**SOFTWARE REQUIREMENTS SPECIFICATION**

**PROJECT: [*InstaShop* Mobile Application Social Network for Shopaholic and Trading Enthusiasts]**

Note: This is a simplified version of the IEEE specification standard. It is intended only for educational purposes.**Document Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Document ID | Version | Author/Editor | Date |
| Insta\_shop\_req | **1.0** | Batool Hamawi  Fatimah Al-Mashi | 4/10/2016 |
| Insta\_shop\_req | 2.0 | Batool Hamawi  Fatimah Al-Mashi |  |
| Insta\_shop\_req | 3.0 | Batool Hamawi  Fatimah Al-Mashi |  |

**TABLE OF CONTENTS**

[**1. Description 4**](#_Toc450077422)

[**1.1. Business perspective 4**](#_Toc450077423)

[**1.2. Business Product/Service functions 5**](#_Toc450077424)

[**1.3. User classes and characteristics 5**](#_Toc450077425)

[**2. System Functions 6**](#_Toc450077426)

[**2.1. Shopper Functions 6**](#_Toc450077427)

[**2.2. Trader Functions 11**](#_Toc450077428)

[**2.3 Functional requirements 13**](#_Toc450077429)

[**3. Nonfunctional Requirements 17**](#_Toc450077430)

[**3.1. Performance requirements 17**](#_Toc450077431)

[**3.2. Safety requirements 17**](#_Toc450077432)

[**3.3. Security requirements 18**](#_Toc450077433)

[**3.4. Software quality attributes 18**](#_Toc450077434)

[**3.5. Other Operational requirements 19**](#_Toc450077435)

# Description

## Business perspective

The *InstaShop* mobile application is a Social Network for shopaholic and trading enthusiasts, and it is a new mobile applicati­­on which used to connect entrepreneurs and shopping enthusiasts with each other to allow them shopping and trading. The application allows the traders to sell products, manage and organize their selling processes, and allows other users to browse and buy products, then allows all users to share their social commercial activities with others.

The *InstaShop* is employed Freemium[[1]](#footnote-1) as business model, by delivering our app to customers free of charge, we can achieve an extensive distribution for our app. Adopting this freemium business model is a way of adapting to the changing market and the conditions of production. We can generate revenue with the Freemium model in two ways:

* **In-App Purchases:** Once our customers are using our *InstaShop* app, we provide them with chances to expand its functionality by purchasing content packages within the app. There will be a monthly fee for the paid packages (with an option to pay yearly upfront for a discount). For example: manage product packages will be priced (i.e. $11.99, $24.99, etc.).
* **In-App Advertising**. Selling in-app advertising will be a good way to generate revenue.

In the world of commerce applications, there are many sites and applications that help users to start their business using their devices and help them to find their needs. some similar applications to *InstaShop* to compare them with it, that is providing an organized commercial environments and those which rely on social interactive commerce between users. The competitor’s applications are listed below:

* **The Amazon App**

The most famous application for selling and buying digital and physical products through millions of shops in amazon catalogues and allow users to manage their orders from anywhere.

* **Modah**

An Arabic mobile application with an easy user-friendly interface that allows users to start their own shops and upload their own products. Also, allow users to browse different products and buy from anywhere.

* **Instagram**

Instagram is a regular social media for posting and sharing general and different videos/ photos between users. However, Instagram used as first target competitor for *InstaShop* application, because it is being used by the target users of m-commerce applications and they adapt Instagram to fit their needs. That is due to its features, because it is totally based on the idea of interacting user-activities, and sharing moments between users. Finally, this is the project’s basic target competitor and its users are the project’s target users.

## Business Product/Service functions

The InstaShop mobile application is a Social Network for shopaholic and trading enthusiasts, which acts as an interactive, organized, and enjoyable environment of user-generated content which allows users to share their shopping and trading activities.

**Goals:** It aims to combine the features of social networking with commerce functionalities in one application and to overcome the limitations of using social networks for trading purposes.

**Services:** In InstaShop, users can open their own shops, organize their products in categories, receive and manage orders, retrieve order history, and assist their reach to larger numbers of customers. Where it is realized that consumers trust recommendations of friends and family as well as word of mouth from trusted sources far more than any advertising message.

We benefit from InstaShop’s social environment to obtain trusted peer recommendation and increase profit opportunities. Customers can easily find their desired products by seeking friend's and follower's help, request the products online, and track their orders.

