Photographer Matching Service

System and Software Requirements Definition(SSRD)

Team 2

- 2014210011 고나현 2013210020 임원준
- 2013210039 이민섭 2013210081 박진호
- 2013210086 김영훈 2014210065 이상진
- 2014210101 김원경 2014210059 정희석
- 2015410005 이정섭

Table of Contents

1 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.

1.1 Purpose of SSRD

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.

1.2 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 fifteen weeks
Measurable:	The analysis phase of the system must be completed by 12/10/2018
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:	Interview various amateur photographers and ordinary people
Measurable:	Interview and survey with Google Form
Achievable:	Set the date, meeting place and make a google form to gather more views

Relevant:	Schedule
Specific:	We can find missing parts and replace them or develop them.
Reference:	

Project requirement:	PR-4
Description:	Open API offers the useful tools and resources to visualize the system.
Measurable:	Tools and information needed for development can be obtained.
Achievable:	Creating accounts to get into Open API.
Relevant:	Schedule
Specific:	Resources we get will shorten 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.

3.2.1 Nominal Requirements

Requirement:	*RQ-1*
Title:	Log-in
Priorirty:	Very High
Description:	Login function for app access. There are two types of ID: the Photographer and the general user.
Input(s):	• ID, Password, ID Types
Source(s):	User input
Output(s):	• UI 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):	
Mainstream Scenario:	When the application is run, a login window appears and the user/photographer enters his or her ID and password to access the application.
Exception Handling Scenario:	

Requirement:	*RQ-2*
Title:	Searching Photographer
Priorirty:	Very High
Description:	Users can search by photographers' ability and by their location.
Input(s):	• ability, location
Source(s):	Photographers' Information
Output(s):	• searching results
Destination(s):	User interface
Precondition(s):	. Information on writers' abilities and locations should be stored.
Postcondition(s):	
Proposed Activity:	
WinWin Agreement(s):	
Mainstream Scenario:	Users can search photographers by keyword for their ability or location. If there are photographers matching the search term, the list is shown to the user. If there is no matching photographer, display the appropriate message.
Exception Handling Scenario:	

Requirement:	*RQ-3*
Title:	Ability to exchange messages between amateur photographers and models
Priorirty:	Very high
Description:	One of the key functions of the application. Ability to communicate with each other (send, receive) to photograph with the photographer the model wants
Input(s):	What you want to say to the photographer or model
Source(s):	messages
Output(s):	A confirmation message or a note received from you that the transfer has been completed.
Destination(s):	User interface
Precondition(s):	Only users with real name authentication must send and receive messages
Postcondition(s):	The Matching of Photographic Models with Amateur Photographers
Proposed Activity:	-
WinWin Agreement(s):	-
Mainstream Scenario:	Amateur photographers post photos of themselves in the same format as their portfolio, and the photo model finds the photographer they want (RQ-6), then contacts him directly with a note to communicate with him or her.
Exception Handling Scenario:	-

Requirement:	*RQ-4*
Title:	Upload photos
Priorirty:	High
Description:	Key to the system and the ability to tell what the system is. The function to share a brief description and tell you what kind of picture is by selecting a previously presented category(food, scenery, people, animals, wedding, friendship photos, etc) Specify the location tag, what kind of picture it is, and can be posted with the description
Input(s):	• photo & photo type
Source(s):	Photos intended for user to upload
Output(s):	• a photo post + information about which pictures are
Destination(s):	UI Main page
Precondition(s):	A registered user, a self-possessed photograph
Postcondition(s):	Whether the photos you uploaded are reliable
Proposed Activity:	OCD 4.5.1 PA-I.02 (?)
WinWin Agreement(s):	Win-AG-29(?)
Mainstream Scenario:	Users can publish their photos with a brief description by making it easy for them to select the category platform(Information of what kind of picture it is) they want to publish. You can check the picture on the UI Main Page.
Exception Handling	*ONRQ-1(?)*

Scenario:	

Requirement:	*RQ-5*
Title:	Photographer Rating
Priorirty:	High
Description:	rank photographers. can understand the photographer's personality. Only the person who receives the photo can evaluate it because it matches the photographers. Evaluation item-photo performance, correction, matching count, friendship, etc. The evaluations accumulate
Input(s):	Evaluation item
Source(s):	User input
Output(s):	Photographer’s grade-pentagon shape
Destination(s):	UI
Precondition(s):	Only matched photo owners can evaluate. (Assessment 1 for Matching 1)
Postcondition(s):	Once the assessment has been performed, the assessment can be modified.
Proposed Activity:	
WinWin Agreement(s):	
Mainstream Scenario:	A customer who matches a desired photographer receives a photo and then evaluates it. Evaluation items include satisfaction with photos, post-calibration, etc. The assessment can be revised later. The assessment is cumulative, and the photographer's ratings are fixed and the photographer 's attitude can be understood. (pentagon)
Exception Handling Scenario:	

Requirement:	*RQ-6*
Title:	Picture-searching by condition
Priorirty:	High
Description:	One of the main functions of the system. Users can search for uploaded photos by region, feeling (bright, dark, sunny, etc.) and recommendation number.
Input(s):	search keyword, condition selected by filters.
Source(s):	pictures uploaded by authors
Output(s):	Photos suitable for the filter and keywords
Destination(s):	User interface
Precondition(s):	Each picture should have basic information such as key word settings (ex. Hashtag) and location and date.
Postcondition(s):	
Proposed Activity:	

