

Software Design Document for A Mobile Barter Shop

-EPIC FAIL-

Anıl Doğan

Bahadır Hatunoğlu

Onur Özkan

Semih Baştak

December 29, 2013

Version1.1

B-xchange Project

Changelog!

*A: Added, M: Modified, D: Deleted

Version number	Date	Section	A,M,D*	Title or Brief Description
Version 1.0	1/12/2013			Original.
Version 1.1	29/12/13	Section 3.9	A	Second paragraph is added for clear specification and understanding.
		3.6.1.22	A	An Use case added
		3.6.1.23	A	An Use case added
		3.6.1.24	A	An Use case added
		3.6.1.25	A	An Use case added
		3.6.1.26	A	An Use case added
		3.6.1.27	A	An Use case added
		4	A	Project Planning added

Preface

This document contains the system design information for B-xchange project. The document is prepared according to the “IEEE Standard for Information Technology – Systems Design – Software Design Descriptions – IEEE Std. 1016 – 1998”.

This Software Design Documentation provides a complete description of all the system design and views of B-xchange Project.

The first section of this document presents scope, purpose, overview, intended audience, reference material and definitions and abbreviations of the project.

The second chapter describes the conceptual model for software design descriptions.

The third chapter of this document specifies the design viewpoints for the whole system.

The last chapter of this document includes requirements matrix of the B-xchange project.

Contents

1	INTRODUCTION	8
1.1	Scope	8
1.2	Purpose	8
1.3	Overview	8
1.4	Intended Audience	9
1.5	Reference Material	9
1.6	Definitions and Acronyms	9
2	SYSTEM OVERVIEW	11
2.1	Technologies Used	11
2.2	Application Overview	11
3	DESIGN VIEWPOINTS	13
3.1	Composition Viewpoint	13
3.2	Patterns use Viewpoint	14
3.3	Information Viewpoint	15
3.4	Structure Viewpoint	17
3.5	Interface Viewpoint	18
3.5.1	Overview of User Interface	18
3.5.2	Screen Images	18
3.5.2.1	Welcome to Bxchange Screen	19
3.5.2.2	Login Screen	19
3.5.2.3	Register Screen	20
3.5.2.4	Main Page	21
3.5.2.5	Profile Page	22
3.5.2.6	Search Result Screen	23
3.5.2.7	Product Description Page	24
3.5.2.8	Favorite Items Page	24
	Figure 10 – Product Description Page	24
3.5.2.9	Favorite Sellers Page	25
3.5.2.10	Messages Page	25
3.5.2.11	Shopping Cart Page	26

3.5.2.12 Publish Page	26
3.5.2 Screen Objects and Actions	28
3.6 Context Viewpoint	30
3.6.1 Use cases	30
3.6.1.2 Use Case : Login	31
3.6.1.3 Use Case : Log Out.....	31
3.6.1.4 Use Case : Search	32
3.6.1.5 Use Case : Browse Profiles	32
3.6.1.6 Use Case : Browse Categories.....	32
3.6.1.7 Use Case : Browse Items	33
3.6.1.8 Use Case : Edit Profile	33
3.6.1.9 Use Case : Buy Credit	33
3.6.1.10 Use Case : Add Item(s) to Cart.....	34
3.6.1.11 Use Case : Purchase Item(s) with Credit.....	34
3.6.1.12 Use Case : Request an Exchange.....	34
3.6.1.13 Use Case : Approve the Exchange	35
3.6.1.14 Use Case : Decline the Exchange	35
3.6.1.15 Use Case : Notify for Shipment.....	35
3.6.1.16 Use Case : Add Item.....	36
3.6.1.17 Use Case : Edit Item	36
3.6.1.18 Use Case : Remove Item	36
3.6.1.19 Use Case : Delete Profile.....	37
3.6.1.20 Use Case : Mark as a Favorite	37
3.6.1.21 Use Case : Rate and Comment	37
3.7 State Dynamics Viewpoint.....	40
3.7.1 Adding Item.....	Error! Bookmark not defined.
3.7.2 Viewing, Rating, Commenting, Adding to Cart and Marking items as Favorite	Error! Bookmark not defined.
3.7.3 Accepting and Rejecting Exchange Request.....	Error! Bookmark not defined.
3.7.4 Deleting Account, Logging Out, Editing Item, Buying Credits	Error! Bookmark not defined.
3.8 Interaction Viewpoint.....	44
3.8.1 Login Controller	44
3.8.2 Register Controller	44

3.8.3 Profile Controller.....	45
3.8.4 Item Profile Controller	45
3.8.5 Favorite Seller Page Controller	46
3.8.6 Favorite Item Page Controller	46
3.8.7 Message Controller.....	47
3.8.8 Search Page Controller	47
3.8.9 Shopping Cart Page Controller.....	48
3.8.10 Picture Upload Controller.....	48
3.8.11 Title Category Controller	49
3.8.12 Specifications Controller	49
3.8.13 Format Price Controller	50
3.8.14 Submit Controller	50
3.9 Logical Viewpoint.....	51
3.9.1 Login Controller	51
3.9.2 Register Controller	52
3.9.3 Profile Controller.....	52
3.9.4 ItemProfileController	52
3.9.5 FavoriteSellerPageController	53
3.9.6 FavoriteItemPageController	53
3.9.7 MessagePageController.....	53
3.9.8 SearchPageController	54
3.9.9 ShoppingCartPageController.....	54
3.9.10 PictureUploadController.....	54
3.9.11 TitleCategoryController	54
3.9.12 CISDController	55
3.9.13 FPPController	55
3.9.14 SubmitController	56
3.9.15 Repository Service.....	56
3.9.16 User Class (Model).....	56
3.9.17 UserController	57
3.9.18 Item Class (Model).....	57
3.9.19 ItemController	58
3.9.20 Messages Class (Model).....	58

3.9.21	MessageController.....	58
3.9.22	Comment Class (Model).....	58
3.9.23	CommentController.....	59
3.9.24	FavoriteItem Class (Model).....	59
3.9.25	FavoriteItemController.....	59
3.9.26	FavoriteSeller Class (Model).....	59
3.9.27	FavoriteItemController.....	59
3.9.28	ShoppingCart Class (Model).....	59
3.9.29	ShoppingCartController	59
4	REQUIREMENTS MATRIX	62

1 INTRODUCTION

1.1 Scope

This document gives a detailed description of B-xchange which will be an Android application of an exchange system by which goods or services are directly exchanged by other goods or services. B-xchange will provide users the following functionalities:

- Put goods/services to the barter market.
- Search goods/services in the barter market
- Search goods/services according to geographic information
- Follow goods/services
- Manage previous trades/exchange history
- Continue trade without having real money

1.2 Purpose

Purpose of this document is to present a detailed description of conceptual design of B-xchange Project according to IEEE STD 1610-2009.

This document shows how the software system of B-xchange project will be structured in order to satisfy the requirements identified in system requirement specification document. Details of major software components and expected functionalities of the system with diagrams will be defined in this document.

1.3 Overview

This document prepared to help reader to understand and visualize the solution to the project presented. This SDD shows how the system software will be structured to satisfy requirements stipulated in SRS document through viewpoints.

1.4 Intended Audience

The intended audience of this document is end users, developers and testers of B-xchange in order to provide guidance about selection, organization, and presentation of design information.

1.5 Reference Material

IEEE, IEEE STD. 1016-1998 Recommended Practice for Software Design Descriptions, 1998-09-23, The Institute of Electrical and Electronics Engineers, Inc., (IEEE)

IEEE, IEEE 1016 Software Design Document (SDD) Template for CENG491

1.6 Definitions and Acronyms

TERM	DEFINITION
Database	Collection of all the information monitored by this system.
User	Any person or company logged on the system.
Android	A mobile device operating system developed by Google Inc.
Social Media	Facebook or twitter.
IEEE	The Institute of Electrical and Electronics Engineers (IEEE) is a professional association headquartered in New York City that is dedicated to advancing technological innovation and excellence.
Software Requirements Specification (SRS)	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

System Design Document (SDD)	A document contains the system design information which provides a complete description of all the system design and views of project.
ADT (Android Developer Tools) Bundle	A bundle of Android developer tools which contain the essential Android SDK components and a version of the Eclipse IDE with built-in ADT to streamline android app development.
Eclipse IDE	Eclipse is a multi-language Integrated Development Environment comprising a base workspace and an extensible plug-in system for customizing the environment.
HTTP	Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems.
REST	Representational State Transfer (REST) is an architectural style consisting of a coordinated set of constraints applied to components, connectors, and data elements, within a distributed hypermedia system to achieve desired architectural properties.
Jetty	It is a web server that provides an HTTP server and Servlet container capable of serving static and dynamic content either from a standalone or embedded instantiations

2 SYSTEM OVERVIEW

2.1 Technologies Used

The system is developed with Android SDK which provides us the API libraries and developer tools necessary to build, test, and debug Android applications.¹ ADT bundle is preferred for development since it includes the essential components for Android application development and an Eclipse IDE with Android Developer Tools (ADT). ADT bundle simply includes:¹

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform Tools
- The latest Android Platform
- The latest Android system image for the emulator

We will use MongoDB as database since it is an open source document database. REST interface will be used as HTTP interface which based on Jetty, namely MongoDB Java REST server.² Jetty provides a webserver and javax.servlet container and supports many integrations such as SPDY, WebSocket, OSGI, JNDI and etc.³ The storage of MongoDB is document oriented and includes JSON style documents with dynamic schemas offer simplicity and power.⁴

2.2 Application Overview

Our project aims to collect people who want to share/exchange goods or services without money on a mobile platform which provides them a user friendly application on this purpose. The end product of B-xchange will be a mobile platform for Android devices such as mobile phones or tablet systems.

¹<https://developer.android.com/sdk/index.html>

²<http://docs.mongodb.org/ecosystem/tools/http-interfaces/>

³<http://www.eclipse.org/jetty/>

⁴<http://www.mongodb.org/>

The B-xchange project is principally designed for users who want to exchange goods or services for other goods or services which are desired to. The trade runs on some virtual currency which named Bx-coin. Instead of money, users will get Bx-coins valued at goods or services they exchanged or sold. The application also allows users to buy Bx-coins if they do not have anything to exchange or when they do not have enough Bx-coins. The reason behind this procedure is to achieve aptitude of moneyless trade after a while. More clearly, users will probably use money when they sign up the system since there is no exchange happened and no Bx-coin collected, but after some trades, most users will have various amounts of Bx-coins which allow them to continue trade with using Bx-coins instead of money. More simply after a while, the application runs on existing Bx-coins which are created at the entrance of system and flowing between users.

The system will be designed with functionalities to help the user to:

- Put goods/services to the barter market
- Search for goods/services at the market
- Follow goods/services
- Manage previous trades/ exchange history
- Filter search according to geographic information
- Continue trade without having real money

3 DESIGN VIEWPOINTS

In this section of the document, brief explanations of the design viewpoints are given.

A viewpoint is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views.⁵

3.1 Composition Viewpoint

B-xchange composition viewpoint describes how the project design structured into sub modules and the roles of these modules. Below figure 1 shows the deployment diagram for B-xchange system.

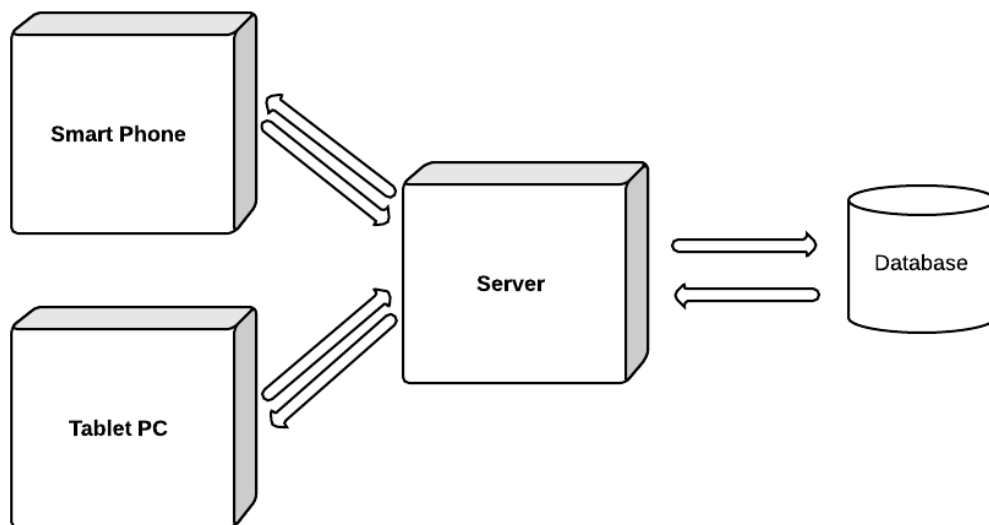


Figure 1 – The deployment diagram of B-xchange System

⁵<http://www.viewpoints-and-perspectives.info/home/viewpoints/>

User invokes the server by an android phone or tablet pc. The request from user will be handled by server side application which uses REST Interface. It will do the required processes, communicate with database and return a response to the phone/tablet. The data will be understandable by phone/tablet side of the application and that will result to a view created by controllers to the user, which is the view part of MVC pattern in this case.

3.2 Patterns use Viewpoint

A design pattern can be defined as a general repeatable solution to a commonly occurring problem within a given context in software design.⁶ This section of the document contains patterns use viewpoint which addresses design ideas as collaboration patterns involving abstracted roles and connectors.⁷ All design patterns are reusable and can be adapted to particular context.

- **Model View Controller Pattern:** is a software architecture pattern which separates the modeling of the domain, the presentation, and the actions based on user input into three separate classes which are, Model, View, and Controller.
 - **Model:** Manages the behavior and data of application. It notifies its associated views and controllers when there has been a change in its state. With this notification views produce output and controllers change the set of commands.
 - **View:** Manages the display of information. It request information, which needs for creating an output representation to the user, from the model.
 - **Controller:** Interprets the inputs from user (mouse, keyboard, touchscreen) then informing the model and/or the view. It can send commands to the model or its associated view to update its state.

- **Chain of Responsibility:** Sign in/Log in method uses authenticate to validate the user profile. If the user entered correct combination of username and password, the method directs the user to main page. Otherwise, it will ask the username and password again. That makes this method to be an example of chain of responsibility.

⁶http://sourcemaking.com/design_patterns

⁷<http://www.it.iitb.ac.in/arndg/brain2013/sites/default/files/sdd.pdf>

3.3 Information Viewpoint

B-xchange has a database that holds the persistent data content about all the information and data related to the system. Administrators of the system can interact with this database directly. End users only interact with the database via user interface, indirectly.