## User classes and characteristics

All people who have Android smart phone can use the application in general. However, the target users for this application are those entrepreneurs who own small businesses and may have no private brands or trademarks, or they could be simply selling homemade products.

However, and for advertising their products purpose, those entrepreneurs use non-commerce applications for commercial purposes to get more customers, and to promote to their own products.

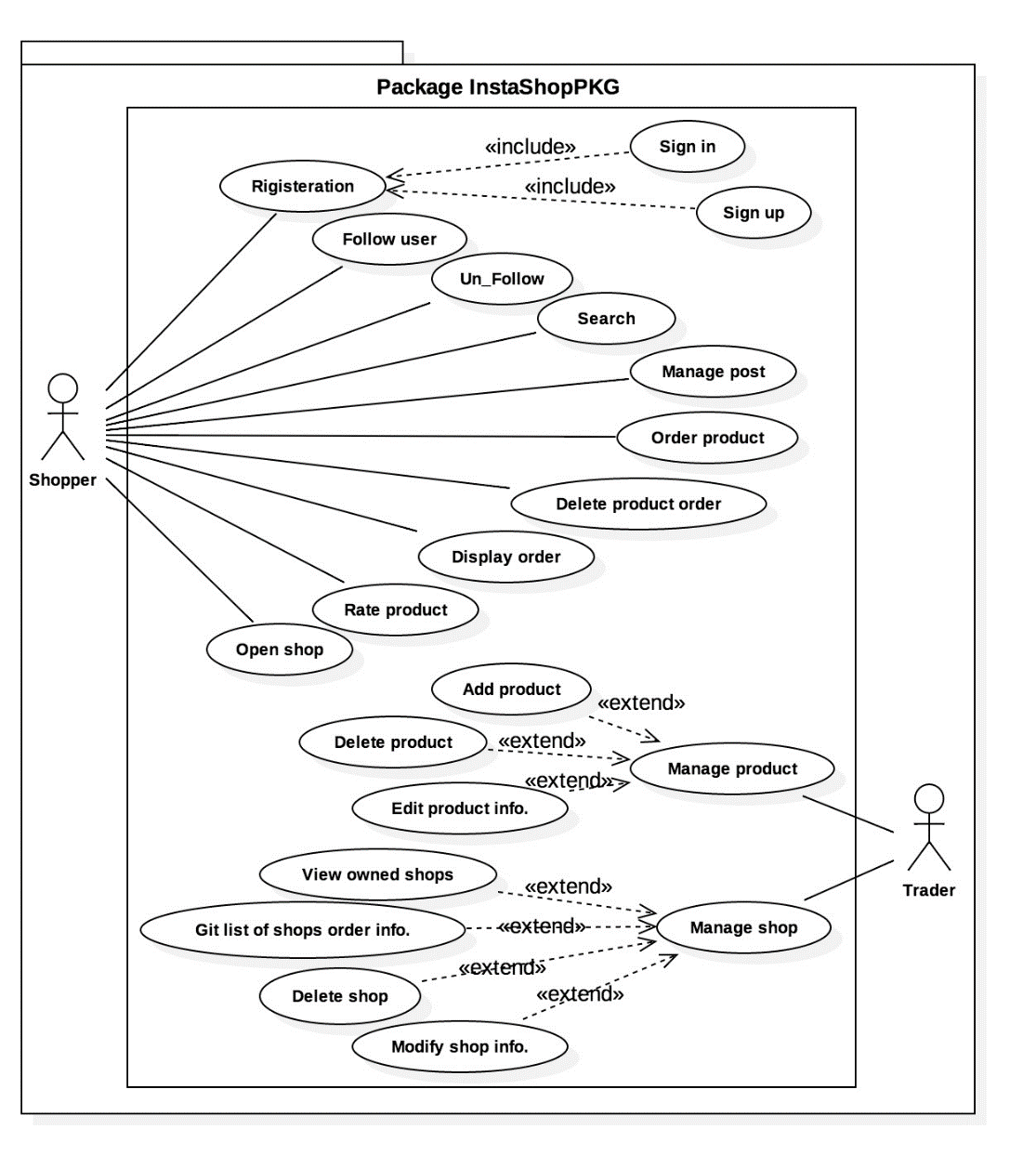
**There are two actors for the proposed system:**

**Shopper:**  A general user of the application, who order, buy products from sellers, search about products users, or shops. Any person use the application is a **Shopper** by default.