WinWin Agreement(s):	
Mainstream Scenario:	Users can use the search function to inquire about the atmosphere or structure of photos that they want to take. Through a search, you can view pictures of your favorite destinations and find a photographer based on them and send a request for a photo.
Exception Handling Scenario:	

Requirement:	*RQ-7*
Title:	The ability to express oneself by pressing 'Like' on a photographer's picture.
Priorirty:	High
Description:	It is the most basic system for grading photographers. When the author uploads a photo, the customer (model) who wants to use the service shows his or her fondness for the photo through "Like."
Input(s):	• 'Like' icon
Source(s):	uploaded photos
Output(s):	• When you click the 'Like' icon, click 'Fine' +1 / click
Destination(s):	User interface
Precondition(s):	
Postcondition(s):	
Proposed Activity:	
WinWin Agreement(s):	
Mainstream Scenario:	Customers who want to use photographer's photography service can see the photographer's photos on the app. Next to the photos are how much 'like' each picture received, and all users can check the number of 'like' pictures. All users can also press 'like' for each user. Photographers are graded based on 'like' numbers.
Exception Handling Scenario:	

Requirement:	*RQ-8*
Title:	Portfolio management
Priorirty:	High
Description:	Individual photographers can upload their own portfolio and store and disclose their own photos for their career management. Photos stored in a portfolio can be made public/public and registered portfolios can be accessed and viewed by others.
Input(s):	• Explanation of photos you uploaded
Source(s):	Previously uploaded by yourself
Output(s):	• Personal Portfolio Album
Destination(s):	User Interface
Precondition(s):	The photos you want to register must be uploaded in advance.

Postcondition(s):	You can get off later.
Proposed Activity:	
WinWin Agreement(s):	
Mainstream Scenario:	Photographs of customers should be taken, approved, and then placed on the platform. If any of the photos on the platform want to be put into a portfolio, the photos will be registered in the portfolio and a brief description can be attached.
Exception Handling Scenario:	*Registration of a portfolio without a photo will be rejected.*

Requirement:	*RQ-9*
Title:	Review of postings
Priorirty:	low
Description:	You can write reviews or comments on posts uploaded.
Input(s):	• Review
Source(s):	User input
Output(s):	• User name, simple review
Destination(s):	To get people's real reaction and empathy.
Precondition(s):	Only logged-in users can be filled out.
Postcondition(s):	
Proposed Activity:	
WinWin Agreement(s):	
Mainstream Scenario:	The photographer uploads the pictures. Users can view pictures and write comments with various responses. The user photographed is marked with a subject.
Exception Handling Scenario:	

3.2.2 Win Agreements

Agreement	*AGR1*
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	*AGR2*
Desired Results	Want to prove photographers' qualification.
Guidelines:	Voting/Rating system for qualifying their capability.
Resources:	Thumbs up, 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	*AGR3*
Desired Results	Want various spot's and people's photos.
Guidelines:	Must can be searched if the pictures are uploaded.
Resources:	Uploaded pictures.
Accountability:	Pictures' information might be included.
Consequences:	It is very comfortable and easy to see a lot of photos.

Agreement	*AGR4*
Desired Results	Want to make picture perfect by retouching.
Guidelines:	Give time to photographer to retouch photos.
Resources:	
Accountability:	Need enough time.
Consequences:	Photo will be beautiful or perfect.

Agreement	*AGR5*
Desired Results	Want to take a picture 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	*AGR6*
Desired Results	Satisfying when see my photos.
Guidelines:	After finishing all process of service, photographer has to send photos to the owner and owner should present his/her feedback.
Resources:	
Accountability:	Need easy transmitting method and feedback system.
Consequences:	Both of users (customer, photographer) satisfy of the result.

Agreement	*AGR7*
Desired Results	Taking pretty photos.
Guidelines:	Using good equipment with nice place and model.
Resources:	References of pretty photos.
Accountability:	
Consequences:	Getting pretty photos.

3.2.2 Off-Nominal Requirements

Requirement	*ONRO-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:	

6 Evolution Requirements

6.1 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:	-

6.2 Interface Evolution Requirements

Project	CR-3

requirement:	
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:	

6.3 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:	-

6.4 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:	-

6.5 Level of Service Evolution Requirements

None identified at this time

7 Common Definition Language

MQTT

(Message Queuing Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922) publish-subscribe-based messaging protocol. It works on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker.

SMTP

Short for Simple Mail Transfer Protocol, a protocol for sending email messages between servers. Most email systems that send mail over the Internet use SMTP to send messages from one server to another. In addition, SMTP is generally used to send messages from a mail client to a mail server.

LDAP

Short for Lighweight Directory Access Protocol, a set of protocols for accessing information directories. LDAP is based on the standards contained within the X.500 standard, but is significantly simpler. And unlike X.500, LDAP supports TCP/IP, which is necessary for any type of Internet access.

ASP

Active Server Pages (ASP) is a technology developed by Microsoft. It allows the programmer to create HTML pages containing code in a scripting language that can be executed on the server or on the client.

CGI

Common Gateway Interface (CGI) is a way to execute programs on a server and send their result to a Web browser. For each request, a CGI program runs another instance of itself and that’s why they may consume a great amount of system resources.

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.