Cairo University

Faculty of Computers and Information

CS251

Software Engineering I

Valley e-Commerce Platform

Software Requirements Specifications V1.0

Team: Pied piper

Oct 2017

Contents

Team

ID	Name	Email	Mobile
20150309	Sherif Mohamed Abd El Naby	SherifAbdlNaby@gmail.com	01001272160
20150204	Mohamed Ashraf Abd El Hady	Mohamed1refaie@hotmail.com	01113307095
20150101	Khaled Ahmed Hamed	Khaled.hamedt@gmail.com	01063163654
20150221	Mohamed Magdy Hassan	Mohamed14magdi@gmail.com	01122678532
20150176	Omar Farag El Khodairy	Omar.farag1996@gmail.com	01280361766

Definitions, acronyms, and abbreviations

V.A.L.L.E.Y	Value Accessibility Luxury Leadership Efficiency Yield
PCI DSS	The Payment Card Industry Data Security Standard.
HTTPS	Hyper Text Transfer Protocol with Secure sockets layer.
OAuth2	thorization framework that enables applications to obtain limited access to user accounts on an HTTP service.
IE9	Internet explorer version 9.
PBKDF-2	Password-Based Key Derivation Function 2.
REST APIs	Representational State Transfer application program interface.

Document Purpose and Audience

The purpose of this document is to provide a detailed overview over "Valley e-Commerce platform" and its outline and how it is expected to perform. Also provide descriptive and detailed requirements for its software. It illustrates and describe the software purpose, functionality, software/hardware requirements, interface, usage flow, constraints and roles.

The document is intended for the customer review and approval, and as a reference for the developing team during the developing and testing phases.

Introduction

Software Purpose

Valley is an online e-commerce platform that allow its users to buy and sell products online in a convenient and concise way by handling the inconvenience of online shopping for buyers such as transactions and delivery, and also provide sellers and online platform to offer their products without the extra cost for developing and maintaining an online store.

Software Scope

- Valley is an online e-commerce platform to buy and sell virtual and physical products online.
- Users can browse the platform products by various search filters (e.g by brand, by name, by store, etc)
- There are two types of products, a virtual product that doesn't require delivery, and physical products.
- A User can apply to create his own store on the platform which isn't published until passing the verification process.
- A store can be of two types, an online virtual store which is limited only to virtual project, or a physical store which sells physical products.
- A store can only sell products that exists on the platform and are added and verified by the platform administrators.
- A store can request and apply products to be added to the platform that are not published until passing the verification process.
- A user can add multiple products to his shopping cart and proceed to payment with a single checkout.
- The platform supports multiple payment methods including on door payments and credit/debit cards.
- The platform supports voucher, discount and promotion cards during the checkout process.

Requirements

Functional Requirements

- A Homepage with latest products, hottest deals and featured stores.
- A User can access homepage, browse stores and check products without registering to the platform.
- A shopping cart for users to collect products they want to buy to be processed in a single checkout.
- A User cannot use shopping cart or checkout without registering.
- A User should be able to register to the platform by a unique email address and a unique username.
- A User should be able to save his own address, zip code, and credit card information to his profile.
- A User should be able to buy all his cart products in a single checkout process and a single interface.
- A User cannot buy physical products unless he has a valid address and zip code.
- A User cannot pay using credit card unless he has a valid and verified credit card information.
- A User can enter a promotion/voucher/discount card code during the checkout process.
- Promotion cards codes should be unique, used once and have an expiration date.
- Promotion cards can be store specific or global.
- Physical products are shipped to buyer’s address and online products are sent to the buyer’s email.
- A User can apply to create his own store that is either an online only store or a normal store.
- Normal store require the owner to provide the store address when applying that is to be verified by the platform administrators.
- A Store owner can only add products to the store from the platform products list.
- A Store owner can request to add a certain product to the product lists.
- Products are added to the platform by the platform administrators after verification.
- Products are either an online product or a physical product and not both.
- Products should have a Name, Brand and a manufacturing Company.
- A Store owner should be able to apply to issue promotion cards that are specific to his store only.
- A Store owner should have a dashboard that contains:
    - Requested Products (That are still to be picked and shipped to a customer)
    - Old sales history
    - Statistics and view counts about his products and store.
    - Balance
- An administrator should add stores that are applied by users after s/he verifies it.
- An administrator should add products that are applied by store owner after s/he verifies it.
- An administrator should issue promotion cards codes that are either applied by store owners or are global (by the platform)
- An administrator should be able to manage all stores and products data.