**Trader:** A user who opens shops to sell products.

**Note:** The trader can always act as a shopper, but not the opposite. Once the shopper opens a shop, he/she can act as a trader also.

# System Functions



## Shopper Functions

* + 1. **Searching**:

**• Identifier: REQ\_SHOPPER\_FUNC\_2.0**

**• Description and priority (very high)**

Any shopper can search about specific user, shop, product by their name and the system will retrieve all the related results.

**2.1.2 Personalized application access:**

* **Identifier: REQ\_SHOPPER\_FUNC\_1.0**
* **Description and priority (very high)**

User can register by new account with his own information (user name, password, email, first name, last name). Registered user can also, logging in to the application using his information (user name, password). In addition, registered user can retrieve the password whenever he/she forget it using the registered email.

* + 1. **Posts Managing**:

**• Identifier: REQ\_SHOPPER\_FUNC\_3.0**

**• Description and priority (high)**

Shopper can add different types of posts [text, photo, event, or comment]. Also, shopper can view, or delete any his/her written posts.

* + 1. **Timeline Displaying**:

**• Identifier: REQ\_SHOPPER\_FUNC\_4.0**

**• Description and priority (medium)**

Any shopper can load, or refresh his/her own timeline to appear the last updated posts.

* + 1. **Orders Managing**:

**• Identifier: REQ\_SHOPPER\_FUNC\_5.0**

**• Description and priority (high)**

Any shopper can order products’ items from any shop. The shopper can modify the order information such as the quantity, and the preferred color. Furthermore, the shopper can cancel any of his/her orders and it will be deleted from traders and shopper side. The shopper also, can view list of his/her all orders with a details information about each of them.

* + 1. **Rating:**

**• Identifier: REQ\_SHOPPER\_FUNC\_6.0**

**• Description and priority (high)**

Any shopper can choose one of the available rating’s measures for any product, or post. Once the rate is being added, the shopper can update it any time.

* + 1. **Opening Shop**:

• **Identifier: REQ\_SHOPPER\_FUNC\_7.0**

**• Description and priority (high)**

Any shopper can choose to open any number of electronic shops. Also, shopper can add shop’s information (name, shop picture, shop description, shop category).

* + 1. **Viewing a Shop and Product**:

**• Identifier: REQ\_SHOPPER\_FUNC\_8.0**

**• Description and priority (medium)**

Any shopper can view the shop profile to see the available uploaded products. Any user can view any product profile.

* + 1. **Following/ Un-Following other shopper accounts:**

**• Identifier: REQ\_ SHOPPER \_FUNC\_9.0**

**• Description and priority (medium)**

Any shopper can follow other shopper account to be his/her friend in the application. Any shopper can un-follow any of his/her following’ shoppers to delete the friendship relationship in the application.

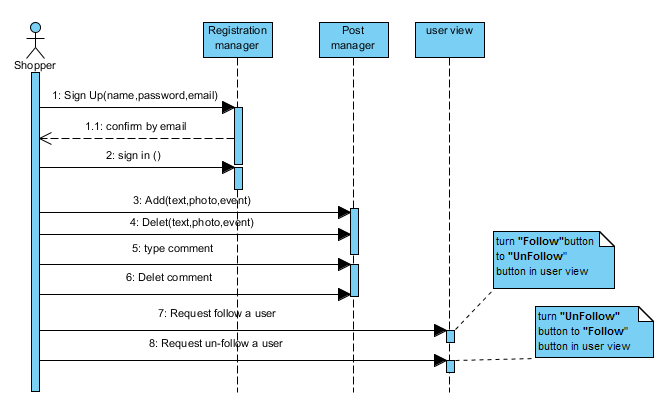
* + 1. **Profile Managing**:

**• Identifier: REQ\_SHOPPER\_FUNC\_10.0**

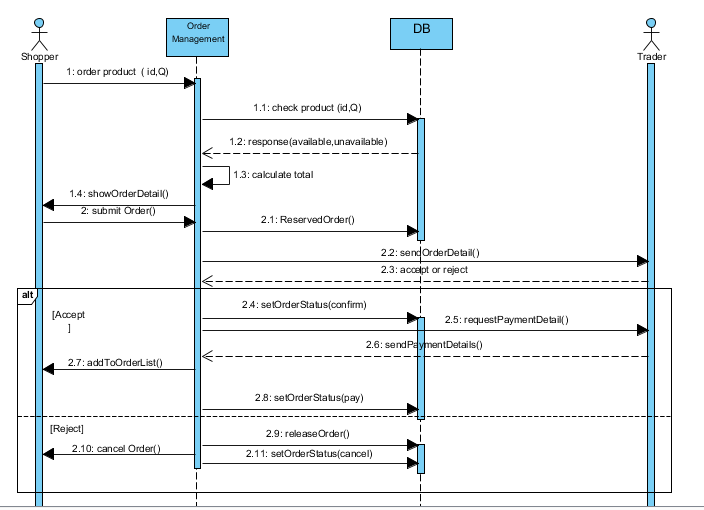
**• Description and priority (low)**

Any shopper can add more details personal information in his/her profile (Account Privacy status, picture, birthdate, phone number]. Any shopper can modify, or view his/her profile’s information.

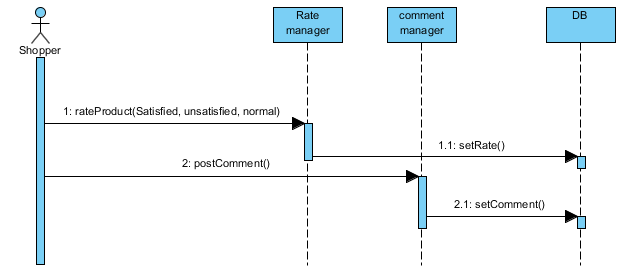
* **Action/Result**

****

**Description:** shopper will register by new account with his own information, after register shopper can logging in to the application using his information. Also, Shopper can add different types of posts [text, photo, event, or comment]. Also, delete any his/her written posts. Shopper can follow/un-follow other shopper account to be his/her friend in the application.

****

**Description**: shopper will choose a product for order, then he/she will receive and order details information. The shopper will submit the order; the trader will receive the order details. In case the trader accepts the order, the order will be confirming and the payment details will be send to the shopper. In case the trader rejects the order, the order will be cancel and the shopper will receive the order cancelation.



**Description:** shoppers can rate the displayed product which is one of these measures (Satisfied, unsatisfied, normal), then shopper can post comment about his/her opinion of product.

## Trader Functions

**2.2.1 Shop managing**:

**• Identifier: REQ\_TRADER\_FUNC\_11.0**

**• Description and priority (very high)**

Any trader can view the shop’s information, and its products. Modify its information, retrieve all information that related to the shop and delete the shop and all related contents.

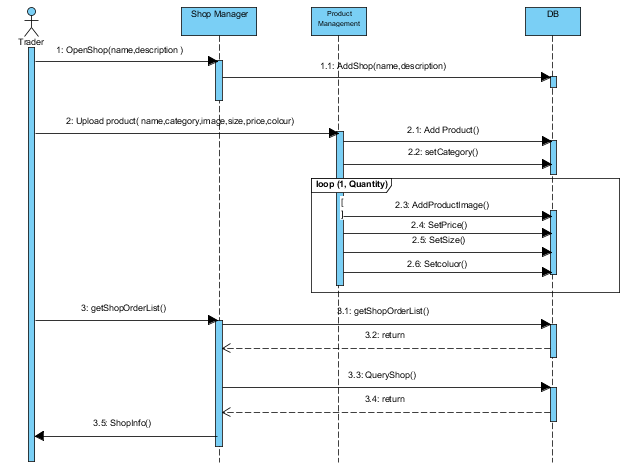
* + 1. **Products managing**:

**• Identifier: REQ\_TRADER\_FUNC\_12.0**

**• Description and priority (very high)**

Any trader can upload new product with its image. The trader can add /modify information of product and delete it.

* **Action/result**



**Description**: Trader canopen any number of electronic shops. Also, trader can add shop’s information (name, shop picture, shop description). Then, the trader can upload product and its information. The trader can retrieve all information that related to the shop.