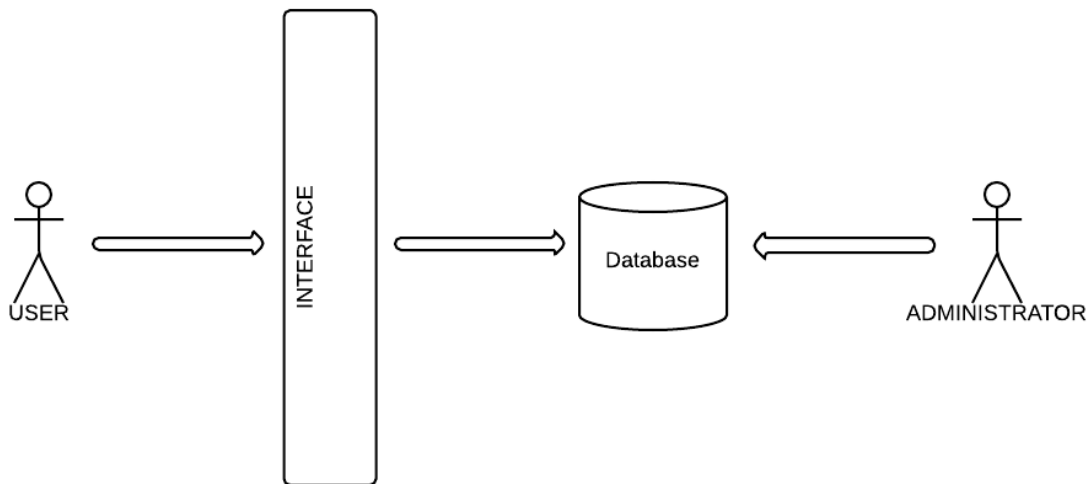
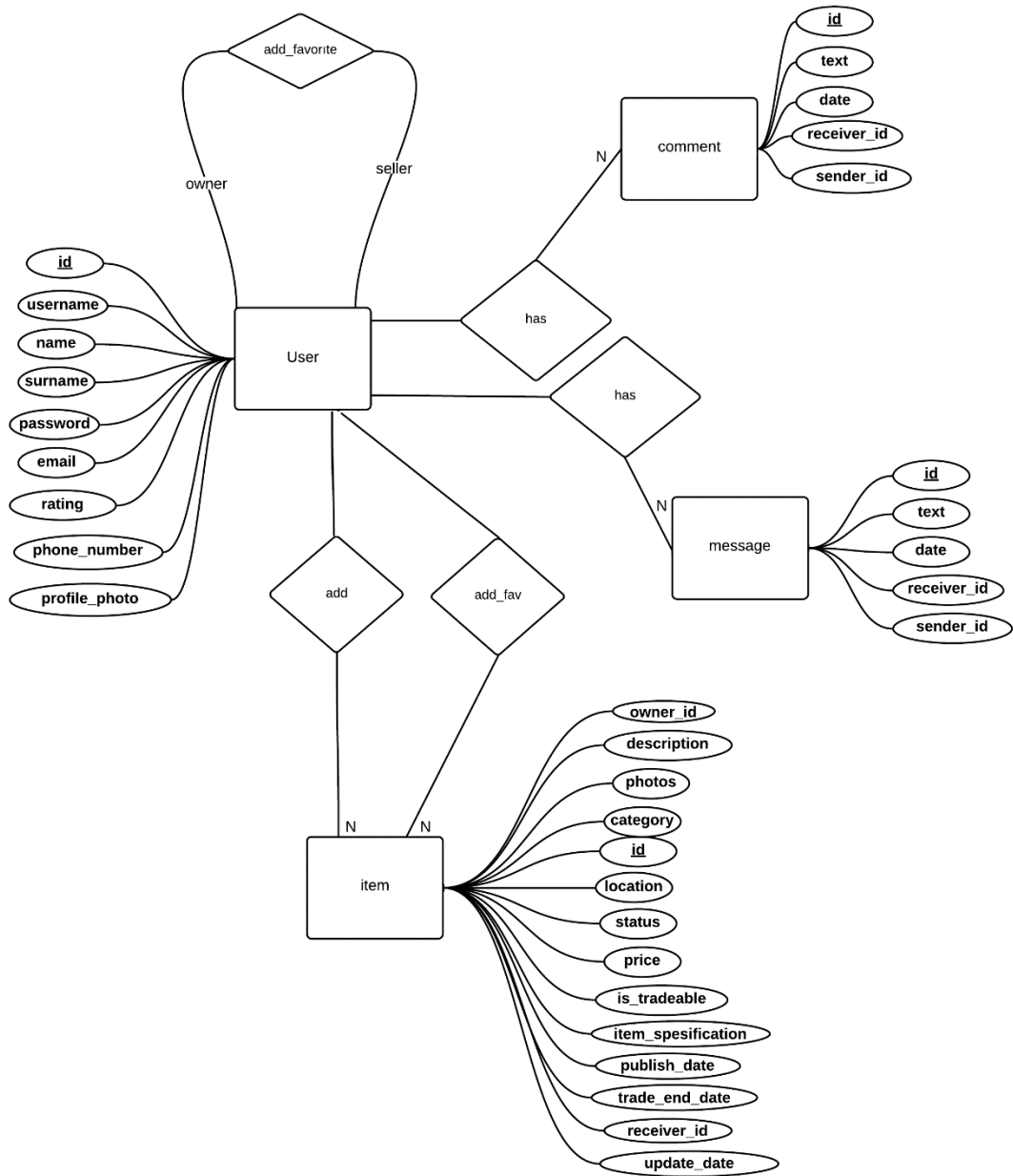


Figure 2– Information Viewpoint Diagram

This E-R Diagram which is shown below this explanations explains general data design for B-xChange project. All of the information is kept in the database which can only be accessed by Server-side application. There are several relationships among entities as below:

1. First a user can add multiple items so this relationship is 1-N.
2. Add-fav relation is N-N because several items can be added to a user's t favorites and an user can add several items to their favorites.
3. User can have several comments, therefore this relation is 1-N.
4. Other relations is that a user can have several messages so this relation is also 1-N.
5. Finally a user can add several user to their favorite user list and a user can be added to several users' favorite user list so this relation is N-N like add-fav relation.



3.4 Structure Viewpoint

Structural view is often equated with the architecture of the software system.⁸B-xchange structure depends on Model View Controller pattern. It uses the REST Interface as HTTP interface. There is a database, which is MongoDB, and server connection rely on REST API. The view is generated by controllers using models. Each model has a controller and related to database. Users can indirectly interact database through user interface, while administrators have a direct interaction. Below figure 3 shows brief structure of B-xchange.

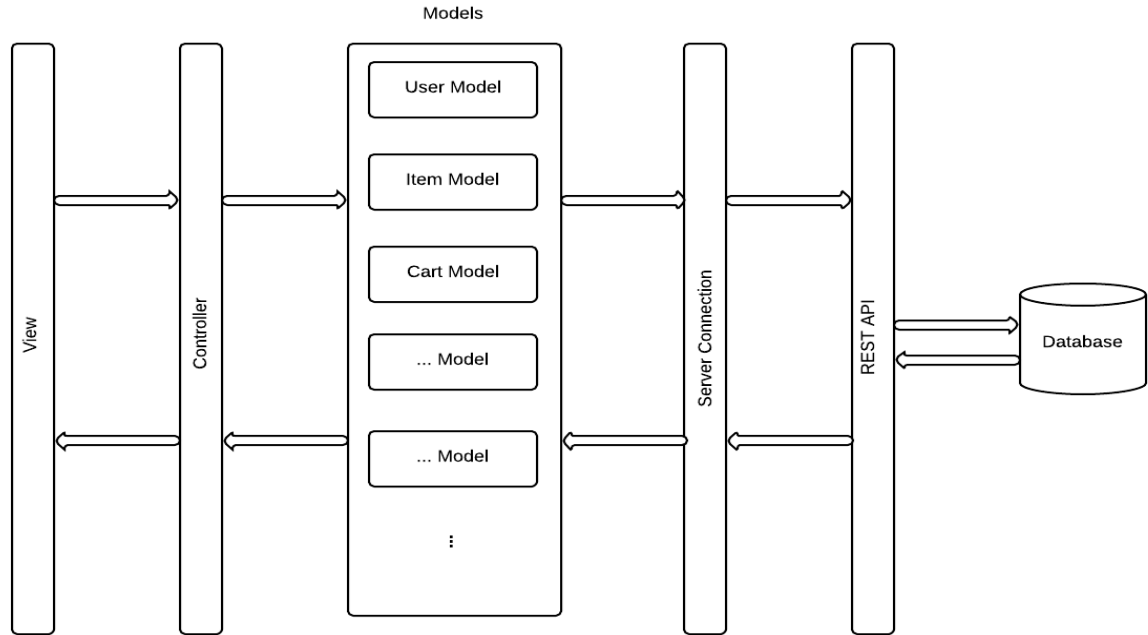


Figure 3 – Draft structure model of B-xchange

⁸http://www.researchgate.net/publication/2358090_Views_and_Viewpoints_in_Software_Systems_Architecture/file/d912f50b51e19ec348.pdf

3.5 Interface Viewpoint

3.5.1 Overview of User Interface

When the user launches barter application, it directs to sign in / register screen .If the user has already have an account, they can fill the required information (username password) to login to Bxchange application, otherwise user can use the options register or take a tour without login. If the user wants to register to the system, he/she clicks the register button and then needs to fill the register form with some personal information (name, surname, username, password and email). After this process, user will be prompted searching page. The user can search any item from this page in all categories or specific one. In this page, the user can reach his/her own profile page by using “my account” button. In the profile page, the user can view owned sold, bought and exchanged items history, or manage his/her items put for exchange/sale, favorite items/sellers and account settings. They can easily add new item(s) for exchange/sale or can see purchased stuffs until now. Moreover, users may see the sellers profile history in order to find good and reliable sellers according to their ratings.

3.5.2 Screen Images

This section briefly describes the interfaces and interface components of Bxchange.

3.5.2.1 Welcome to B-xchange Screen

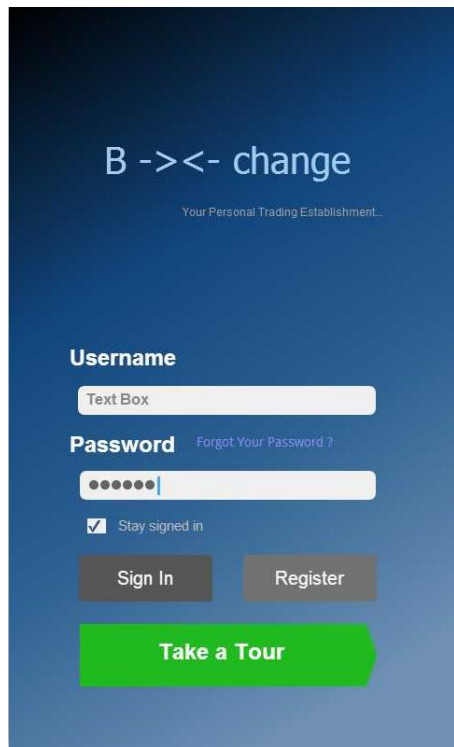


Figure 1 – Welcome Screen

3.5.2.2 Login Screen

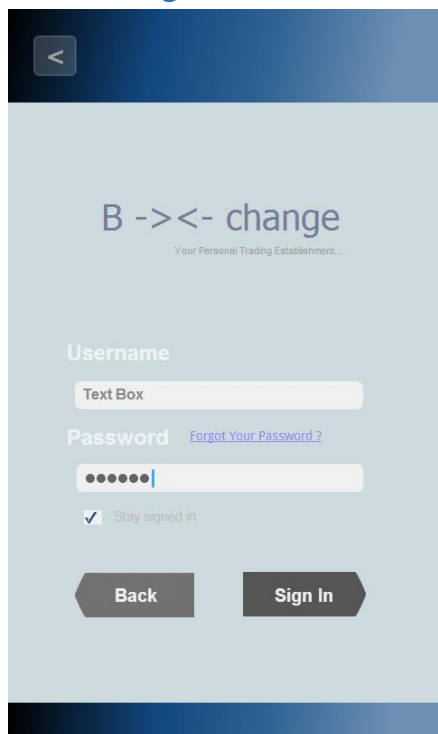


Figure 2 – Login Screen

The first step to buying and selling on Bxchange is to register as a member. This page enables users to log in or register to the Bxchange application. If users forgot their password, they can use “Forgot Your Password?” link to reset password. Moreover, by clicking take a tour button, they can use Bxchange application without login or register.

The user search stuffs without login and find something that he/she needs. At this moment, he/she can go to this page to login B-xchange application.

3.5.2.3 Register Screen

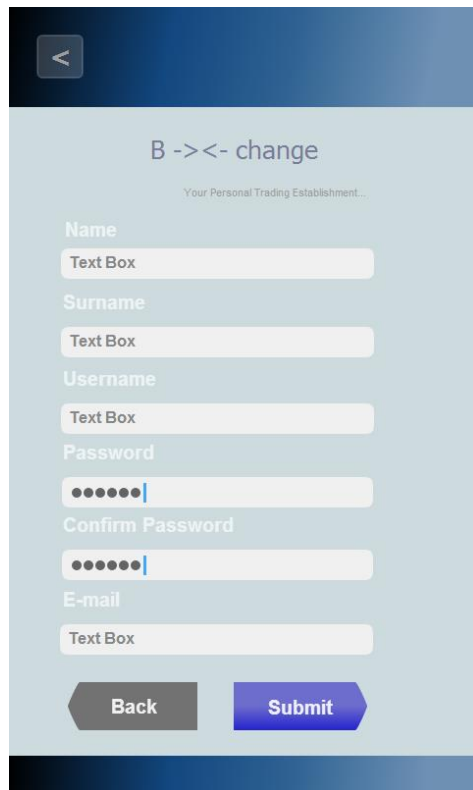
The image shows a mobile application register screen. At the top, there is a dark blue header with a white back arrow icon. Below the header, the title "B -><- change" is displayed in a light blue font, with the subtitle "Your Personal Trading Establishment..." in a smaller, lighter blue font. The form consists of several text input fields: "Name", "Surname", "Username", "Password", "Confirm Password", and "E-mail". Each field has a light gray border and a "Text Box" label inside. The "Password" and "Confirm Password" fields have a series of black dots for masking. At the bottom of the form, there are two buttons: a dark gray "Back" button and a blue "Submit" button. The entire form is set against a light blue background.

Figure 3 – Register Screen

If the user is not registered, he/she can register through this page by filling some personal information. And then the user manage his/her own profile page.

