### Quality Attributes

#### **a. Usability**

* The system has a friendly and good-looking GUI which makes it easy for users of all ages to access the product, helps users feel comfortable when using the application.
* The interface of main functions is learned, inherited and developed from previous successful systems such as Facebook, Instagram, ... This gives users a sense of familiarity when using the software and helps users access the functions of the system more effectively.

#### **b. Reliability**

* All data entered by the user is validated before sending it to the server
* When guest register a new account, the email must be valided by OTP.
* The system complies with business rules.
* The display of personal information such as date of birth, measurements, ... will be decided by the owner.

#### **c. Performance**

* Two functions (Message and Notification) are real-time updated . When a new message or notification appears, users will receive it almost immediately

#### **d. Security**

* Each email is only validated for one account. User can use Google or Facebook account to login as long as it has the same email as the one used to authenticate the account.
* The system must use HTTPS to encrypt the information transferred between clients and servers.
* The user authorization table is as follow:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | | **Super Admin** | **Staff** | **Users** |
| Login | | ✔ | ✔ | ✔ |
| Logout | | ✔ | ✔ | ✔ |
| Manage staffs | | ✔ |  |  |
| Manage accounts | | ✔ | ✔ |  |
| Manage posts | Report posts |  |  | ✔ |
| Ban posts | ✔ | ✔ |  |
| Other managements |  |  | ✔ |
| Manage users | | ✔ | ✔ |  |
| Manage notifications | |  |  | ✔ |
| Manage messages | |  |  | ✔ |
| Manage vorchers | | ✔ | ✔ |  |
| Manage configure email | | ✔ |  |  |
| Manage react of posts | |  | ✔ | ✔ |
| Manage data of recommendation system | | ✔ |  |  |

#### **e. Maintainability**

This system uses RESTFUL API structure with ASP.NET Core, with the following advantages to maintain:

* We can easily maintain our application because of separation of concern.
* At the same time we can split many developers' work at a time. It will not affect one developer work to another developer work.
* It supports TTD (test-driven development). We can create an application with unit test. We can write won test case.

#### **f. Portability**

* System is created by using API and React Native which can be developed in the future to supports responsiveness for all screens: mobiles, desktops, tablets, laptops (For candidate view)

#### **g. Dependability**

N/A

#### **h. Supportability**

N/A

#### **i. Design Constraints**

**N/A**

#### **k. Support Documents**

**N/A**

#### **l. Purchased Components**

**N/A**