### Functional requirements

|  |  |  |
| --- | --- | --- |
| ID | Description | Rationale |
| REQ\_SHOPPER\_FUNC\_1.1 | Logged in users can search for user by (user name, email), or for shop by its name or for product. The application suggests a list of all appropriate results | For finding shops, products, friends |
| REQ\_SHOPPER\_FUNC\_2.1 | User need to register with unique username, email, and unique password before browsing the application |  |
| REQ\_USER\_SHOPPER\_Exception\_2.1 | Invalid email address or weak password, or missing required field, the application will prevent user by error messages | In order to protect user security and identity. To prevent malicious person to guess user name and password. |
| REQ\_ SHOPPER \_FUNC\_2.2 | shopper need to login using registered user name and password |  |
| REQ\_SHOPPER\_FUNC\_Exception\_2.2 | shopper try to log in with incorrect user name or password, the application will prevent shopper by error messages |  |
| REQ\_ SHOPPER \_FUNC\_2.3 | Shoppers who have registered and forget their password, the shopper should click on the “forget password” link, then automatic link will be send to their email to change the old password and to allow them to enter their new password | For security concerns to prevent hacker from access the account |
| REQ\_ SHOPPER \_FUNC\_3.1 | Registered shoppers can post text and this post will appear in the followers timeline and in his\her profile, text length is limited to 200 letters and it can be edited by the shopper after posting | 1 letter=1 byte  200byte\* 500Post =100KB/day  This is the limit that we can manage  Also, we thinks that 200 letter is enough since the goal of app is commercial |
| REQ\_ SHOPPER \_FUNC\_3.2 | Registered shoppers can post photo by uploading pictures from upload button and this post will appear in the followers timeline, each image size must not more than 2MB | 2MB\*500post  =10 Giga/day  This is the limit that we can manage. |
| REQ\_ SHOPPER \_FUNC\_3.3 | Registered shoppers can make a comment by click on comment button, about any type of posts. then the time of the comment appears and the name of the user who write this comment | Help the traders in improving their products.  Creating and maintaining the relationship between the users of application |
| REQ\_SHOPPER\_FUNC\_3.4 | Registered users can post any event such as (graduation, party, etc) and ask their friends or followers to help them to find products related to the event. this post will appear in the followers timeline also. | Creating and maintaining the relationship between the users of application |
| REQ\_ SHOPPER \_FUNC\_4.1 | Logged in user can view his/her own published posts, the posts of other followed users such as ( text, photo….etc) and the system generated post such as birthday event by click on view timeline button | To show what the recent products based on following traders’, following friends’ activities. |
| REQ\_ SHOPPER \_FUNC\_5.1 | Logged in shoppers after viewing product profile, he/she can order product he/she liked from any shop to buy it by clicking on order button, then the shopper will choose the product (quantity, size, color) then the shopper will click on submit if he/she really want to buy.  after that the order request will sent to the seller. The order information will record and displayed in the shopper and trader order list. |  |
| REQ\_USER\_FUNC\_5.2 | After order is sent the buyer have a chance to modify the order (quantity, color, size, etc) by click on the Order Modify button. | From usability perspective |
| REQ\_SHOPPER\_FUNC\_5.3 | The shopper can delete or cancel his/her order by click on the cancel button, then the order cancellation will record at the shopper and trader order list. | From usability perspective |
| REQ\_ SHOPPER \_FUNC\_5.4 | The shopper has an option to display all the orders he/she did for products by click on My Orders, to retrieve the order history. The list of order history is ordered ascending based on the time created of each order, the newest date order is the first to be displayed. Also the list is displayed with order status (Active, submit, confirmed, paid, cancelled) and order details( Price ,date, order quantity) | Providing up-to-date information |
| REQ\_ SHOPPER \_FUNC\_6.1 | Logged in shoppers can rate the displayed product, by clicking on the desired rate button, the rate measures are based on three different emotions. Each of them has different meaning (Satisfied, unsatisfied, normal) the number of rate is displayed to the chosen rate button | In order to help the traders in improving their products, and inform shopper friends and family about product quality, this rate will help other customers to ensure the credibility of the trader. |
| REQ\_USER\_FUNC\_6.2 | Logged in shoppers after rate, he/she can also change his/her rates any time | From usability perspective |
| REQ\_SHOPPER\_FUNC\_Exception\_6.2 | The application prevent the users to choose more than one rate at a time for each product. | For trustworthy and accuracy |
| REQ\_SHOPPER\_FUNC\_7.1 | Logged in shopper have a choice to open any number of shops by click on open shop button to start selling and can add or change any shop profile information. The profile include the basic information about the shop which is visible to all other users, The information automatically added to the user profile indicates that the user has a shop and he/she become trader. |  |
| REQ\_ SHOPPER \_FUNC\_8.1 | Logged in user while browsing the app can view shop profile by click on the shop name and see the uploaded products. Then, by click on product image, the user can view the product profile to check its information (product name, quantity, description, price, product picture). |  |
| REQ\_ SHOPPER \_FUNC\_9.1 | a logged in shopper can follow/un-follow other users by click on follow/un-follow button to get their post in his/her timeline | Increasing brands awareness and reputation |
| REQ\_ SHOPPER \_FUNC\_9.2 | If a shopper account is private, no other users can follow him until accepts the following request. and if a shopper un-followed another he will stop getting his posts |  |
| REQ\_SHOPPER\_FUNC\_10.1 | Once the user log in correctly in the application he/she has his/her own profile and he/she can update his/her profile information (first and last name, picture, birthday, phone number). |  |
| REQ\_SHOPPER\_FUNC\_Exception\_10.1 | Logged in users enter invalid phone number, the application automatically prevent the user by error message |  |
| REQ\_SHOPPER\_FUNC\_10.2 | Logged in users have the choice to make their account private, since the default choice of account privacy is to be public. User’s private account can only view by his followers, non-followers can only see the user (name, picture). |  |
| REQ\_TRADER\_FUNC\_11.1 | Any shopper who open shop become a trader and he/she can view it’s shops information. He/she can modify any of shop’s information(shop name, shop picture, shop description, shop category) by click on edit button and update the shops information. |  |
| REQ\_TRADER\_FUNC\_11.2 | The trader has an option to display all orders he received from shoppers for each shop separately, trader has to click on ShopOrders to query the list of products which ordered by buyers and to retrieve all information that related to the shop, and the transactions that done through it. | Providing up-to-date information |
| REQ\_ TRADER \_FUNC\_11.3 | The trader can delete the shop and all related contents by click on delete button |  |
| REQ\_TRADER\_FUNC\_12.1 | The trader can upload new product by adding its information (product name, product quantity, product color, picture, price) by click on upload button where trader can upload the image of the product. |  |
| REQ\_ TRADER \_FUNC\_12.2 | The trader can modify the uploaded product and its information (product name, product quantity, product color, picture, price) by click on edit button. |  |
| REQ\_ TRADER \_FUNC\_12.3 | The seller can delete any of the uploaded products and its related information |  |