3.5.2.4 Main Page

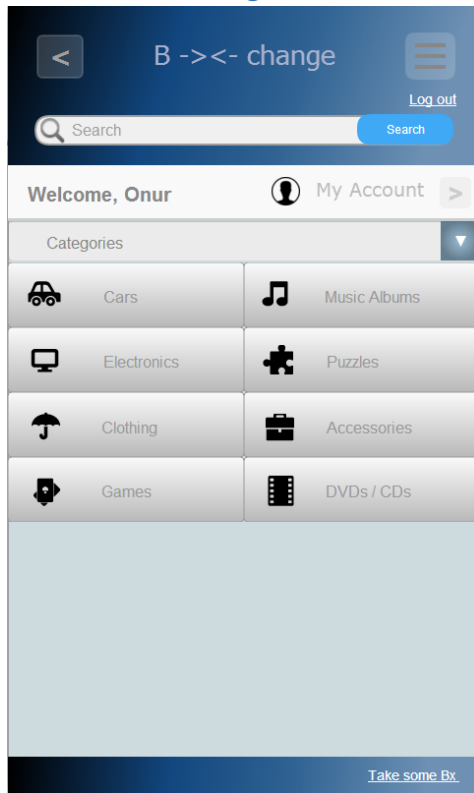


Figure 4 – Main Page (Guest)

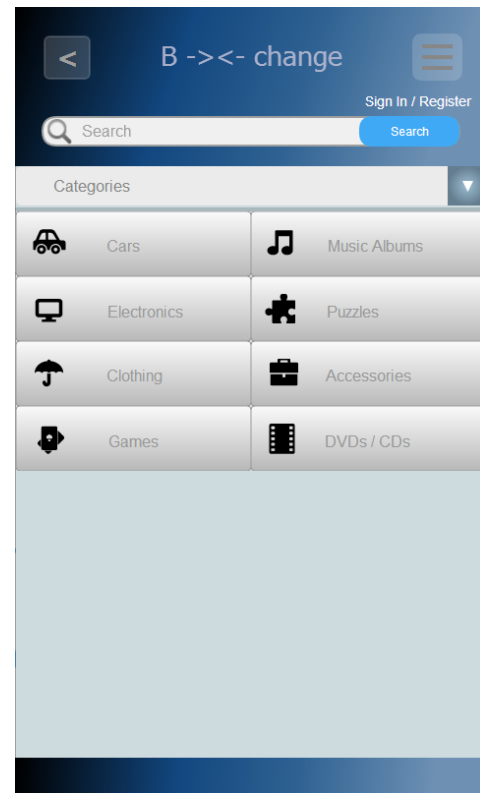
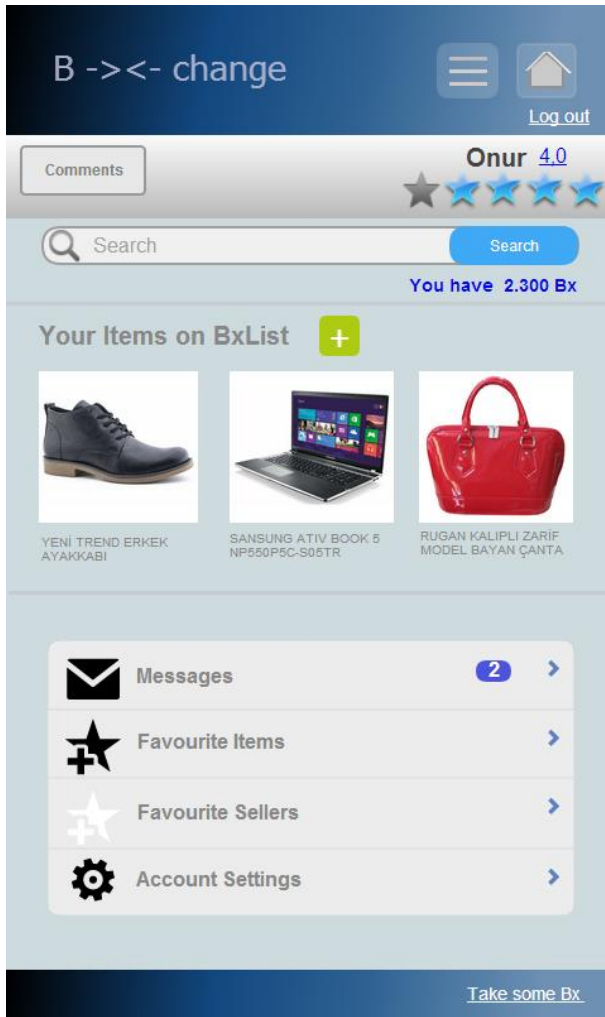


Figure 5 – Main Page (login)

From this page user find a specific item quickly, enter 2 or 3 descriptive words into the search field, and he/she will receive search results for items that match those words. If the user is not sure what he/she is looking for, or if he/she just want to explore the variety of items on B-xchange, you can browse B-xchange's listings. Browse from this page by clicking a category link. If the user login before, he/she will see left main page screen.

3.5.2.5 Profile Page



Users can see their items, rating, having Bx-coins. They can manage items by touching green plus button to go items list page. So they can add, edit or delete his/her items. Furthermore, they can reach comments, messages, favorite items, favorite sellers and configure account settings.

Figure 6 – Profile Page

3.5.2.6 Search Result Screen

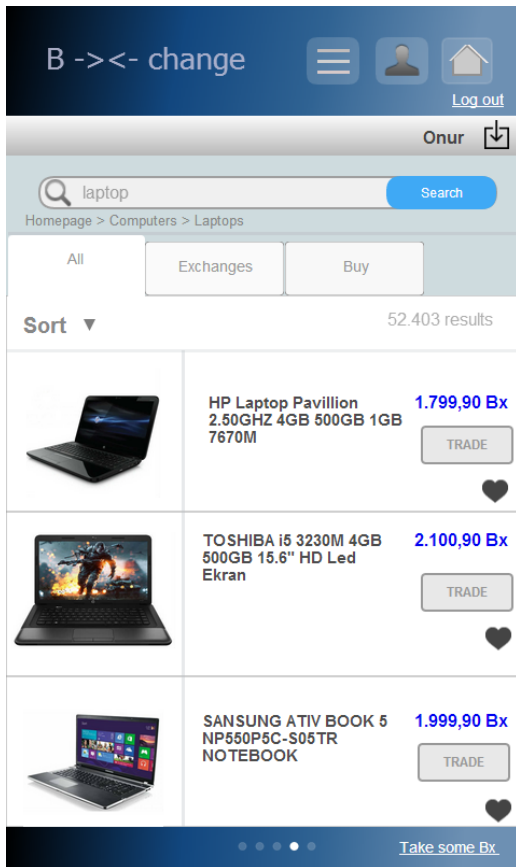


Figure 7 – Search Result Page (login)

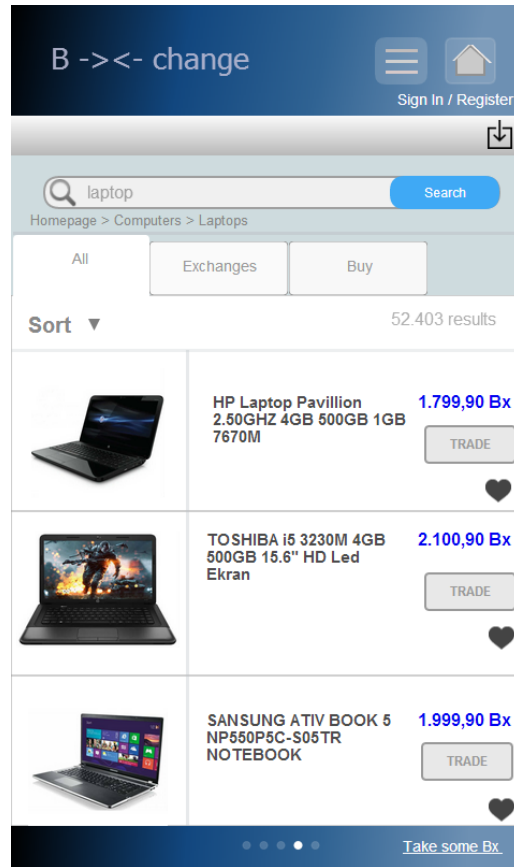


Figure 8 – Search Result Page (Guest)

This page lets users write a words and search for it in the database. Search results will be shown in the screen (3 items per page) and users can swipe left and right to see the rest of results. Moreover, they can sort results according to price low, price high, publish date etc. Users can also filter results by changing result options (all, exchange and buy) and they can see number of results and hierarchy. If guest user wants to click item or buttons (favorite, trade), they will direct to register page in figure 3.

3.5.2.7 Product Description Page

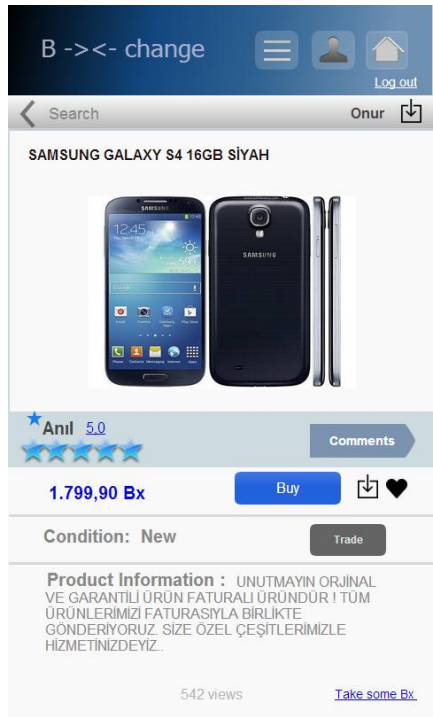


Figure 9 – Product Description Page

3.5.2.8 Favorite Items Page

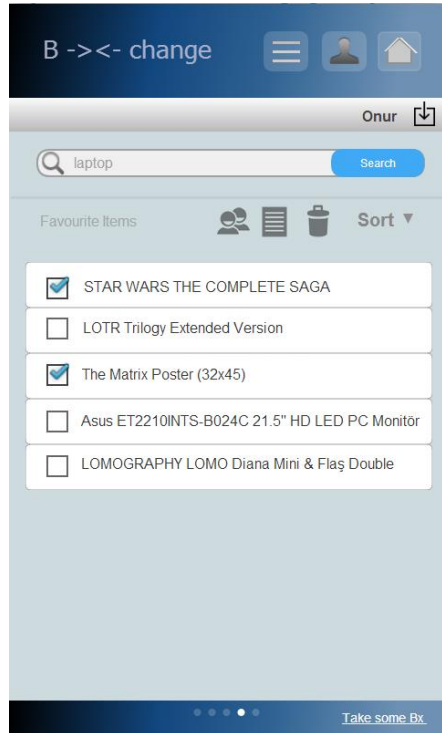


Figure 10 – Product Description Page

In Product Page, name and brief description of the item can be seen, as well as an option to add the item to the cart or favorite items. Users can also touch comments button and can take information about the seller by reading other users' comments or touch the seller's name and view his/her profile page. If the user wants to buy item with bx-coins, he/she can touch buy button. Moreover, he/she can offer his/her own item to the seller to exchange with this item by touching trade button. If the user wants to go other pages, he/she can use home, my profile, search buttons.

The user can see and manage favorite items in favorite items page. He/she can select one or more items and delete them. Furthermore, He/she can sort items according to names and also change viewing touch to list button or contact with seller. If you wants to item's description, He/she can go to product page by touching item's name. If the user wants to go other pages, he/she can use home, my profile, search ore cart buttons.

3.5.2.9 Favorite Sellers Page

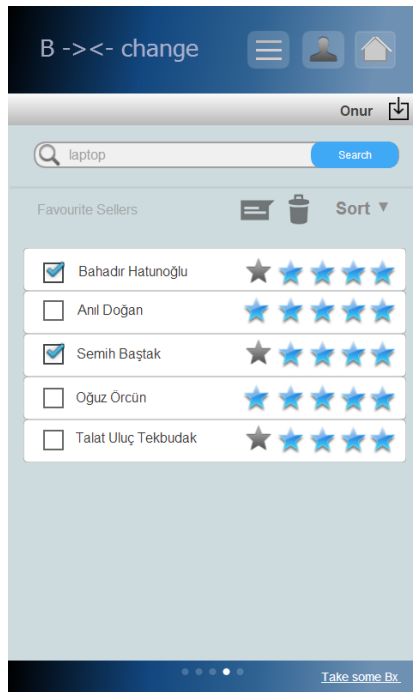


Figure 11 – Favorite Sellers Page

3.5.2.10 Messages Page

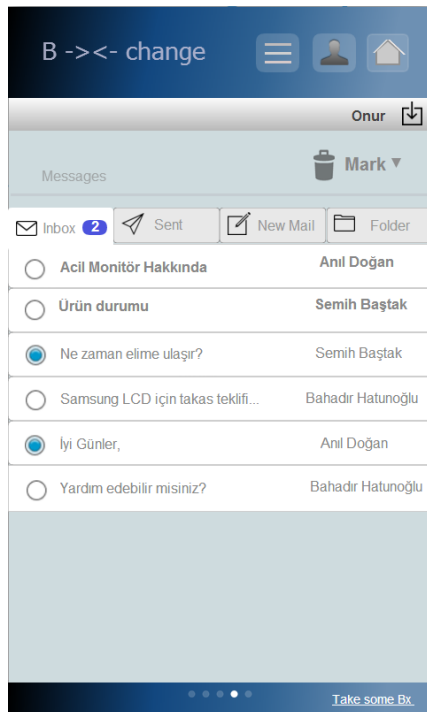


Figure 12 – Messages Page

The user can see and manage favorite sellers in this page like a favorite items page. He/she can select one or more sellers and delete them. Furthermore, He/she can sort items according to names or ratings. If the user wants to contact with one, he/she select and then touch the message button. If the user wants to go other pages, he/she can use home, my profile, search or cart buttons.

The user can contacts with other users by using personal message box. He/she can see new message in Inbox and also see sent messages in Sent. He/she can select one or more items and delete them. Moreover, he/she wants to keep important messages in folder. If the user wants to go other pages, he/she can use home, my profile, search or cart buttons.

3.5.2.11 Shopping Cart Page

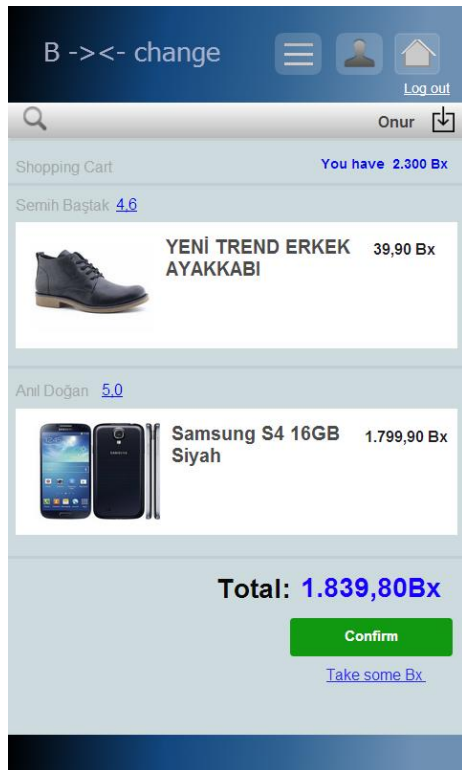
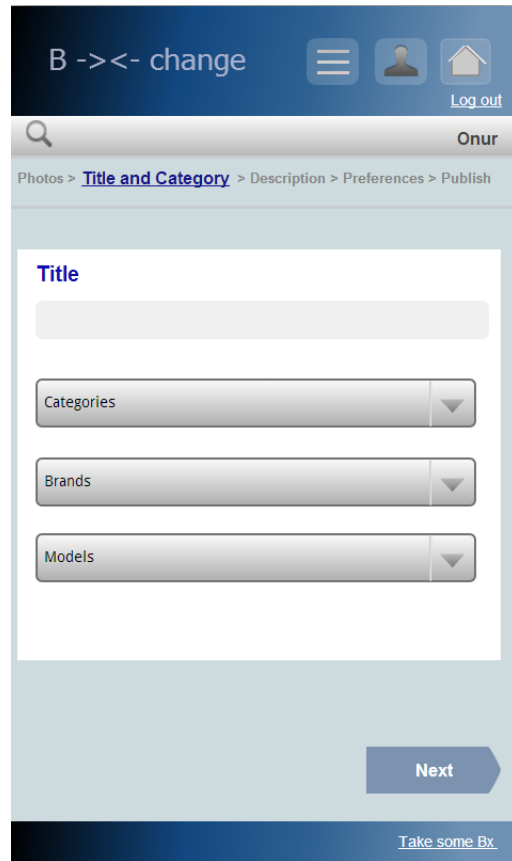
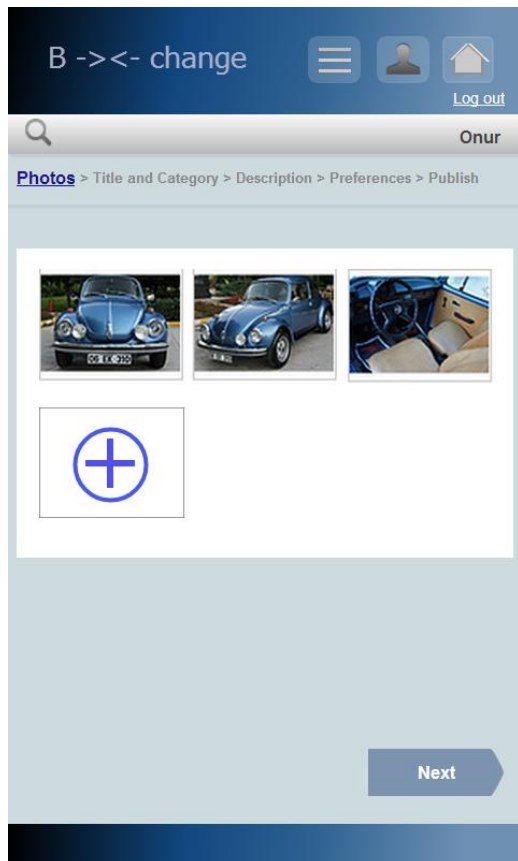


Figure 13 – Shopping Cart Page

When the user decide to finish shopping, he/she touch the shopping cart button which is near the username to direct to shopping cart page. In this page, he/she can see all items which are in shopping cart and total bx-coins. If the user touch the confirm button, he/she will accept exchanging items with bx-coins. The user have to check his/her bx-coins. If his/her bx-coins are not enough for accepting shopping, he/she can buy more bx coins by touching “Take some Bx”.

