**SOFTWARE REQUIREMENTS SPECIFICATION**

**PROJECT: [*SwiftShop* 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 |
|  |  |  |  |

**TABLE OF CONTENTS**

[1. Description 4](#_Toc342728864)

[1.1. Business perspective 4](#_Toc342728865)

[1.2. Business Product/Service functions 4](#_Toc342728866)

[1.3. User classes and characteristics 4](#_Toc342728867)

[2. System Functions 5](#_Toc342728868)

[2.1. System function 1 5](#_Toc342728869)

[2.1.1. Identifier 5](#_Toc342728870)

[2.1.2. Description and priority 5](#_Toc342728871)

[2.1.3. Action/result 5](#_Toc342728872)

[2.1.4. Functional requirements 5](#_Toc342728873)

[3. Nonfunctional Requirements 7](#_Toc342728874)

[3.1. Performance requirements 7](#_Toc342728875)

[3.2. Safety requirements 7](#_Toc342728876)

[3.3. Security requirements 7](#_Toc342728877)

[3.4. Software quality attributes 7](#_Toc342728878)

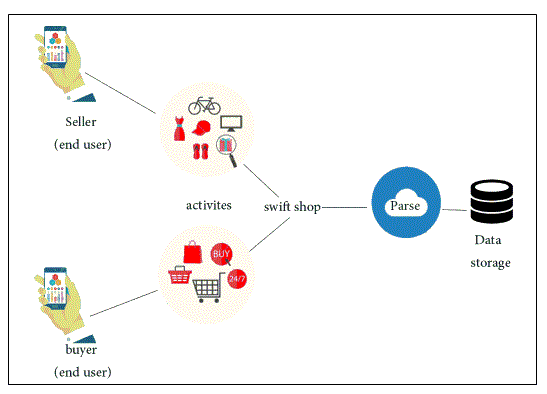
[3.5. Other Operational requirements 8](#_Toc342728879)

**Introduction**

# Description

## Business perspective

The *SwiftShop* mobile application is a Social Network for shopaholic and trading enthusiasts, is a new mobile application which developed on the android platform using Parse as a back end.



## Business Product/Service functions

The SwiftShop 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. 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. In SwiftShop, 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 SwiftShop’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.  In addition, the application itself will recommend products to users according to others 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. They left the way of using certified familiar commercial sites, or applications to spread their products. Instead, they use a familiar social network -that contains a huge number of users, which might be their customers one day- to share their products with different methods of sharing such as, text posts and photo posts.

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

**User:**  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 User by default.

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

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

صور

<http://www.servicetechmag.com/I79/1213-3>

<http://www.slideshare.net/thest0rm/v2v-design-presentation-final>

business revenue

<https://www.quora.com/What-are-the-various-sources-of-mobile-apps-earning-revenue>

<https://www.quora.com/How-do-I-earn-revenue-through-a-free-free-education-site-mobile-app>

<https://www.quora.com/What-are-some-of-the-most-successful-mobile-apps-revenue-models>

<http://www.developereconomics.com/how-to-make-money-with-apps/>

<https://www.quora.com/How-much-money-can-an-Android-developer-earn-from-a-free-app-on-the-Google-Play-store>

<http://www.avocarrot.com/blog/can-make-money-building-mobile-apps-developer-survey/>

<http://www.itworld.com/article/2957215/development/how-to-make-money-as-an-independent-developer.html>

<http://ibuildapp.com/make-money-by-creating-and-selling-your-own-mobile-features/>

<http://blog.personagraph.com/show-me-the-money-business-models-for-app-developers>

<https://www.linkedin.com/pulse/mobile-app-development-which-business-model-right-your-peter-brooke>

# System Functions

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. Organize this section by functional hierarchy.>

## User Functions

* **Personalized application access:**
* **Identifier: REQ\_USER\_FUNC\_1.0.0**
* **Description and priority (very high)**
  1. User can register by new account with his own information [user\_name, password, e\_mail, first\_name, last\_name].
  2. Registered user can logging in to the application using his information [user\_name, password].
  3. Registered user can retrieve the password whenever he/she forget it using the registered email.
* **Action/result**
* **Following/ Un-Following other user accounts:**

**• Identifier: REQ\_USER\_FUNC\_2.0.0**

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

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

**• Action/result**

* **Posts Managing**:

**• Identifier: REQ\_USER\_FUNC\_3.0.0**

**• Description and priority (medium)**

1. User can add different types of posts [text, photo, event, or comment].
2. User can view, or delete any his/her written posts.