# Nonfunctional Requirements

## Performance requirements

* **Identifier: REQ\_NONFUNC\_1.0**
* **Description and priority (high)**

The software must behave under critical conditions -as low battery, bad network coverage, low available memory. It must be also try to choose good algorithms to increase the software speed and to have best memory utilization.

## Safety requirements

* **Identifier: REQ\_NONFUNC\_2.0**
* **Description and priority (medium)**

Both trader and shopper want to guarantee their commercial benefits. The application should provide a guidance to users of how to pay or sell and it must show a warning message to both shopper and trader to take care about this point.

## Security requirements

* **Identifier: REQ\_NONFUNC\_3.0**
* **Description and priority (medium)**

The application must be secure against any malicious users. Some users may steal any user's account who has a common shop or hundreds of followers.

## Software quality attributes

* **Identifier: REQ\_NONFUNC\_4.0**
* **Description and priority (very high)**

|  |  |
| --- | --- |
| **ID** | **Description** |
| **REQ\_NONFUNC\_4.1** | **Maintainability:**  The application must be easy to maintain, so developers can update it by correcting possible errors or by adding a new feature to the software like developing a new selling method which is by Visa card or e.g. adding a feature that allow the user to know who looks at his profile. Consequently, the software must be highly maintainable. |
| **REQ\_NONFUNC\_4.2** | **Robustness:**  When a fault is happening in the application, it must be robust and dose not damage the entire application and can continue even if the faults didn’t correct yet. Also from the data replication aspect, the database should be backed up weekly in a separated server, so if a failure happens we can retrieve the data. |
| **REQ\_NONFUNC\_4.3** | **Usability:**  The application has competitor apps, it must support an attractive interface and services to users to gaining them. Also it must be easy to use, so everyone can use it and it must be just for professional users or for say computer experts. |

## Other Operational requirements

**Hardware Requirements**

1. Smart phones running an Android OS and Internet connection.

**Software Requirements**

1. Developing languages: Java
2. Development environment: Eclipse Integrated Development Environment (IDE), Software Development Kit for Java (SDK), Android development Tool (ADT) plugin for UI.

1. . The word freemium is made up from the words free and premium [↑](#footnote-ref-1)