3.5.2.12 Publish Page



B -><- change

Log out

Onur

Photos > Title and Category > **Description** > Preferences > Publish

Condition Item

Item Specifics

Description

Next


B -><- change

Log out

Search

Onur

PARS CLASSIC-TÜRKİYEDE BÖYLE BİR TOSBA DAHA OLDUĞUNU SANMIYORUM



Anıl 5.0

Comments

1.839,80 Bx

Buy

Condition: Used

Trade

Product Information : ARACIN KAPI DÖŞEMELERİ TAVAN DÖŞEMESİ KOLTUK DERİLERİ TABAN HALILARI GİBİ İÇ AKSAMI ARA KONSOL VE TORPIDOSU FABRİKADAN ÇIKTIĞI ORJİNAL EKİPMANLARI OLUP TOPLAMA VEYA SONRADAN YAPILMA DEĞİLDİR

SUBMIT

B -><- change

Log out

Onur

Photos > Title and Category > Description > **Preferences** > Publish

Format

☒ Exchange ☒ Buy 1.799,90 Bx

Preferences

Next

The Wizard will allow you to publish new item for trades step by step. In the first step, the user can upload item photos. In the second step, he/she write title and choose category, brand and model. If he/she choose wrong category, the item may not visible in search result properly. In the third step, he/she specify item's condition, item specifics and description. In the fourth

step, he/she choose format, write bx-coins price and also write preferences. In the last step, he/she can see last version and submit.

3.5.2 Screen Objects and Actions

Page	Button	Action
Welcome Page	Sign In	Directs to Main Page when valid user name and password are entered
	Register	Directs to Register Page
	Take a Tour	Directs to Main Page as guest
Login Page	Sign In	Directs to previous page when valid user name and password are entered
	Back	Directs to previous page
Register Page	Submit	Directs to previous page when user fills forms completely
	Back	Directs to previous page
Main Page	Search	Directs to Search Result Page
	Cars, Electronics, Clothing, Games, Puzzles, Accessories, Music Albums, DVDs/CDs	Directs to Search Result Page in selected category
	My Account	Directs to Profile Page
	Log Out	Logs out from the application
Profile Page	Comments	Directs to Comment Page
	Favorite Items	Directs to Favorite Items Page
	Favorite Sellers	Directs to Favorite Sellers Page
	Messages	Directs to Messages Page
	Account Settings	Directs to Account Settings Page

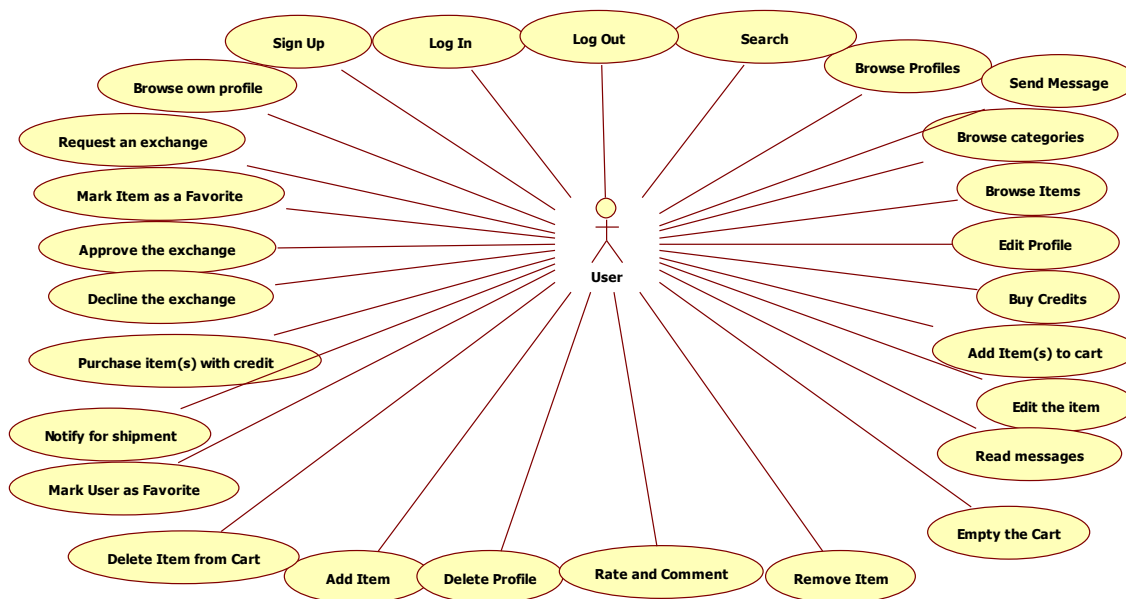
	Green Plus (Manage Items)	Directs to his/her Items List Page to add, delete and edit items
Search Result Page	Trade	Directs to Messages Page to offer items
	Favorite	Add item to favorite items list
	Sort	Sorting results according to selected
Product Description Page	Buy	Directs to confirm shopping page
	Trade	Directs to Messages Page to offer items
	Shopping Cart	Add item to shopping list
	Comments	Directs to seller's profile page
Favorite Items Page	Contact	The user can contact with seller
	View	Change favorite items page of view
	Trash	Delete selected item(s)
	Sort	Sorting items according to their names
Favorite Sellers Page	Messages	Contact with selected seller
	Trash	Delete selected seller(s)
	Sort	Sorting sellers according to their names or ratings
Messages Page	Trash	Delete selected message(s)
	Mark	Mark all, unread or read messages
Shopping Cart Page	Confirm	To end shopping
Publish Page	Next	Go to next process
	Submit	Accept publish

3.6 Context Viewpoint

Context viewpoint describes the relationships, dependencies, and interactions between the system and its environment such as the people, systems, and external entities with which interacts.⁹ It is applicable for all systems and its stakeholders are especially acquirers, users, and developers.

3.6.1 Use cases

The user has these user cases.



⁹<http://www.viewpoints-and-perspectives.info/home/viewpoints/context/>

3.6.1.1 Use Case: Sign Up



In this case, user enters his/her username, name, surname, password, mail address sets a profile photo. When the user fills all necessary fields, an activation mail will be sent to users mail address. User will be able to log in and start using the system just after activating his/her account.

3.6.1.2 Use Case: Login



In this case, user enters his/her username or email and his/her password and logs in the system.

3.6.1.3 Use Case: Log Out



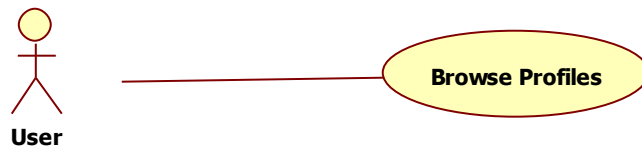
In this case, user touches log out button, and logs out of the system. The user will not use the system until he/she logs in again.

3.6.1.4 Use Case: Search



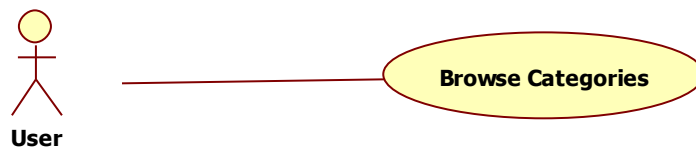
In this case, user is able to search items. User can search items according to their price, their location, their category.

3.6.1.5 Use Case: Browse Profiles



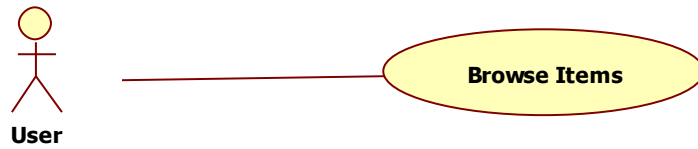
In this case, user is able to browse the profiles other users. User can see the other users' user names, profile photos, and the users' items.

3.6.1.6 Use Case: Browse Categories



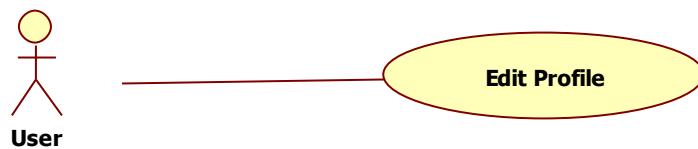
In this case, user is able to browse categories according to his/her wishes. User can also see how many items each category has.

3.6.1.7 Use Case: Browse Items



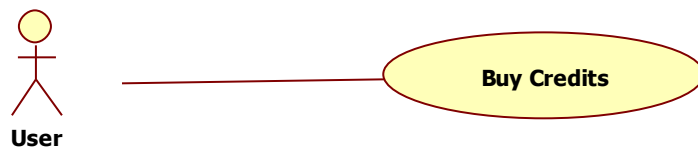
In this case, user is able browse items. User can see the items' pictures, prices, and other features.

3.6.1.8 Use Case: Edit Profile



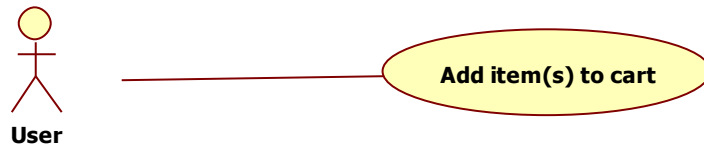
In this case, user can change his/her username, email address, profile picture and password. If the user changes his/her username or password, a notification mail will be sent to the user's email address. The changes will apply if the user clicks on the link in the email.

3.6.1.9 Use Case: Buy Credit



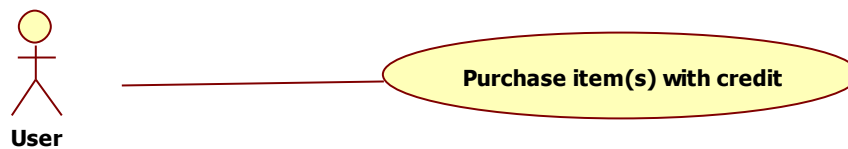
In this case, user can buy credits from the system. He/she can use credit card or his/her PayPal account.

3.6.1.10 Use Case: Add Item(s) to Cart



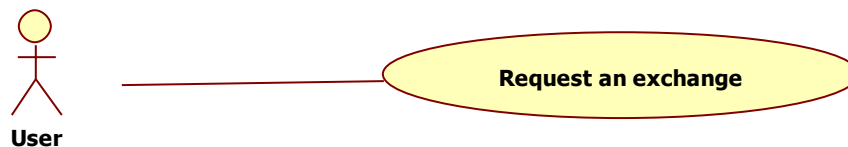
In this case, user can add the items he/she likes to his cart by touching “Add to the cart” button.

3.6.1.11 Use Case: Purchase Item(s) with Credit



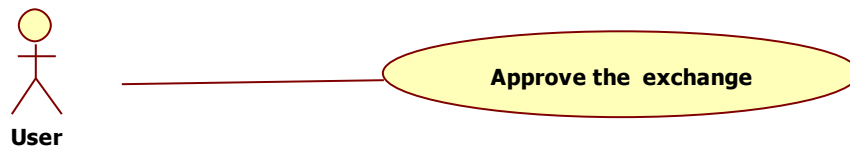
In this case, the user can buy items he/she has added to cart with credits, if the items are available for buying with credits.

3.6.1.12 Use Case: Request an Exchange



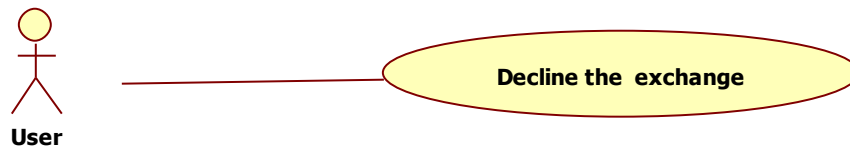
In this case, if the item the user wants is available for exchange, the user can request the exchange of items that he/she has with the item he/she wants.

3.6.1.13 Use Case: Approve the Exchange



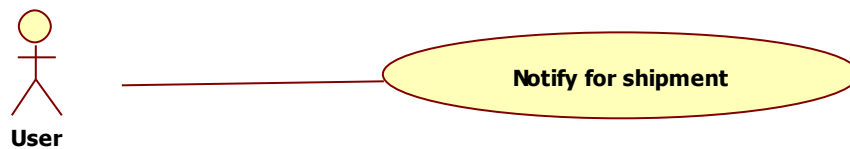
In this case, the user can approve the exchange that is offered by another user by clicking “Approve” button.

3.6.1.14 Use Case: Decline the Exchange



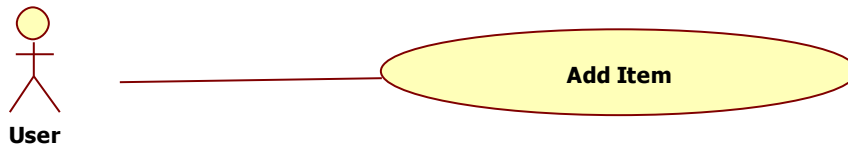
In this case, the user can decline the exchange that is offered by another user by clicking “Decline” button.

3.6.1.15 Use Case: Notify for Shipment



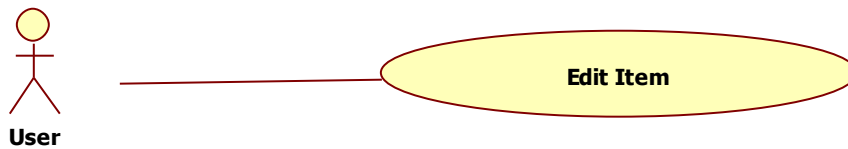
In this case, the user notifies other user that the shipment of the item has started.

3.6.1.16 Use Case: Add Item



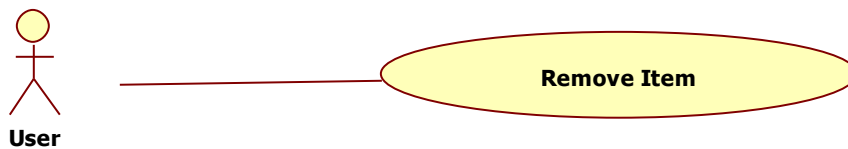
In this case, the user can add as many as items he/she wants. The user can also set the price, set the amount, upload pictures of the item and add details of the item.