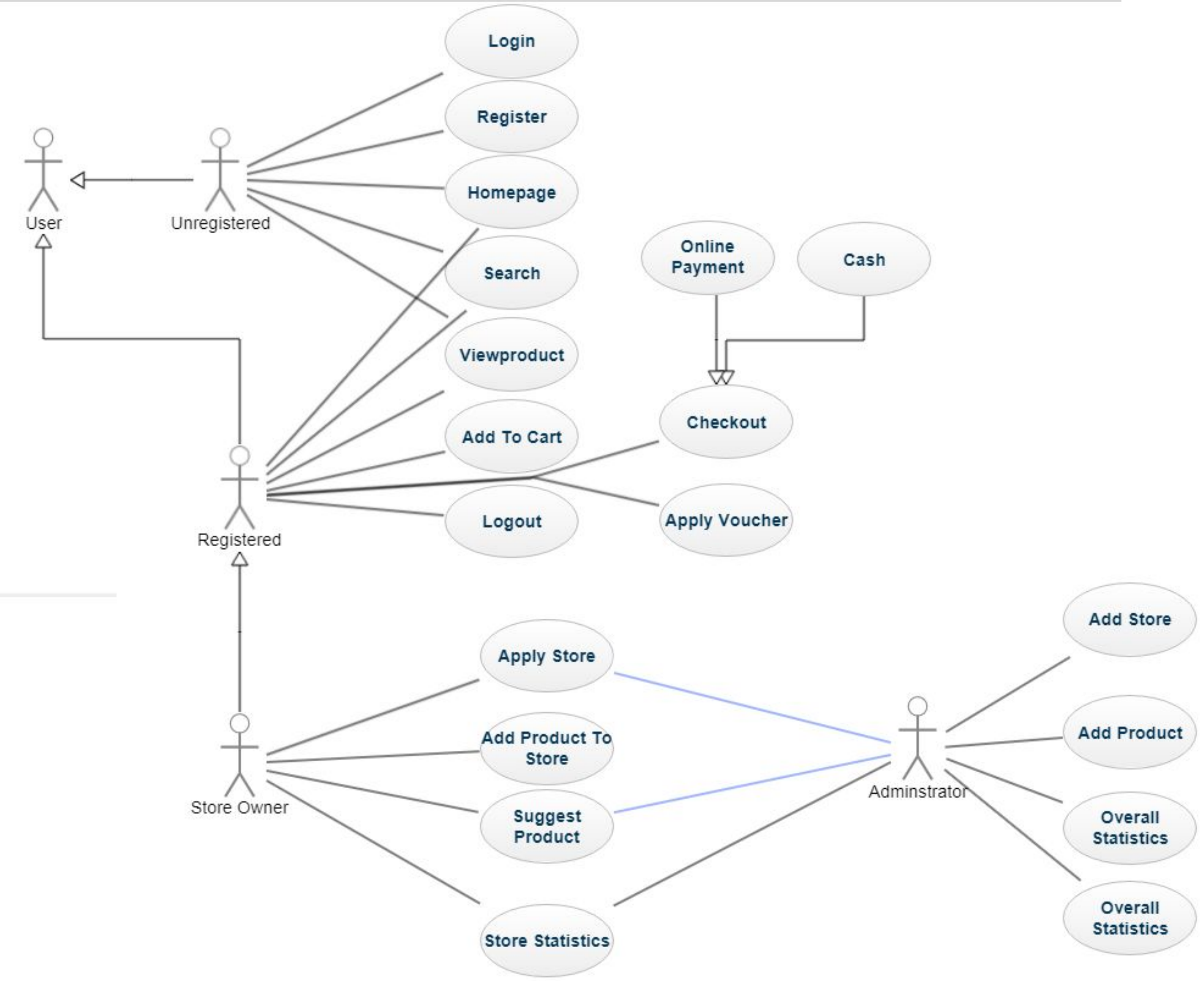
Non Functional Requirements

Type	Details
Performance	● Average page-load time should be under 1500ms using a 2Mb/s internet connection.
Scalability	● The system should withstand 1000 page request per second on normal traffic days. ● The system should withstand sudden spikes and scale up accordingly on high-traffic days (e.g prime day, black friday, valentines, christmas, etc) ● The system should be easily scaled up either vertically or horizontally.
Availability	● The system should score minimum of 90.0% uptime under most situation.
Backup	● The system should be backed up weekly, with exception of transactions logs which must be backed up daily.
Documentation	● A Documentation for all the system's architecture, components, models, functions and schemas should be submitted upon finishing the system.
Extensibility	● The system should be designed with extensibility in mind, all system constants should be editable without editing source code via a configuration interface. All System components should be adaptive to change without drastic change in the core source code.
Privacy	● All Users Passwords should be salted using a random salt and hashed using PBKDF-2. ● All Users credit-card information should be stored and encrypted according to the PCI DSS standards.
Security	● All the system traffic and http requests should be bidirectionally encrypted and authenticated using HTTPS protocol. ● All the system REST-APIs should use the OAuth2 security protocol. ● All Backups should be encrypted at rest.
Portability	● The platform should be accessible through all browsers modern browsers including IE9. ● The platform should be accessible for users using both Android and iOS devices. ● The platform backend and frontend should be vendor independent.
Reliability	● The system should avoid all possible single point of failure. ● The system should contain a failure detection, logging and analysis system.

