**Project Specification**

Functionality Description

**Summary**

This document aims to demonstrate the functionalities of web application developed by “42”, team 197. The features and services provided by our product will be illustrated in details.

Basically, our product will contain all of the functionalities, listed as different modules, as below:

* user administration
* post administration
* comments administration
* short video administration
* user recommendation system
* post recommendation system
* post ranking system
* text-based searching system
* image-based searching system
* Google map location integration (using Google map APIs)
* Facebook integration (using Facebook APIs)
* Instagram integration (using Instagram APIs)
* web crawler integration (using Scrapy)

**Demonstration**

This section discusses the services provided by each module of our web application.

1. User administration

This module mainly designed for all features and requirements for user administration in our product. Our product is a C2C e-commerce platform. Thus, the “user” refers to the individual end-user of our web application.

* Non-registered user can register a new account. To register a new account, users should at least provide a username, password and email.
* Each username and email can only be registered for one account.
* Users have to verify their email before they activate their account.
* While registering, user can specify other personal information, such as gender, birthday, location and phone number, etc. Those fields are not required.
* Registered user can modify their password by “Change Password”.
* Registered user can modify their email by “Change Email”. If the email is changed, the old email can be used to register a new account while the new email can not.
* Registered user can modify their personal information by “Edit Profile”.
* Registered user will have a default profile picture, and they can change it by “Upload New Picture”.
* Registered user has the authentication to the services provided by our web, including create a new post, view the details of a post, follow other users, use recommendation system, use location system and so on.
* Registered user can link their account to their Facebook account under the same email, and this help them to user the Facebook integration service.
* Logged-in user can follow/unfollow other users and view the posts and profiles of other users.
* Non-registered user can also log in with their Facebook/ Instagram account, with which they are allowed to use the integration services.
* Non-registered or non-logged-in users are only allowed to see the global stream page. They don’t have the authentication to use any other services provided by our web application.

2. Post administration

This module describes the features related to the “post” in our application. A “post” refers to the contents shared by one user to all other users.

* A post must contain a post title, at least one picture, a short description (up to 300 characters), post category, post date and post location. The relationship between the user and the posts is “one to many”.
* A created post have a two counting numbers, recording how many users like or dislike this post.
* All web site users, including non-logged-in users, can visit the global stream of the posts, named as “global posts”. When the user visits the website, the global stream page will be displayed directly.
* The “global posts” will be sorted automatically in reverse-chronological order. However, user can choose to sort the global posts based on the score, post title, the category of the post or the location of the post.
* Only the logged-in user can display the posts from who she/he followed, named “follower posts”.
* Logged-in user can view her/his own posts. And they can delete the existing posts.
* As an extension, the user can upload one extra short video for one post.

3. Comments administration

This module is tightly related to the post administration module.

* Comments administration aims to manage the comments and replies of one post. A comment is towards all users, and a reply is towards one specific comment.
* A post also contains multiple comments and multiple replies.
* A comment will have multiple replies. Usually a user can create a comment to a post, then all follow up discuss under this comment are treated as the replies.
* The comments and replies can happen among any registered users.
* The date when the comment or reply is generated, as well as the user who create this comment or reply, will be recorded.
* A reply have to be indicated to specific user, otherwise it will be treated as comment.

4. Short video administration

This module is an extension of the post module. Besides the pictures and other basic information a post can have, users can also upload a short video to their posts.

* A short video could be at most 1 minutes long, 50 MB big.
* When a new post is created, a user must specify either to upload images or a short videos. If a user choose to upload a short video, then no need to upload images.

5. User recommendation system

This module is used for recommending the other users for current user. The recommendation is based on Cosine Similarity\*.

* Cosine Similarity: in our application, we see the ratio of “the number of the common users followed by user A and user B” to “the product of the number of the total users followed by user A and user B” is the similarity between user A and user B. \*More specifically, if user A follows {C, D, E} and user B follows {D, E, F, G}, then the similarity between user A and user B is: |2|/|3|\*|4| = 1/12.
* During the phase of development, when the user log in, the client side will send a request to the server, asking for the recommendation.
* The recommendation system will automatically run in the backend server. When it gets the results from the calculation, it will return to top 20 recommended users to server.
* The recommended list will be stored in a data model, aligned with the recommendation date.
* Running a recommendation system is time-consuming especially when the user base is big. This is why we design to store the temporary recommendation result and reserve it for future usage.
* Each time there will be 5 recommended users sending back to the main page, and the recommended user list will be appended to the left of the main page.

6. Post recommendation system

This module is used for recommending other posts that is related to the current post the user are viewing.

* Similar to user recommendation system, when the user click on one posts, the client side will send a request to the server, asking for the recommendation.
* The recommending posts is from the users who the current user follows, or the users who is recommended to the current users.
* To calculate the recommendation, we first get the users list described above. Then we randomly pick up 5 users from the user lists.
* From this list, we get all the posts from those users, then filter the posts in the same category.
* Then we randomly select the 50 posts, which is mostly recent as well as mostly popular.
* Each time, the recommendation system will return 5 recommended posts to the client sides.

7. Post ranking system

This module is used for an overall ranking of all the posts in one specific category. Logged in users can see the ranking result from a link in the main page.

* The post ranking is based on the category of the post.
* The ranking system will rank the posts in current category based on the following formula: number of like/ (number of like + number of dislike).

8. Text-based searching system

This module enable to logged-in user to search the contents that she/he might be interested in.

* The search entry is located in the top of the main page.
* User can input a string and send the search request to the server.
* The server will run the search by matching the input string to the title or the description of the posts.
* The user can also specific the searching scope. For example, the user can specify only to search the post titles, which will result in a much faster searching. The search level contains: by title, by content, by commodity.
* When the search result is found, return to the client and refresh the certain part of the html in the main page.

9. Image-based searching system

This module enable to logged-in user to search the similar pictures to the provided picture.

* We will call external Google image searching APIs to complete this module.
* By given a picture, the system will be able to return a set of similar picture of the given one.
* The return set is a set of the posts that contains the picture, instead of the picture itself.

10. Facebook integration (using Facebook APIs)

This module is for the user who attempts to log-in with their Facebook accounts.

* Unregistered user can log in our web application using the same account for Facebook.
* When the user is logged in with Facebook account, she/he can share her/his new posts to Facebook.

11. Instagram integration (using Instagram APIs)

This module is for the user who attempts to log-in with their Instagram accounts.

* Unregistered user can log in our web application using the same account for Instagram.
* When the user is logged in with Instagram account, she/he can share her/his new posts to Instagram.

12. Google map location integration (using Google map APIs)

This module is designed for user to share their position when they create a new posts.

* When the user is creating a new post, the user can specify her/his current location by calling the Google Map APIs.
* The location of the post will be displayed when the post is displayed.
* From the main page there links to a Google Map, which can show the geographical distribution of the global posts.
* In this Google Map there is also a search bar. Users can filtered out the posts by giving a string.

13. Web crawler integration (using Scrapy)

This module is a supplemental for the testing data for our projects. It will not provided in the final project.

* Our team decide to implement a separate component, a web crawler, our project to retrieve the data from some other C2C platform, only for the purpose of testing.
* We will use Scrapy, an open source web crawling framework in python, to extract the demo posts content from the websites.
* After gathering the data, we will load them into our database for testing.