3.6.1.17 Use Case: Edit Item



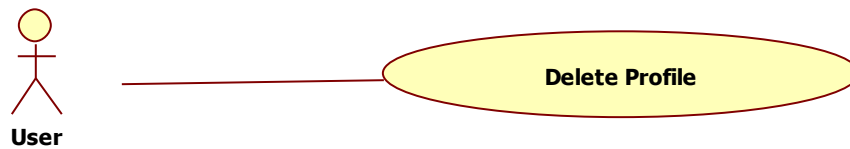
In this case, the user can edit the items that he/she has already put on the system. The user can add/remove pictures, change the price and details of the item.

3.6.1.18 Use Case: Remove Item



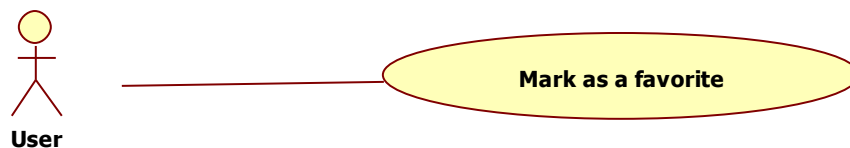
In this case, if the user no longer wishes to sell/exchange the item, he/she can remove the item from the system by touching “Remove the item” button.

3.6.1.19 Use Case: Delete Profile



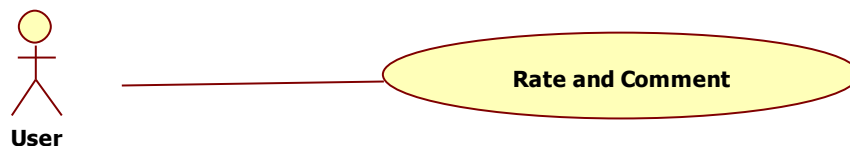
In this case, if the user no longer wishes to use the system, he/she can click “Delete Profile” button. After deleting the profile, the user will no longer be able to use the system with the account details that he entered before.

3.6.1.20 Use Case: Mark as a Favorite



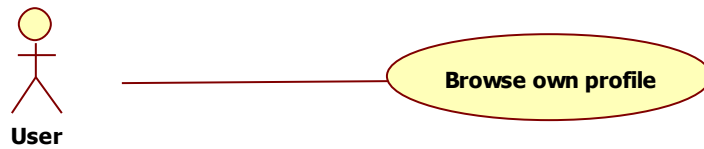
In this case, the user can mark an item as a favorite so that he will be notified if any change is made to the item

3.6.1.21 Use Case: Rate and Comment



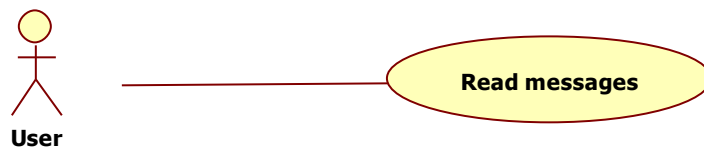
In this case, the user can rate and comment about the users whom he traded with.

3.6.1.22 Use Case: Browse user's own profile



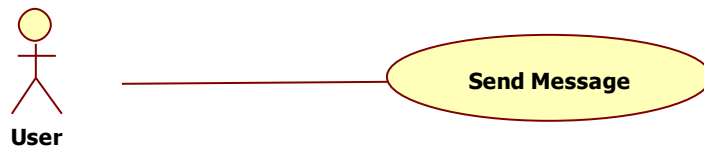
In this case, the user can view his/her own profile, the user can view his/her rating, inbox notifications, favorite items, items for sale, favorite sellers.

3.6.1.23 Use Case: Read Messages



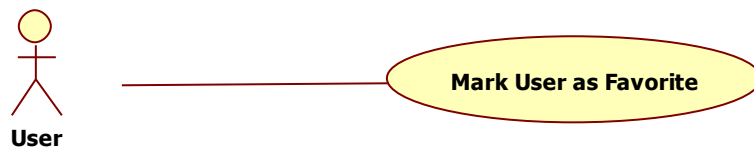
The user can read the messages sent from other users and his/her own messages sent by himself.

3.6.1.24 Use Case: Send Message



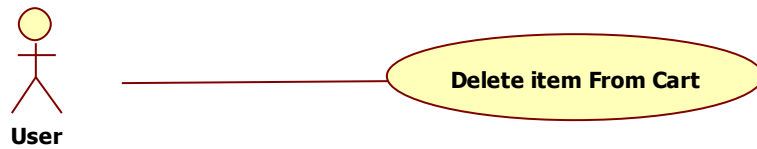
The user can send messages to other users for bargaining and for exchange requests.

3.6.1.25 Use Case: Mark User as Favorite



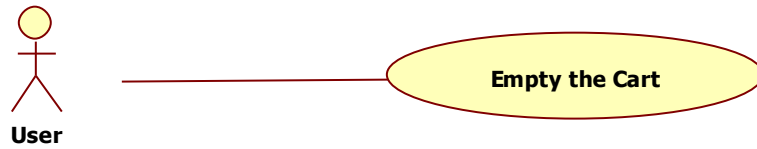
The user can mark other users as favorite. He will be able to see the changes about the favorite users.

3.6.1.26 Use Case: Delete Item from Cart



The user can delete items from his/her cart.

3.6.1.27 Use Case: Empty the Cart

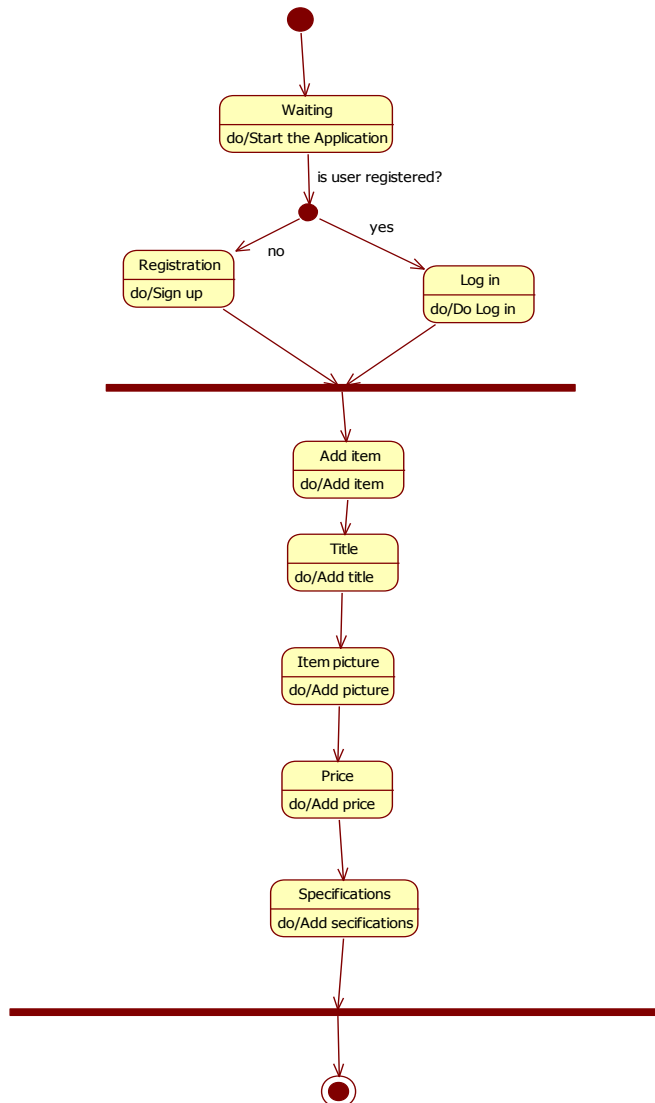


The user is able to delete all items from his/her cart with just one function.

3.7 State Dynamics Viewpoint

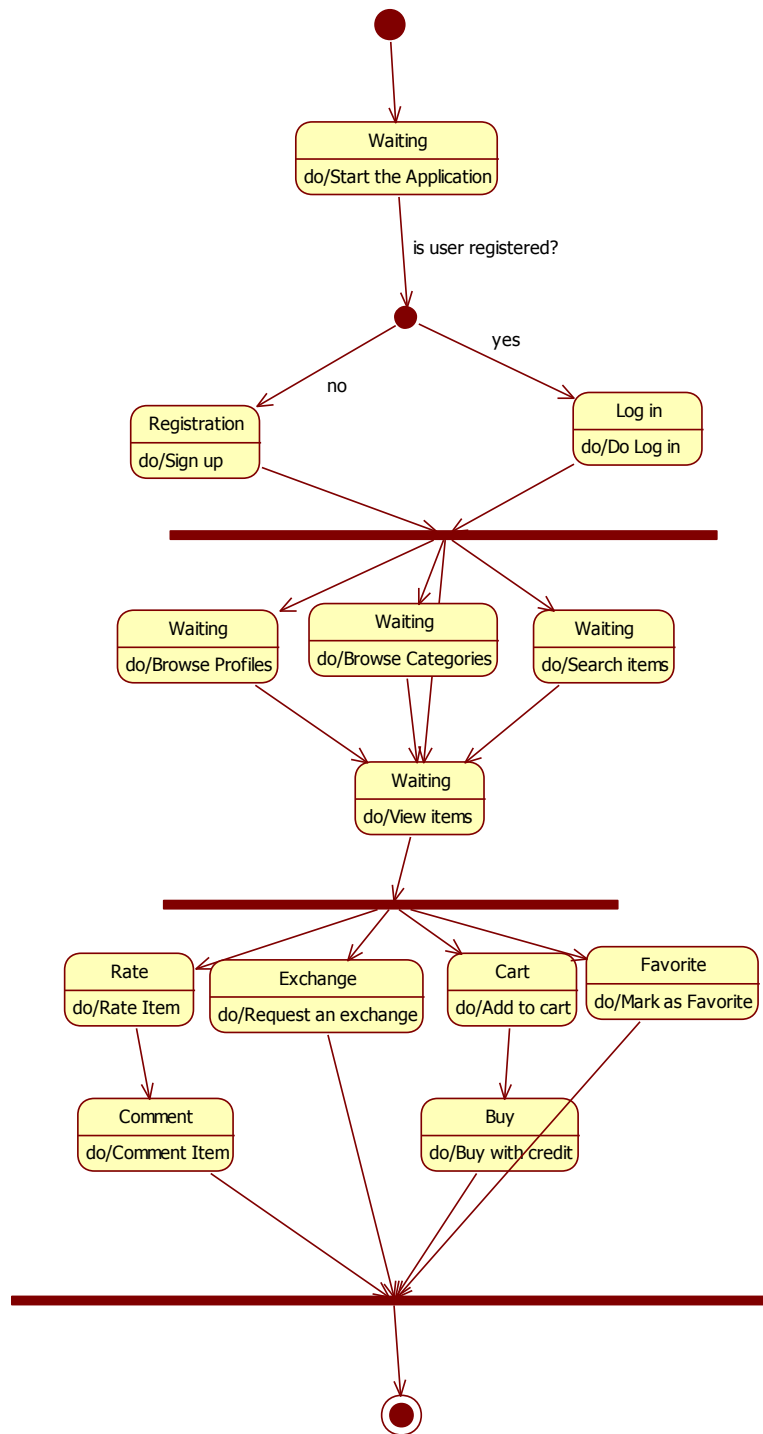
In this viewpoint, the overflow of the system is explained by successive states.

3.7.1 Adding Item



In this chart, adding item is explained. Users starts the application. If the user has already registered the system, the user will log in. If the user is already logged in, the user will add an item. The user will add items in this way: The user will add title first, and then picture, price and finally specifications.

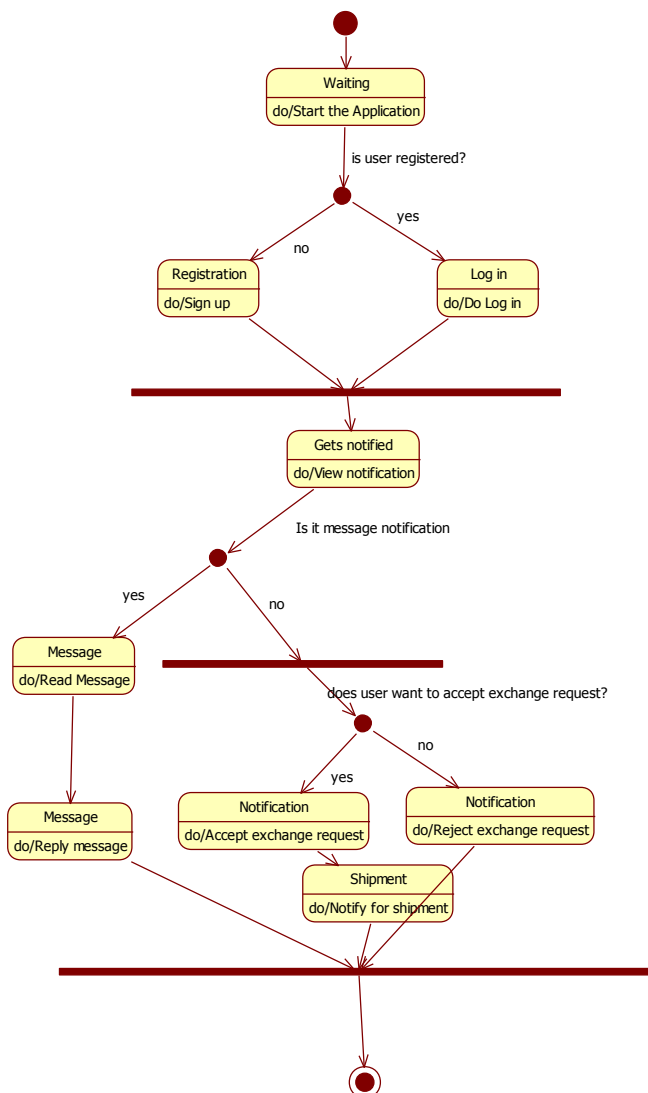
3.7.2 Viewing, Rating, Commenting, Adding to Cart and Marking items as Favorite



In this chart, the user will register or log in the system, according to his/her current situation. The user is able to browse categories, and other users' profiles. The user will be able to see what items other users have, the items in their categories and so on. The user can rate and comment on items.

