s): | * The list of comments |
| Destination(s): | To get photographee's fondness |
| Precondition(s): | Only logged-in users can be filled out. |
| Postcondition(s): |  |
| Proposed Activity: |  |
| WinWin Agreement(s): |  |
| Mainstream Scenario: | The photographer uploads the photos. Photographees can view pictures with description and write comments. It reflects photographees’ fondness. |
| Exception Handling Scenario: |  |

**3.2.2 Off-Nominal Requirements**

|  |  |
| --- | --- |
| *Requirement* | *ONRQ-1* |
| Title: | Error logging in |
| Priority: | Very High |
| Description: | If the user enters a wrong username or password, or selects the wrong account domain he/she won’t be granted access to the system. |
| Traceability: |  |

|  |  |
| --- | --- |
| *Requirement* | *ONRQ-2* |
| Title: | Upload size too big |
| Priority: | Medium |
| Description: | In case the user tries to upload a image file whose size is bigger than what the system allows, the image file will not be uploaded and the user should get a notification pop-up message of the problem. |
| Traceability: |  |

|  |  |
| --- | --- |
| *Requirement* | *ONRQ-3* |
| Title: | Disabled location-based service |
| Priority: | Low |
| Description: | If user has turned off the location-based service required for the image’s location tag, the user is prompted to turn on the service. |
| Traceability: |  |

**4. System Interface Requirements**

This section describes how the proposed system will interact with other systems and with its users.

**4.1 User Interface Requirements**

|  |  |
| --- | --- |
| Project requirement | SIR-1 |
| Description | Easy to understand and use |
| Measurable | Any user, no matter what computer experience he/she has, should be able to use the system |
| Achievable | Use a design familiar to most users, and easy access to every function |
| Relevant | Ease of use will help increase the acceptability and usability of the system |
| Reference |  |

|  |  |
| --- | --- |
| Project requirement | SIR-2 |
| Description | Easy to search and organize photos |
| Measurable | Intuitive way of uploading photos for photographers and searching photos for users |
| Achievable | Use a design familiar to most users, and functions that can be done easily |
| Relevant | Ease of use will help increase the acceptability and usability of the system |
| Reference |  |

The project will be based with the Graphical User Interface which will have the following characteristics:

- Screen with multiple buttons that located at bottom lead the users to other screens.

- Buttons for most commonly used functions actions.

- Text boxes for user input and a text area when composing/replying/forwarding a message

* **Login screen**This will be the main entrance to the system. It will have two text boxes for typing the username and password, a pull down menu for selecting the account type, a OK and a Clear button.s

three buttons for “login with SNS account (kakao, naver, facebook)”

* **Initial Home Screen**This will be the main entrance to the system for clients. It shows your basic personal information and recommends you high ranked photo or photographer
* **Search Screen**There are two top categories in this page which are Photo Search and Photographer Search. Photo Search has subcategories for more specific conditions. (ex. Portrait, Landscape, Wedding etc.) By selecting Photographer Search, you can either search certain photographer’s ID or select new photographer with certain time and location you want.
* **Message Screen**  
  This page is a list of users whom you have shared messages with. By clicking the user’s ID, you can check the past messages you have sent and received. Also, you can send new messages within the same page.
* **History Screen**

This is the screen displays history of photographers that the user met before, and thumbnail of photos.

Photographers' ID will be hyperlinked to his/her own page.

Each thumbnail will be hyperlinked to user's photos at user's own page

* **MY page**

This is the screen displays user's(includes photographers) own page.

It shows user's ID,profile picture,and short intorduction on the top.

There will be buttons for send message.

Especially for photographers, it displays photographer's rate by pentagon icon.

For photographers, his/her portfolio will be displayed at the middle of the screen.

And for models, the photos taken by other photographers will be displayed with the photographer's ID.

Other users can write the review or comments on the photos.

And the models can re-rate the photo unlimitedly.

* **Rating screen**

When the model gets the photos from the photographer, this screen will pop up.

There are four evaluation features: Satisfaction of photo, Satisfaction of edition, Communication, Cost

**4.2 Hardware Interface Requirements**

Devices that are available to access mobile application.

**4.3 Communications Interface Requirements**

\*Not identified at this stage

**4.4 Other Software Interface Requirements**

|  |  |
| --- | --- |
| Project requirement | SIR-3 |
| Description | Interface with DBMS server |
| Measurable | Send queries to retrieve the desired results from the database |
| Achievable | Sending and receiving data should be succeed, especially, make sure those user input data store in our database and server. |
| Relevant | Data transmission and reservation |
| Reference |  |

**5. Level of Service Requirements**

In this section we discuss how well the system should perform.