System Models

Use Case Model

The new user can view the products and register, during registration user's data must be validated, the registered user can purchase physical products if he has a valid address, to checkout on online products verification on the credit card must be applied, Payment can be done using credit card or on door for physical product. Physical products are shipped and online products sent by email, The user can register for an online store or physical store and store validation must be applied. The store owner can add products to his store. The administrator adds the stores and manages them.



Use Case Tables

Use Case ID:	U1	
Use Case Name:	User Viewing products	
Actors:	User	
Pre-conditions:	This use case doesn't require pre conditions.	
Post-conditions:	This use case doesn't require post conditions.	
Flow of events:	**User Action**	**System Action**
	1- User view the home page	
	2- User select a certain a product view its details.	
	3- User click to buy the product.	
		4- Product is added to the shopping cart.
Exceptions:	**User Action**	**System Action**
	1- User try to buy a product without logging in.	
		2- Login/Registration form is opened.
Includes:	User Authentication - Registration - Purchase	

Use Case ID:	U2	
Use Case Name:	Registration	
Actors:	User	
Pre-conditions:	This use case doesn't require pre conditions.	
Post-conditions:	User Authentication	
Flow of events:	**User Action**	**System Action**

	1- User Enter his data	
		2- System Verify user data and the address
Exceptions:	**User Action**	**System Action**
	1- User Enter invalid data	
		2- System rejects to create account for the user
Includes:	User Authentication	
Notes and Issues:		

Use Case ID:	S1	
Use Case Name:	User Authentication	
Actors:	User, System	
Pre-conditions:	The user must enters a data	
Post-conditions:	Approve or Reject	
Flow of events:	**User Action**	**System Action**
	1- User enters registration form data	
		2- Take the info (Address, Credit Card no , Email …)
		3- System Verify all the info

Use Case ID:	U3
Use Case Name:	Purchase, Payment and Checkout
Actors:	User
Pre-conditions:	The user must be registered , Verified and ask to buy a certain product

Post-conditions:	A product is shipped to the user's address or sent by email if it is an online product.	
Flow of events:	**User Action**	**System Action**
	1- User ask to buy a product	
		2- System checks if the product is in stock or not.
	3- User Enter the address and credit card number	
		4-User Authentication and deliver the product
Exceptions:	**User Action**	**System Action**
	1- User try to buy a product without having enough balance.	
		2- Purchase aborted.
Includes:	User Authentication	

Use Case ID:	U4	
Use Case Name:	Store Registration	
Actors:	User, Administrator	
Pre-conditions:	The user must be registered and must ask to register to a store	
Post-conditions:	Approve or reject	
Flow of events:	**User Action**	**System Action**
	1- User ask to register a store and enter store address if it's a normal store.	
		2- The Store information are applied and sent to administrators for validation.

		3- The administrators check and verifies the store information.
Exceptions:	**User Action**	**System Action**
	1- User try to register a fake onsite store	
		2- The application is rejected.

Use Case ID:	U5	
Use Case Name:	Adding Products	
Actors:	User , Administrator	
Pre-conditions:	The user must be registered and has a registered store	
Post-conditions:	The user adds the products	
Flow of events:	**User Action**	**System Action**
	1- User ask to add a product	
		2- Administrator checks if the product is an existing and legal product. if so the system add the product to the platform.
Exceptions:	**User Action**	**System Action**
	1- User try to add a fake or illegal product or an already existing product.	
		2- Systems rejects to add the product.

Ownership Report

Item	Owners
Document Purpose and Audience	*20150309,20150221*
Software Purpose	*20150309,20150101,20150221*
Software Scope	*20150309,20150101,20150221*
Functional Requirements	*20150309,20150204*
Non Functional Requirements	*20150309,20150176*
Use Case Model	*20150309,20150204, 20150101*
Use Case Table	*20150204, 20150309, 20150176*
Definitions, acronyms, and abbreviations	*20150176*