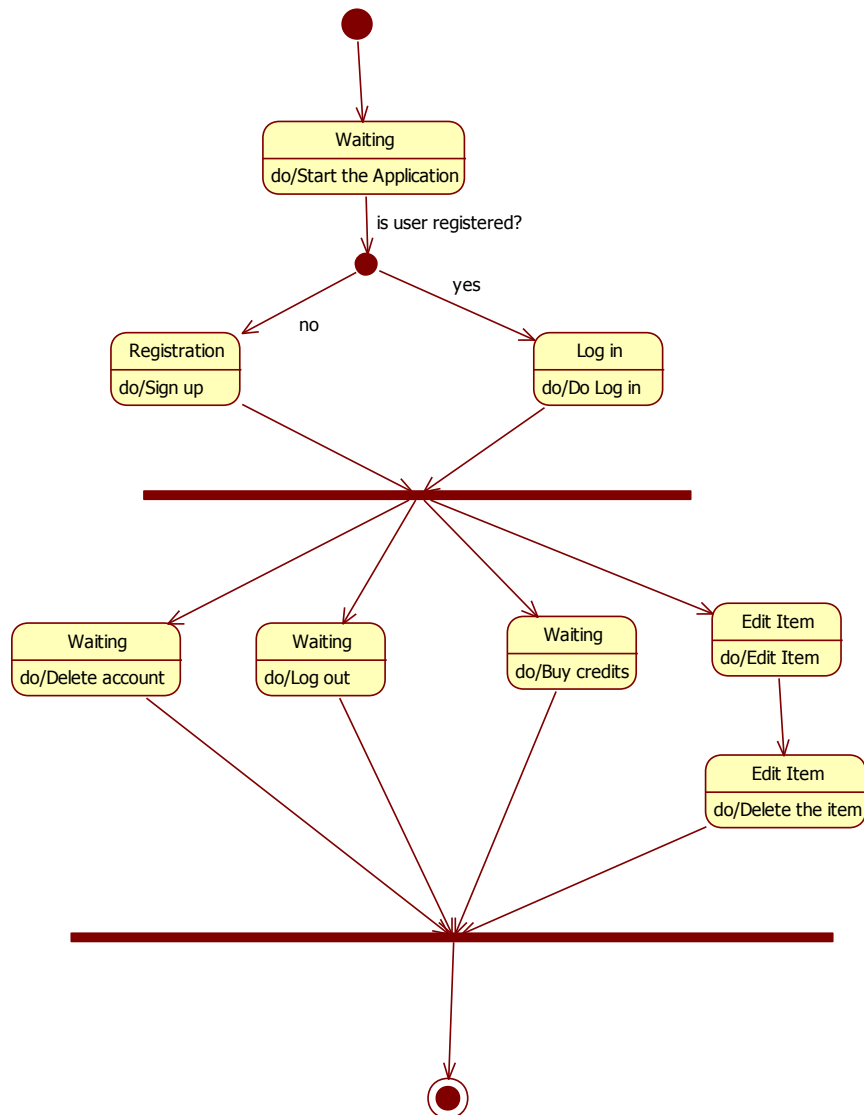
The user can also make a request to exchange an item. The user can add items to cart and buy the items with credits that the user already has on the system. The user can also add an item to his/her favorites. If any change happens to the item, the user will be notified.

3.7.3 Accepting and Rejecting Exchange Request



The user gets notified when he receives a request to exchange items. The user can approve or reject the request. If a user approves the request, the user will notify other user for shipment.

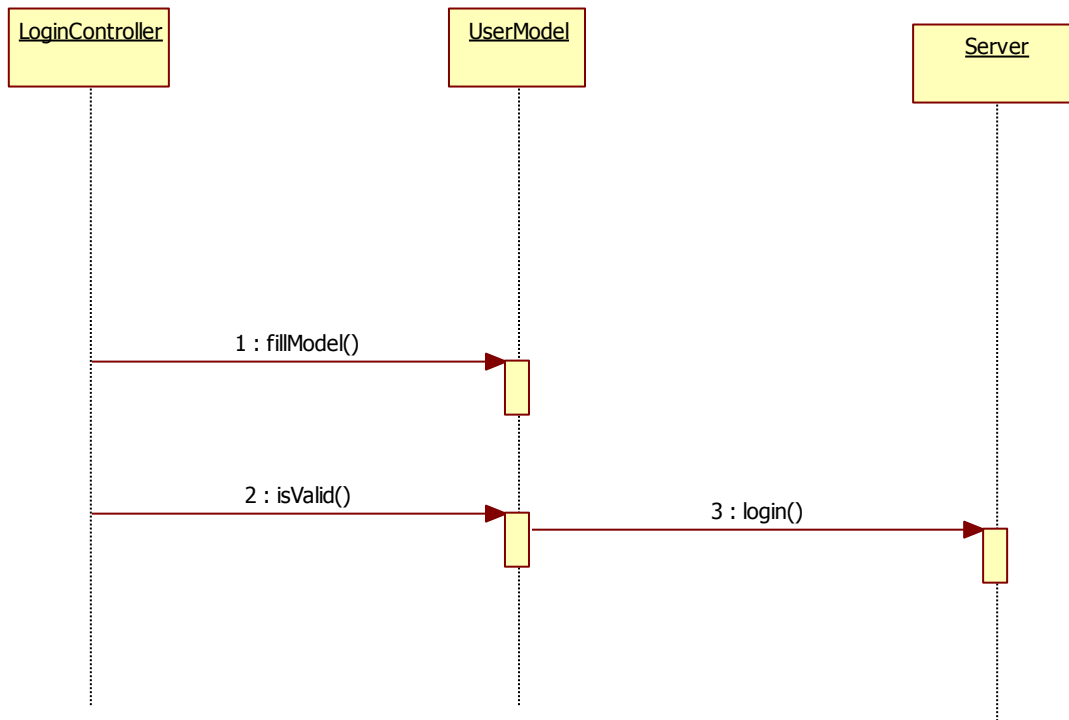
3.7.4 Deleting Account, Logging Out, Editing Item, Buying Credits



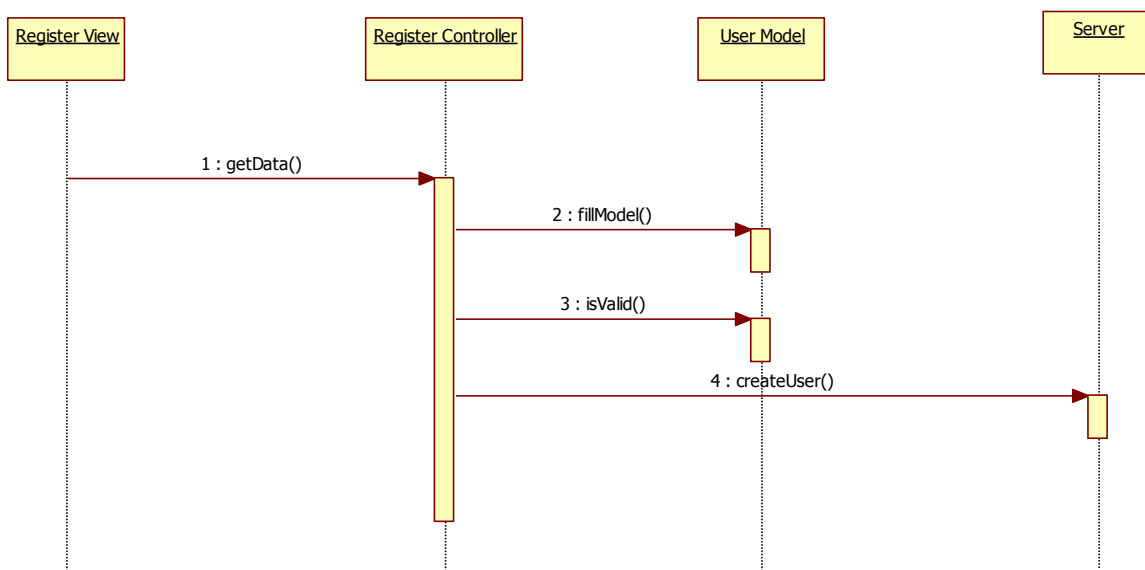
The user can log out or delete his/her account if she/he doesn't want to use the service anymore. The user can also buy credits to get items without making an exchange. The user is also able to edit the items that she/he owns. Moreover if the user can remove the item from the system if she/he no longer wishes to sell/exchange the item.