|  |  |
| --- | --- |
| *Project Requirement:* | *LOR-1* |
| Description: | Meet the current security standard. |
| Measurable: | Unauthorized users who isn’t logged-in shouldn’t be able to gain access to the system or the user’s account. Each user must have access to his/her data only. |
| Relevant: | Security |
| Specific: | Security is a important part. An insecure system won’t be acceptable. |
| Reference: |  |

|  |  |
| --- | --- |
| *Project Requirement:* | *LOR-2* |
| Description: | Sufficient storage of system. |
| Measurable: | It isn’t allowed uploading large size image, and file format must be image file. |
| Relevant: | Stability . |
| Specific: | Most users can use system without worrying about data loss. |
| Reference: |  |

|  |  |
| --- | --- |
| *Project Requirement:* | *LOR-3* |
| Description: | Easy approach for users. |
| Measurable: | Users should be able to use the system regardless of their computer knowledge. |
| Relevant: | Ease of use. |
| Specific: | Most users will be able to start using the system fast through intuitive user interfaces. |
| Reference: |  |

|  |  |
| --- | --- |
| *Project Requirement:* | *LOR-4* |
| Description: | Quick Reaction |
| Measurable: | Most of functions of system, including search/filtering function, should be fast. Result is reflected instantly. |
| Relevant: | Performance. |
| Specific: | Most search results pop-up with in 3 seconds. |
| Reference: |  |

|  |  |
| --- | --- |
| *Project Requirement:* | *LOR-5* |
| Description: | Protection of law |
| Measurable: | There is no legal case related to using system. |
| Relevant: | Reliability. |
| Specific: | The system should be designed to prevent legal accidents such as portrait rights, copyright and financial accident |
| Reference: |  |

# Evolution Requirements

## Capability Evolution Requirements

|  |  |
| --- | --- |
| Project requirement: | CR-1 |
| Description: | Matching an amateur photographer to a model |
| Measurable: | Ability to contact a model for an amateur photographer with high speed and low error rate. Also, exposure of model’s personal data should be minimized. |
| Achievable: | Database capabilities would need to be great to handle the request of a photographer. It would require a user to verify his/her name and social security number. |
| Relevant: | Ease of use. |
| Specific: | Only model can view the photographer’s profile, but a photographer can’t contact to a model before a model contacts him/her. Thus, it would be useful for a photographer to add the way to contact a model. |
| Reference: | - |

|  |  |
| --- | --- |
| Project requirement: | CR-2 |
| Description: | Hashtagging |
| Measurable: | 1. Adding the function of hashtagging with photos 2. Every user can look for his/her concern (certain category, keyword, etc.) by hashtagging |
| Achievable: | It would require indexing functionality to index by hashtag; additionally, the further setting needs to be added. |
| Relevant: | Ease of use |
| Specific: | Tagging a hashtag on photos; similar to that of Instagram where you would be able to see hashtags on photos. . |
| Reference: | - |

## Interface Evolution Requirements

|  |  |
| --- | --- |
| Project requirement: | CR-3 |
| Description: | Provide ads |
| Measurable: | Pop-up of ads will be displayed randomly. |
| Achievable: | Create a graphical tool that contains information of each company so that it appears to users randomly but fairly |
| Relevant: | Ease of use |
| Specific: | To balance the revenue model to provide better services for users |
| Reference: |  |

|  |  |
| --- | --- |
| Project requirement: | CR-4 |
| Description: | Provide administration interface |
| Measurable: | The ISD administrator should be able to administer the system using a graphical tool, instead of editing text files and issuing commands at the OS level. |
| Achievable: | Create a graphical tool that will translate user actions (menu selections, mouse clicks etc.) into OS commands. |
| Relevant: | Ease of use |
| Specific: | Will help in easier administration of the system. |
| Reference: |  |

## Technology Evolution Requirements

|  |  |
| --- | --- |
| Project requirement: | CR-5 |
| Description: | Application can be improved to work well on other operating systems and platforms. |
| Measurable: | User can use the application on non-android based system. |
| Achievable: | Create an interface for other operating systems such as IOS, BlackBerry OS. |
| Relevant: | System evolution |
| Specific: | It would contribute to ease of use, and anytime-anywhere-access by any type of devices. |
| Reference: | - |

## Workload Evolution Requirements

|  |  |
| --- | --- |
| Project requirement: | CR-6 |
| Description: | Workload change from developing application and user interface towards maintaining system and creating profit via advertisements. |
| Measurable: | The workload shifting will be kept up by March 2019. |
| Achievable: | Nice planning and strict timeline are needed. |
| Relevant: | The challenge of this project is not for the development but for promotion of application. |
| Specific: | Our primary goal is to complete developing application until December 2018, and changing workload of maintaining and marketing until March 2019. |
| Reference: | - |

## Level of Service Evolution Requirements **None identified at this time**

# Common Definition Language

**Naver map API**

A Naver map service draws visualized maps on website or mobile apps. It consists of a map JavaScript API and Static Map API.

**Java**

Java is a language developed by Sun Microsystems. Programs written in Java can run on many platforms without any changes or re-compilation.

**Servlet**

Java programs that runs on the Server-side. Can replace CGI scripts and requires fewer resources.

**Instagram API**

In Instagram API, we use the function of hashtag.

**Photographer**

One who is taking a photograph

**Photographee**

One who is photographed.

**Retouch**

Change original photo to be an edited photo by controling some attributes like contrast, sharpness, tone, etc.

**Keyword**

Some specific words for searching

**Review**

Comment or feedback about a uploaded photo

**Grading**

To give a score in range 1 to 5

**Rating**

Such a score or something that represent that photographer's rank in the platform.

**Like**

A feature that allows users to show their support for specific comments or photos.

**Profile**

Photographer's information such as name, sex, location, photos, etc.

**Platform**

Such an operating system, software, etc. having a standard design for use with compatible programs, applications, etc.

**Evaluation**

Grading

**User**

Representing both of photographer and photographee

**Portfolio**

A group of photos of a photographer which already get copyrights from photogrphees.

**User Interface**

General user interface(for mobile)

**Deal**

To make a complete appointment for taking a photo.