* **Timeline Displaying**:

**• Identifier: REQ\_USER\_FUNC\_4.0.0**

**• Description and priority (high)**

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

* **Opening Shop**:

• **Identifier: REQ\_USER\_FUNC\_5.0.0**

**• Description and priority (high)**

* 1. Any user can choose to open any number of electronic shops.
  2. User can add shop’s information [shop\_name, shop\_pic, shop\_description, shop\_category].
* **Viewing a Shop and Product**:

**• Identifier: REQ\_USER\_FUNC\_6.0.0**

**• Description and priority (medium)**

* 1. Any user can view the shop profile to see the available uploaded products.
  2. Any user can view any product profile.
* **Profile Managing**:

**• Identifier: REQ\_USER\_FUNC\_7.0.0**

**• Description and priority (medium)**

* 1. Any user can add more details personal information in his/her profile [ Account\_Privacy\_status, picture, birthdate, phone\_number].
  2. Any user can modify, or view his/her profile’s information.
* **Orders Managing**:

**• Identifier: REQ\_USER\_FUNC\_8.0.0**

**• Description and priority (high)**

* 1. Any user can order products’ items from any shop.
  2. The user can modify the order information such as the quantity, and the preferred color.
  3. The user can cancel any of his/her orders and it will be deleted from seller and customer side.
  4. The user can view list of his/her all orders with a details information about each of them.
* **Rating:**

**• Identifier: REQ\_USER\_FUNC\_9.0.0**

**• Description and priority (high)**

* 1. Any user can choose one of the available rating’s measures for any product, or post.
  2. Once the rate is being added, the user can update it any time.
* **Searching**:

**• Identifier: REQ\_USER\_FUNC\_10.0.0**

**• Description and priority (low)**

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

## Seller Functions

* **Shop managing**:

**• Identifier: REQ\_SELLER\_FUNC\_11.0.0**

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

* 1. Any seller can view the shop’s information, and its products.
  2. The seller can query the shop to retrieve all information that related to the shop, and the transactions that done through it.
  3. The seller can modify any of shop’s information [shop\_name, shop\_pic, shop\_description, shop\_catagory].
  4. The seller can delete the shop and all related contents.
* **Products managing**:

**• Identifier: REQ\_SELLER\_FUNC\_12.0.0**

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

* 1. Any seller can upload new product with its image.
  2. The seller can add /modify information for the uploaded product [prod\_name, prod\_quantity, prod\_description, prod\_pricce, product\_color].
  3. The seller can delete any of the uploaded products.

### Functional requirements

|  |  |  |
| --- | --- | --- |
| ID | Description | Rationale |
| REQ\_USER\_FUNC\_1.0.1 | User need to register with unique username, email, and unique password before browsing the application | * Browsing the app * User registration |
| REQ\_USER\_FUNC\_Exception\_1.0.1 | Invalid email address or weak password, or missing required field, the application will prevent user by error messages | * Create account. |
| REQ\_USER\_FUNC\_1.0.2 | User need to login using registered user name and password | -Browsing the app  -buy product  -sell product |
| REQ\_USER\_FUNC\_ Exception \_1.0.2 | User try to log in with incorrect user name or password, the application will prevent user by error messages | -user log in |
| REQ\_USER\_FUNC\_1.0.3 | Users who have registered and forget their password, the user 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 | -Password retrieval |
| REQ\_USER\_FUNC\_2.0.1 | a logged in user can follow/un-follow other users by click on follow/un-follow button to get their post in his/her timeline. | follow/un-follow |
| REQ\_USER\_FUNC\_2.0.2 | If a user account is private, no other users can follow him until accepts the following request. and if a user un-followed another he will stop getting his posts | follow/un-follow |
| REQ\_USER\_FUNC\_3.0.1 | Registered users 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 user after posting | -posting text |
| REQ\_USER\_FUNC\_3.0.2 | Registered users 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 | -posting photo |
| REQ\_USER\_FUNC\_3.0.3 | Registered users 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 | -posting comments |
| REQ\_USER\_FUNC\_3.0.4 | 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. | * Posting an event |
| REQ\_USER\_FUNC\_4.0.1 | 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 | * Timeline displaying |
| REQ\_USER\_FUNC\_5.0.1 | Logged in user 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 | * Open shop |
| REQ\_USER\_FUNC\_6.0.1 | 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, product\_quantity, product\_description, product\_price, product\_ picture]. | * Browsing the app |
| REQ\_USER\_FUNC\_7.0.1 | 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). | * Modify profile information |
| REQ\_USER\_FUNC\_Exception\_7.0.1 | Logged in users enter invalid phone number, the application automatically prevent the user by error message | * Insert phone number |
| REQ\_USER\_FUNC\_7.0.2 | 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 viewed by his followers, non-followers can only see the user (name, picture). | * Account privacy |
| REQ\_USER\_FUNC\_8.0.1 | Logged in users after viewing product profile, he/she can order product he/she liked from any shop to buy it by clicking on order button, after that the order request will sent to the seller. The order information will record and displayed in the seller and buyer order list. | * Order product |
| REQ\_USER\_FUNC\_8.0.2 | After order is sent the buyer have a chance to modify the order |  |

# Nonfunctional Requirements

## Performance requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software quality attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Other Operational requirements