3.8 Interaction Viewpoint

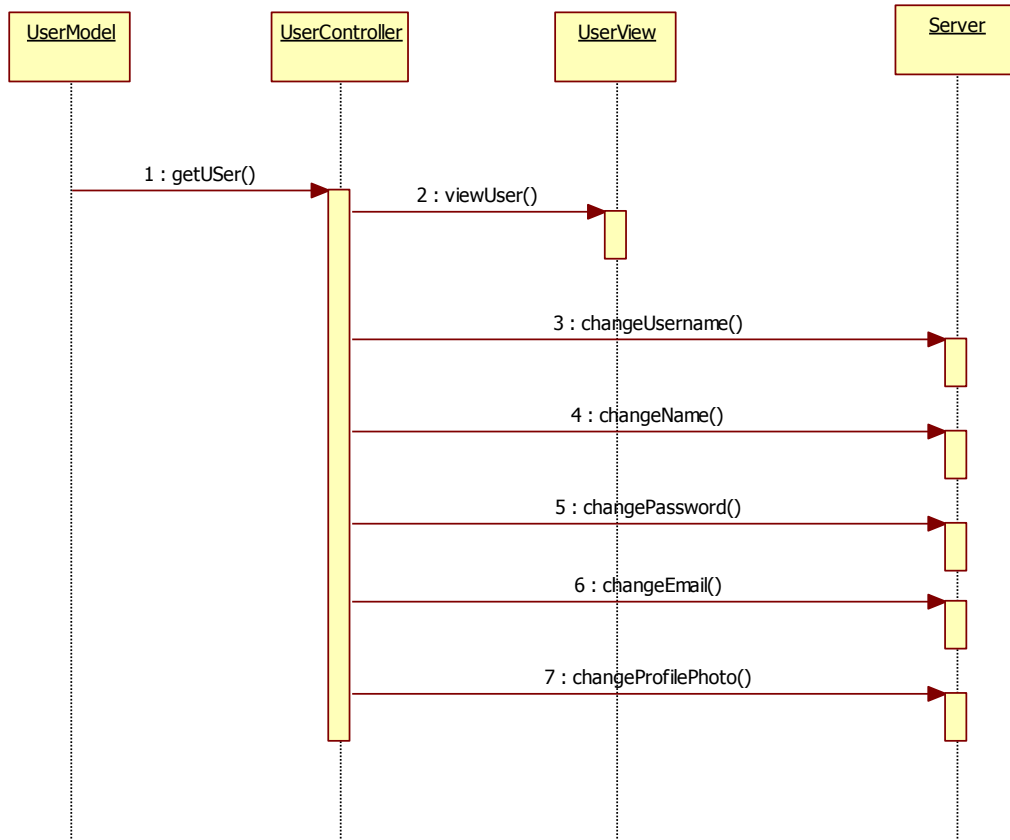
3.8.1 Login Controller



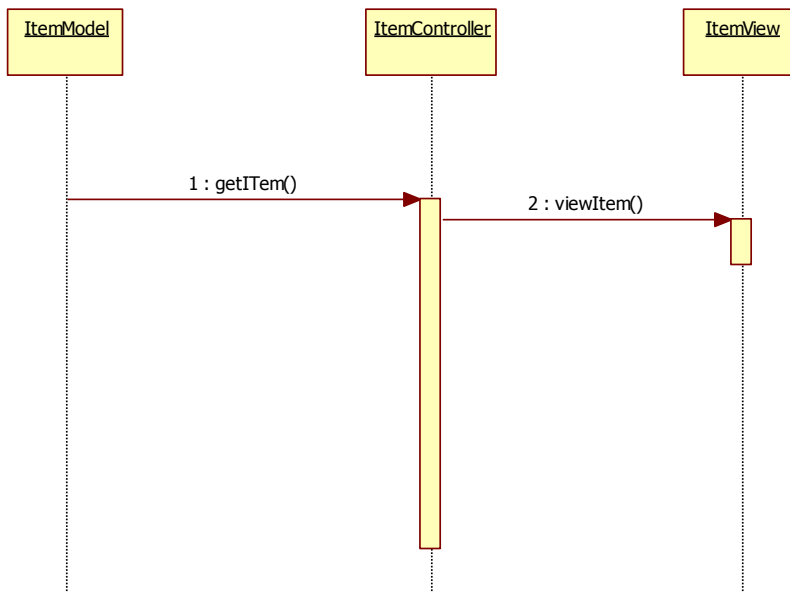
3.8.2 Register Controller



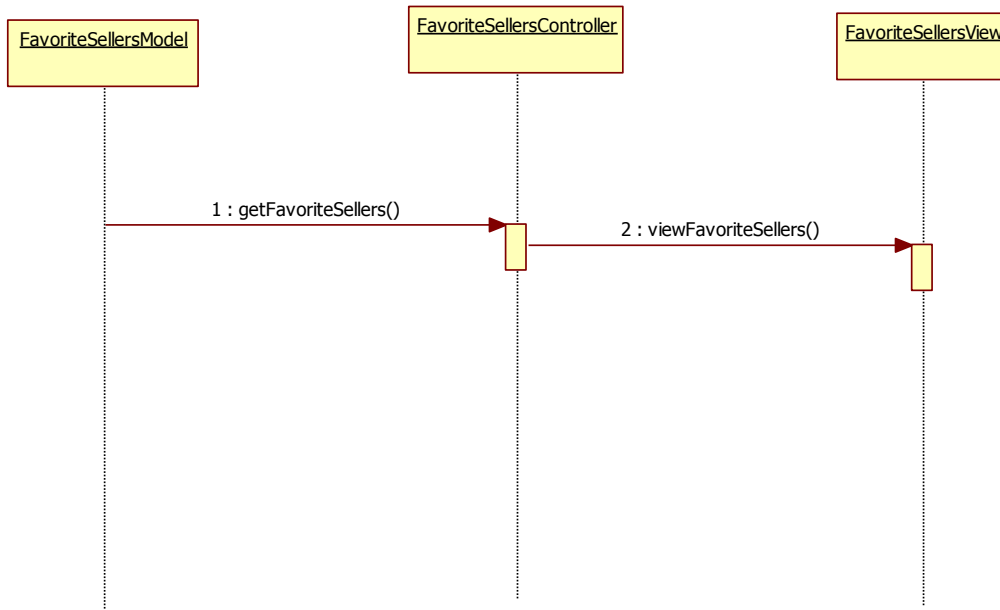
3.8.3 Profile Controller



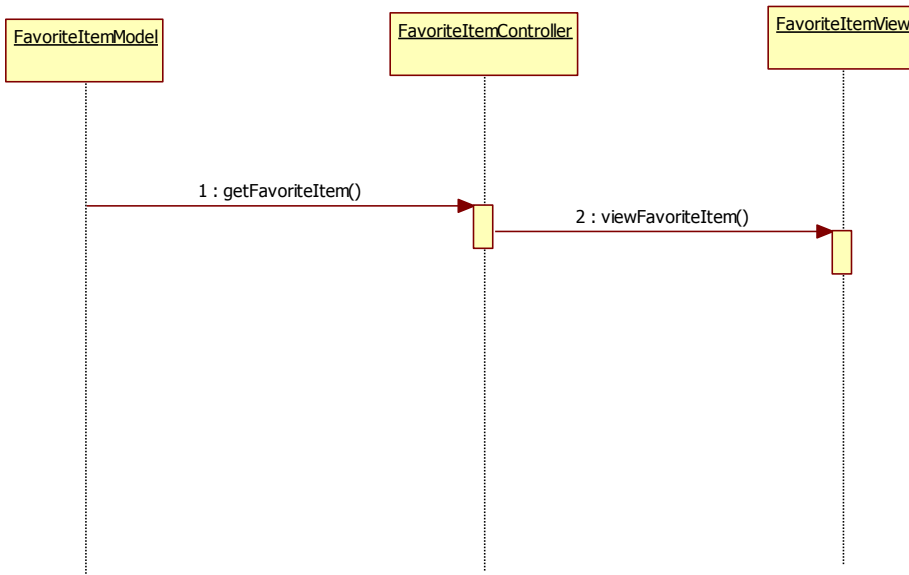
3.8.4 Item Profile Controller



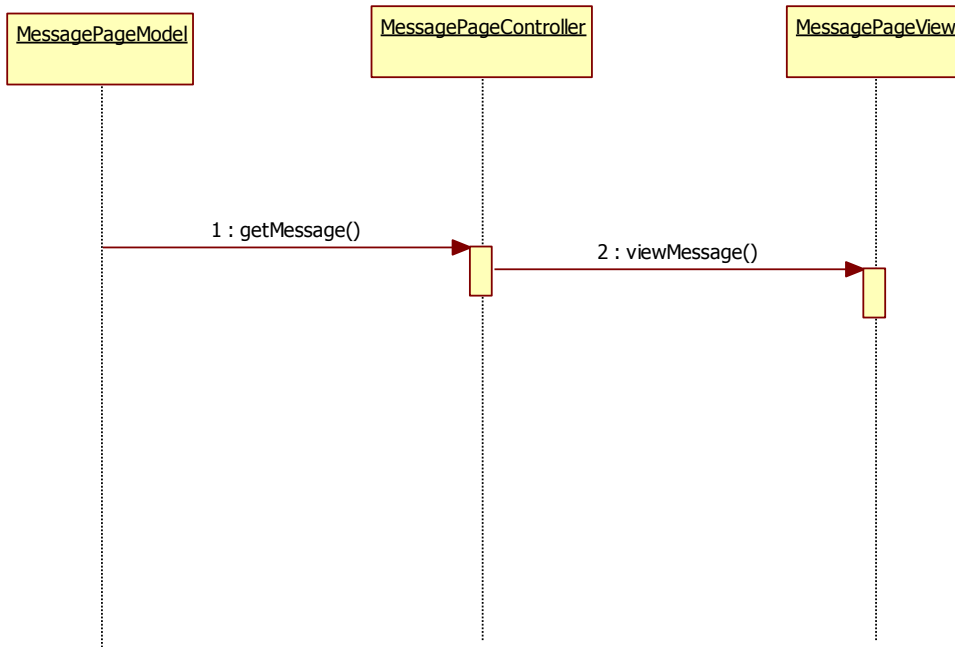
3.8.5 Favorite Seller Page Controller



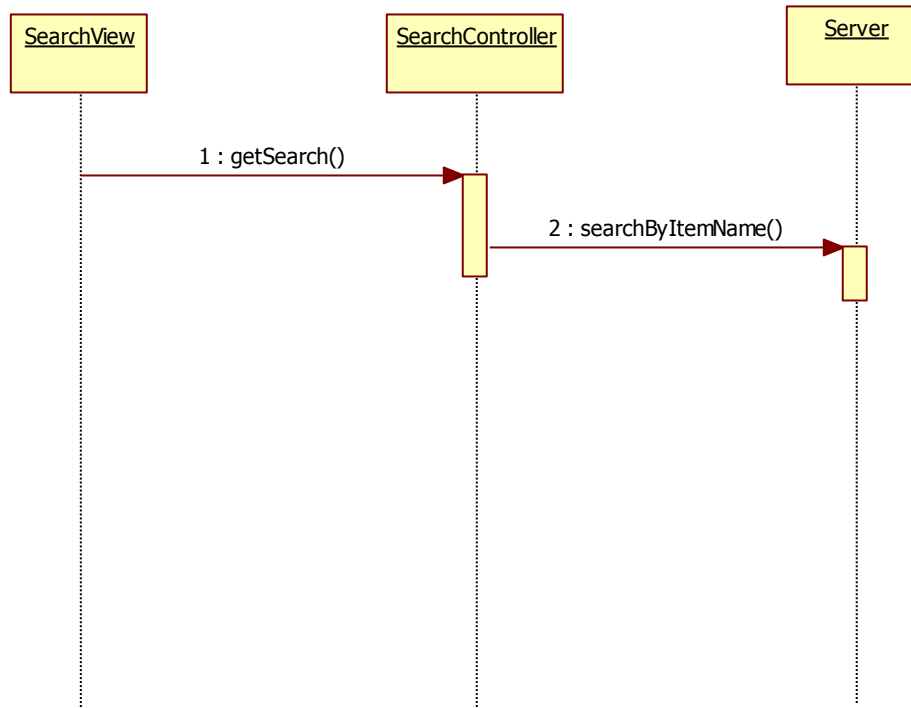
3.8.6 Favorite Item Page Controller



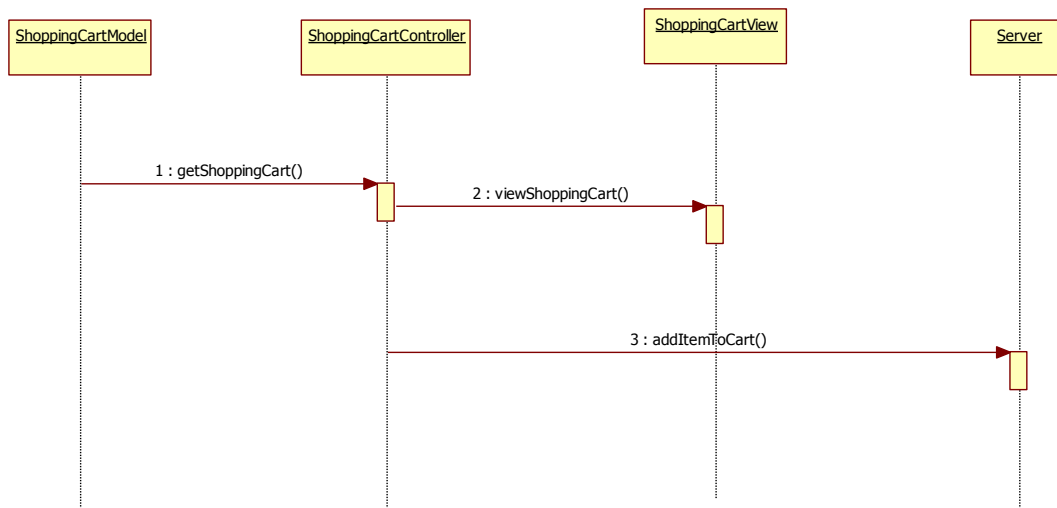
3.8.7 Message Controller



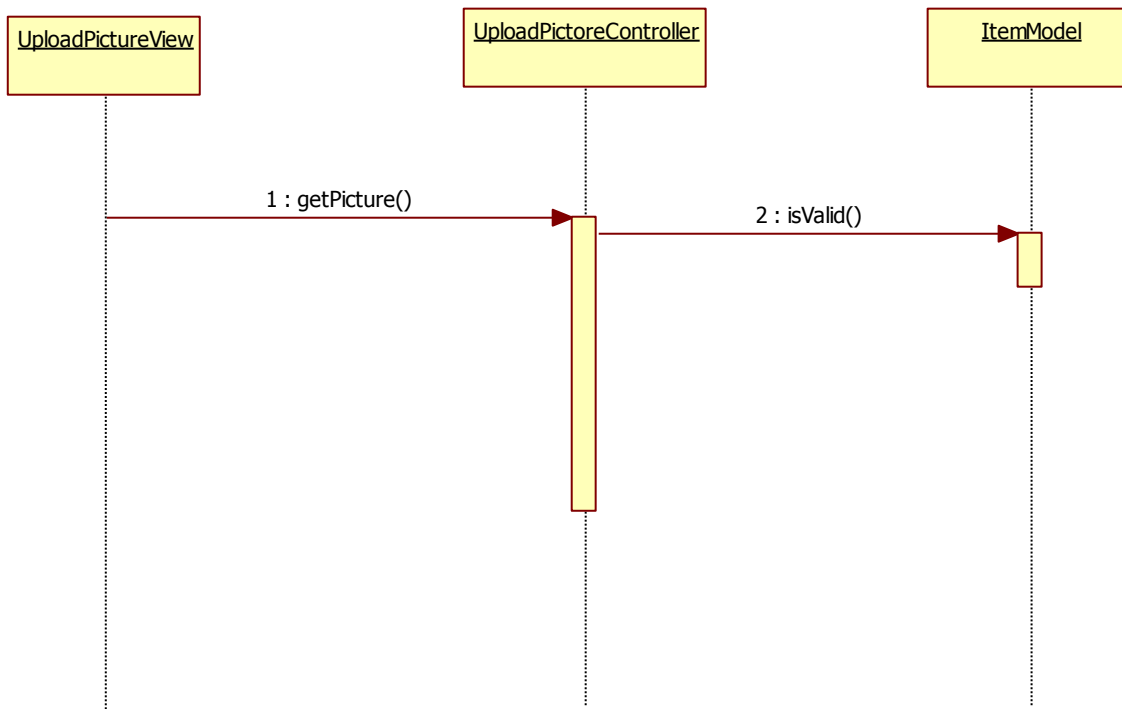
3.8.8 Search Page Controller



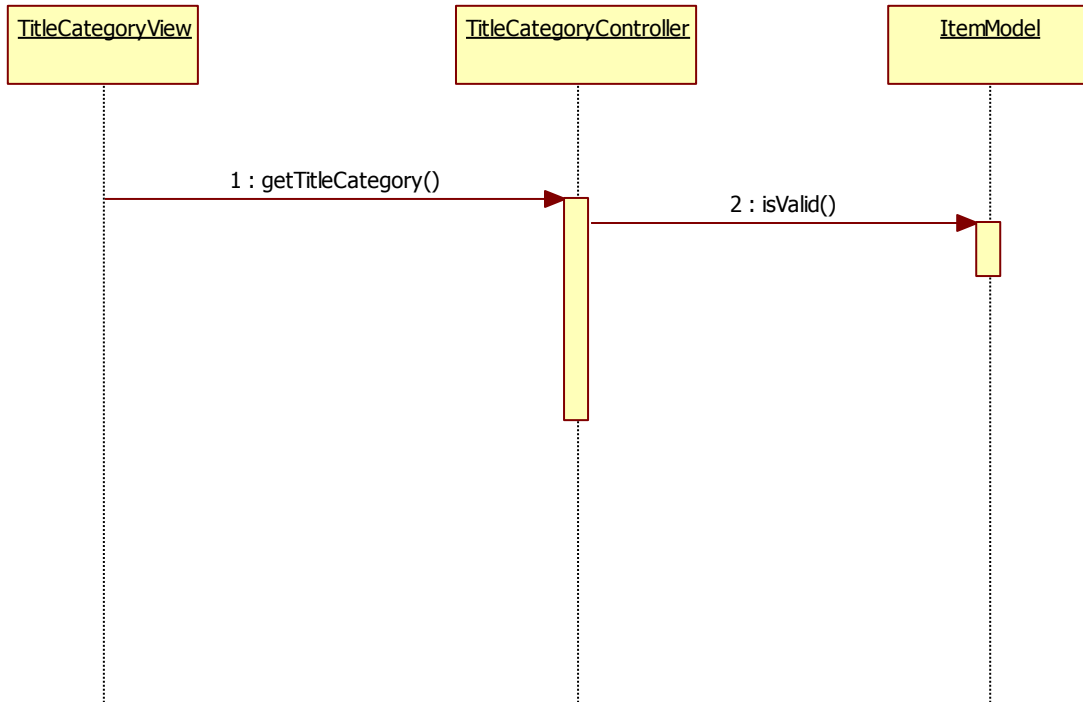
3.8.9 Shopping Cart Page Controller



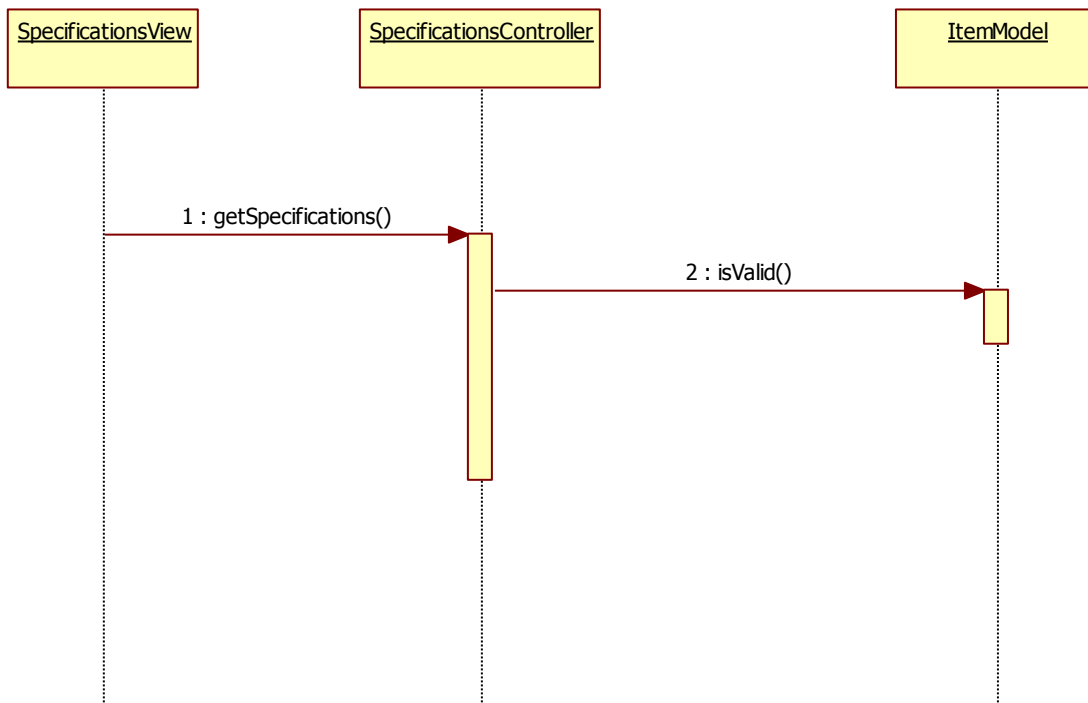
3.8.10 Picture Upload Controller



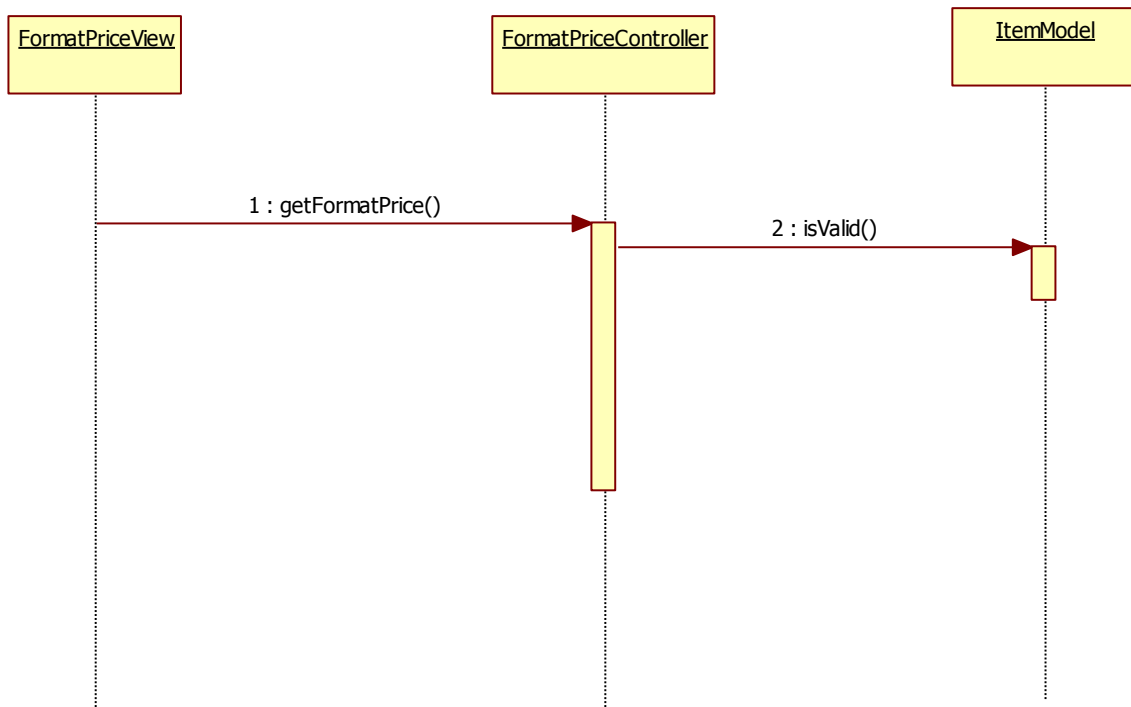
3.8.11 Title Category Controller



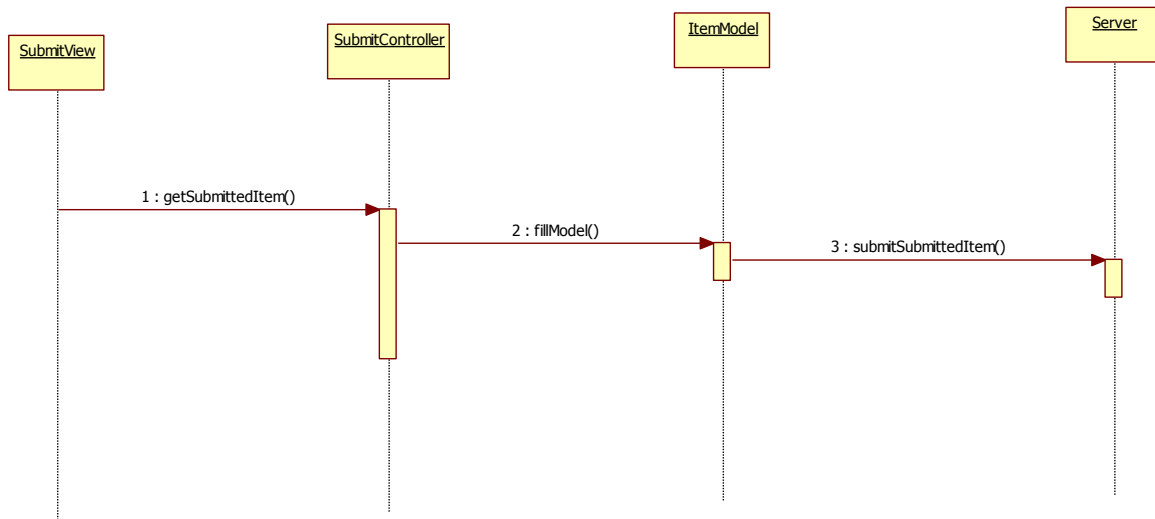
3.8.12 Specifications Controller



3.8.13 Format Price Controller



3.8.14 Submit Controller



3.9 Logical Viewpoint

In this part we explain main classes of the project. Mainly for every page that we are going to use there will be a controller and a view for that controller. In the controller the data which will be shown in view are going to be fetched by methods. After that data will be shown in the view component. Furthermore, there will be controllers for every model which going to effect the model change the database accordingly. This viewpoint interests people with knowledge on code and development of an application rather than general public.

We use model-view-controller pattern as stated on Pattern Use Viewpoint so the usage of the classes throughout the application are determined according to this pattern.

Model

Data Controller Classes: These classes manages the connections of application and data model classes. Their main purpose are filling the model classes, controlling the validation of the model classes, getting information from repository and sending data to repository.

Model Classes: These classes' purpose is keeping the general information of their respective models

Controller

Page Controller Classes: These classes manages the information of their respective views. If these classes need information of data model classes they need to get through their respective controller classes. (Such as ItemProfileController needs information on an item.)

View

This part is consist of page layout files such as xml files of each page.

3.9.1 Login Controller

This controller is for login page. This components' methods are listed below:

Method Name	Return Type	Description
login	Void	Takes data from the fields from the login view sets the fields
isValid	Boolean	Sends a request to the server to check the input password

3.9.2 Register Controller

This controller is for new users to register to the database so that they can use logged in users' special features. This components' methods are listed below:

Method Name	Return Type	Description
isValid	Boolean	Checks the fields. Returns true if they are valid. Returns false and sends information about the missing fields to the Register View they are not valid.
register	Void	Takes the data of fields from Register View and sets the fields
createUser	Void	It sends the Register Model fields to Server to create a user.

3.9.3 Profile Controller

This controller is for users to manage their profile pages. Furthermore, users can change their details via this controller. This component's methods are listed below:

Method Name	Return Type	Description
getUser	User	At the time this component initialized, it fetches the User object from Server and passes it to the view.
changePassword	Void	It sends a request to Server to change the password of the user.
changeName	Void	It sends a request to Server to change the name of the user.
changeSurname	Void	It sends a request to Server to change the username of the user.
logout	View	Ends current session of the user. Triggers LoginViewController.

3.9.4 ItemProfileController

This component is used for users to manage an item and its details. This component's methods are listed below:

Method Name	Return Type	Description
getItem	Item	At the time this component initialized, it fetches the Item object from Server and passes it to the view.

3.9.5 FavoriteSellerPageController

This component is used to manage a user's favorite users. This component's methods are listed below:

Method Name	Return Type	Description
getFavoiteSellers	UserList	At the time this component initialized, it fetches the UsersList object which contains Users that are added as favorite from Server and passes it to the view.

3.9.6 FavoriteItemPageController

This component is used to manage user's favorite items. This component's methods are listed below:

Method Name	Return Type	Description
getFavoiteItems	ItemList	At the time this component initialized, it fetches the ItemList object which contains Items that are added as favorite from Server and passes it to the view.

3.9.7 MessagePageController

This component is used to view user's masseges page. This component's methods are listed below:

Method Name	Return Type	Description
getMessage	Message	At the time this component initialized, it fetches the Message from Server and passes it to the view.

3.9.8 SearchPageController

This component is used to manage searched items as specified in input. This component's methods are listed below:

Method Name	Return Type	Description
getSearchedItem	ItemList	Takes the data of fields from Search View and sends the data to Server and fetch the specified List

3.9.9 ShoppingCartPageController

This component is used to manage items in the shopping card model and add a new item if needed.

Method Name	Return Type	Description
getShoppingCart	ItemList	At the time this component initialized, it fetches the Items from Server and passes it to the view.
addItemToCart	Void	Takes the specified item and add item in the ShoppingCartModel

3.9.10 PictureUploadController

This component is used when adding a new item to the database for uploading pictures of the item. This component's methods are listed below:

Method Name	Return Type	Description
isValid	Boolean	Checks the fields. Returns true if they are valid. Returns false and sends information about the missing fields to the PictureIpload View they are not valid.
getPicture	Void	Takes the data of fields from PictureIpload View and sets the fields

3.9.11 TitleCategoryController

This component is used when adding a new item to the database for setting title and category of the item. This component's methods are listed below:

Method Name	Return Type	Description
isValid	Boolean	Checks the fields. Returns true if they are valid. Returns false and sends information about the missing fields to the TitleCategory View they are not valid.
getTitle	Void	Takes the data of fields from TitleCategory View and sets the fields
getCategory	Void	Takes the data of fields from TitleCategoryView and sets the fields

3.9.12 CISDController

This component is used when adding a new item to the database for uploading pictures of the item. This component's methods are listed below:

Method Name	Return Type	Description
isValid	Boolean	Checks the fields. Returns true if they are valid. Returns false and sends information about the missing fields to the CISD View they are not valid.
getCondition	Void	Takes the data of fields from CISD View and sets the fields
getSpecification	Void	Takes the data of fields from CISD View and sets the fields
getDescription	Void	Takes the data of fields from CISD View and sets the fields

3.9.13 FPPController

This component is used when adding a new item to the database for uploading pictures of the item. This component's methods are listed below:

Method Name	Return Type	Description
isValid	Boolean	Checks the fields. Returns true if they are valid. Returns false and sends information about the missing fields to the FPP View they are not valid.
getFormat	Void	Takes the data of fields from FPP View and sets the fields
getPrize	Void	Takes the data of fields from FPP View and sets the fields
getPreference	Void	Takes the data of fields from FPP View and sets the fields

3.9.14 SubmitController

This component is used when finishing the adding item wizard. All of the fields are filled and this component pushes it to database via server.

Method Name	Return Type	Description
createItem	Void	It sends the Item Model fields to Server to create an item.

3.9.15 Repository Service

This component is an abstraction over actual database. It provides all methods (CRUD methods, namely insert, update, delete and fetch) for database operations. Any other component wants to access database, shall use the methods implement by Repository Service.

The remaining controller and classes are for models and their controller to manage and change details of the models.

3.9.16 User Class (Model)

Methods of this class are getter and setter of the attributes.

The attributes of this class are listed below:

Id: Identity number given from database for each user. This attribute is unique for every user.

Username: User`s nickname for the mobile application.

Name: Name of the user.

Surname: Surname of the user.

Password: The user`s password for logging in to the mobile application.

Email: Email of the user that will be used for the communication information and confirmation.

Phone_number: Phone number of the user that will be used for communication information.

Profile_photo: A photo that will be used for letting other users see the user.

Rating: The average number between 1 and 5 that will be given by other users after their trade transaction with the user, according to their satisfaction of the trade.

ReceivedMessages: The messages that are sent to the user.

SentMessaged: Messages that the user sent to other users.

FavSellers: FavoriteSeller objects that are added as favorite by the user.

FavItems: FavoriteItem objects that are added as favorite by the user.

3.9.17 UserController

This component has a User object .Methods of this component are namely, getUser which return the User specified and setUser which binds the current user to this controller.

3.9.18 Item Class (Model)

Methods of this class are getter and setter of the attributes.

The attributes of this class are listed below:

Id: id of the item. It is unique for every item.

Owner_id: Owner it of the item.

Description: A small description about the item that is intended to trade.

Photos: Photos of the specified item.

Category: Category of the item.

Location: Location information of the item.

Status: Status of the item .If the items` traded then it stays in database but does not appear in the list of items.

Price: Price of the item as amount of credits

Is_tradeable: if owner of the item open to the exchange for another item.

Item_specification: item`s specific details.

Publish_date: First date that the item entered the system.

Update_date: last update date that the item details changed.

Trade_end_date: The date that the items got traded.

Receiver_id: Receiver id of the item.

3.9.19 ItemController

This component has an Item object .Methods of this component are namely, getItem which return the Item specified and setItem which binds the current Item to this controller.

3.9.20 Messages Class (Model)

Methods of this class are getter and setter of the attributes.

The attributes of this class are listed below:

Id: Primary key given by database. This attribute is unique for every message.

Text: The information which is sent from a sender to the receiver.

Date: the date of the sent message.

Receiver_id: The user who receives the message.

Sender_id: The user who sends the message

3.9.21 MessageController

This component has a Message object .Methods of this component are namely, getMessage which return the Messages specified and setMessage which binds the current message to this controller.

3.9.22 Comment Class (Model)

Methods of this class are getter and setter of the attributes.

The attributes of this class are listed below:

Sender_id: Writer id of the comment

Intended_id: Receiver id of the comment

Comment: Text that contains comment

3.9.23 CommentController

This component has a Comment object .Methods of this component are namely, getComment which return the Comment specified and setComment which binds the current Comment to this controller.

3.9.24 FavoriteItem Class (Model)

Methods of this class are getter and setter of the attributes.

The attributes of this class are listed below:

Fav_owner_id: User id of favorite action

Fav_item_id: Item id of favorite action

3.9.25 FavoriteItemController

This component has a FavoriteItem object .Methods of this component are namely, getFavoriteItems which return the FavoriteItems specified and setFavoriteItems which binds the current favoriteItem to this controller.

3.9.26 FavoriteSeller Class (Model)

Methods of this class are getter and setter of the attributes.

The attributes of this class are listed below:

Fav_owner_id: User id of favorite action

Fav_seller_id: Item id of favorite action

3.9.27 FavoriteItemController

This component has a FavoriteSeller object .Methods of this component are namely, getFavoriteItems which return the FavoriteSeller specified and setFavoriteSellers which binds the current FavoriteItem to this controller.

3.9.28 ShoppingCart Class (Model)

This component has list of items that are added to this model Methods are getShoopingCart and setShoppingCart.

3.9.29 ShoppingCartController

This component has a ShoopingCart object .Methods of this component are namely, getShoopingCart which return the ShoopingCart specified and setShoopingCart which binds the current ShoppingCart to this controller.

4 PROJECT PLANNING

4.1 Team Structure

Our team named Epic Fail with members Anıl Doğan, Bahadır Hatunoğlu, Onur Özkan and Semih Baştak. Distribution of tasks as follows:

Name Surname	Student Id	Job	Task(s)
Anıl Doğan	1678846	Project Manager / Team Mentor	<ul style="list-style-type: none">- Manage team members- Research about bottlenecks- System design- Organizing weekly work/meeting plan(s)- Code development (Server Side / Android)
Bahadır Hatunoğlu	1560259	Software Developer	<ul style="list-style-type: none">- Code development (Server Side / Android)- Research- Database>Server>Phone Connection
Onur Özkan	1679505	User Interface Designer/ Developer	<ul style="list-style-type: none">- Designing user interfaces- Generating user interfaces- Binding interfaces to program- Code development (Server Side / Android)- Research

Semih Baştak	1678796	Software Developer	<ul style="list-style-type: none"> - Code development (Server Side / Android) - Database generation - Research
--------------	---------	--------------------	---

4.2 Estimation (Basic Schedule)

Project (SRS):	30 October 2013
Project (SDD):	1 December 2013
Prototype demo:	27 January 2014
Project (STD):	Second Semester
Project (STR):	Second Semester

4.3 Process Model

We are using waterfall process model. In general, this model may be considered as having six different phases:

1. Requirements Analysis
2. Design
3. Implementation
4. Testing
5. Installation
6. Maintenance

5 REQUIREMENTS MATRIX

Use Cases	Matching Component
3.6.1.2 Login	3.9.1 Login Controller
3.6.1.3 Log Out	3.9.1 Login Controller
3.6.1.4 Search	3.9.8 SearchPageController
3.6.1.5 Browse Profile	3.9.3 Profile Controller
3.6.1.6 Browse Categories	3.9.8 SearchPageController
3.6.1.7 Browse Items	3.9.8 SearchPageController
3.6.1.8 Edit Profile	3.9.3 Profile Controller
3.6.1.9 Buy Credit	3.9.17 UserController
3.6.1.10 Add Item(s) to Cart	3.9.29 ShoppingCartController
3.6.1.11 Purchase Item(s) with Credit	3.9.29 ShoppingCartController
3.6.1.12 Request an Exchange	3.9.19 ItemController
3.6.1.13 Approve the Exchange	3.9.19 ItemController
3.6.1.14 Decline the Exchange	3.9.19 ItemController
3.6.1.15 Notify for Shipment	3.9.17 UserController
3.6.1.16 Add Item	3.9.19 ItemController
3.6.1.17 Edit Item	3.9.19 ItemController
3.6.1.18 Remove Item	3.9.19 ItemController
3.6.1.19 Delete Profile	3.9.17 UserController
3.6.1.20 Mark as a Favorite	3.9.6 FaviriteItemController
3.6.1.21 Rate and Comment	3.9.23 CommentController