

Object-Oriented Programming

Project 03: “Shopify”

6388078 Suphawit Xu

6388101 Jittiwat Sanit

6388103 Viphu Sopanakitkosol

Description:

As of today, the technology has improved a lot and it is now in "Internet of thing" era. People yearn for more conveniency and as of that there are increment in online shopping needs. Our website consists of online marketplace along with many features including selling and buying in one platform, E-wallet, search on item tags and much more!!

Benefit:

Our application will improvise uses in database which will contain shops and demonstrate a market like sensation. Customers can add multiple items or products required to the basket and will not have to pay on delivery. Instead, we provide them an E-Wallet system which will upgrade the conveniency and is suitable for this era. Furthermore, merchant can open up shop account uses for selling and add any item as they would like to sale along with tags which will help users to search for any specific item required.

Requirements:

In order to use the website, user needs to:

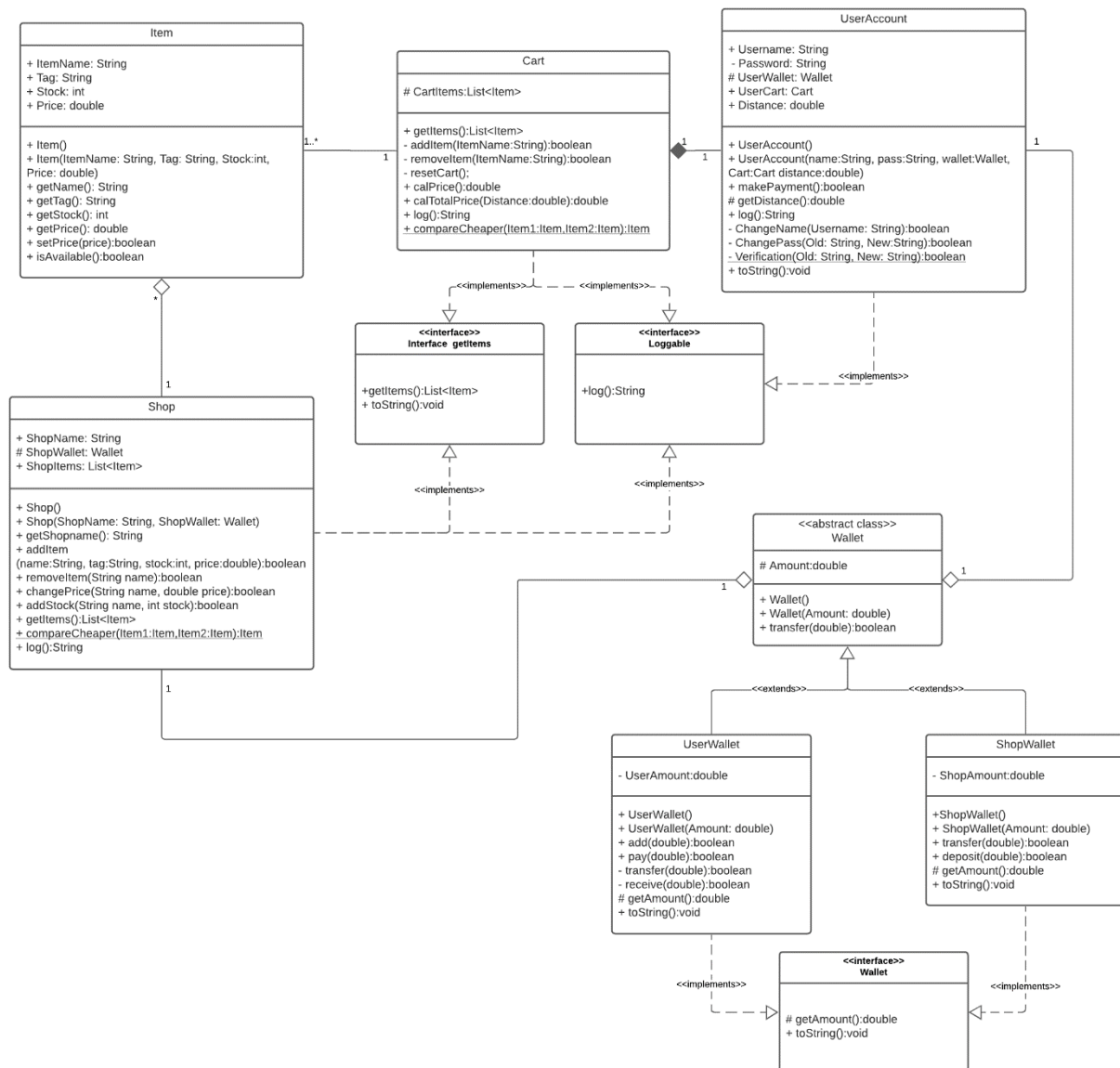
- 1.) Sign in using username and password.
- 2.) Add balance from the top right of homepage in order to make payment.
- 3.) Access the item needed to buy.
- 4.) Add the item to cart.
- 5.) Make a payment through cart page.

Classes:

Class	Description
Datamanagement	Datamangement class contains attribute of shop list along with methods to search for the shop by name.
Shop	Shop consists of any attributes for merchant account for instance name, wallet, and item.
Item	Item class is used to represent attributes of each item used in the system.
UserAccount	UserAccount class is used for containing attributes and information of each users.
Cart	Cart class represents the basket which each user will hold on to. Items can be added into the basket along with the method to calculate final sum of all items ordered.
Wallet	Wallet class keeps information and attributes of each user's wallet.
ShopWallet	A class which extends from the wallet, represents the wallet of the owner of the shop.
UserWallet	A class which extends from the wallet, represents the wallet of the customers.

Interfaces:

Interfaces	Description
Loggable	Turns information in the implemented class to string in order to print out.
Wallet	Wallet interface has a method of getting amount and turns into string in order to print out.
getItem	getItem interface has a method of getting list of items and a method to turn information into string.



UML Diagram

Class UserWallet

public class UserWallet extends Wallet

Class Description: A class which extends from the wallet, represents the wallet of the customers.

Constructor	Description
UserWallet()	Construct a new value for UserAmount variable to 0.
UserWallet(double Amount)	Construct a new value for UserAmount variable with Amount parameter.

Methods	
Modifier and Type	Description
public boolean	add(double sum) <ul style="list-style-type: none">This method is for adding a sum of money to the current amount. Parameters <ul style="list-style-type: none">sum – amount of money needed to add. Throws <ul style="list-style-type: none">none pay(double sum) <ul style="list-style-type: none">This method is for subtracting the current wallet amount with sum parameter. Parameters <ul style="list-style-type: none">sum – amount of money needed to subtract. Throws <ul style="list-style-type: none">none
private boolean	transfer(double sum) <ul style="list-style-type: none">This method is for transferring a sum of money to another account. Parameters <ul style="list-style-type: none">sum – amount of money needed to transfer. Throws <ul style="list-style-type: none">none
	receive(double sum) <ul style="list-style-type: none">This method is for receiving a sum of money from another account. Parameters <ul style="list-style-type: none">sum – amount of money needed to receive. Throws <ul style="list-style-type: none">none

protected double	getAmount()
public void	toString()

Class ShopWallet

public class ShopWallet extends Wallet

Class Description : A class which extends from the wallet, represents the wallet of the owner of the shop.

Constructor	Description
ShopWallet()	Construct a new value for ShopAmount variable to 0.
ShopWallet(double amount)	Construct a new value for ShopAmount variable with amount parameter.

Methods	
Modifier and Type	Description
public boolean	transfer(double sum) <ul style="list-style-type: none"> This method is for transferring a sum of money to another account. Parameters <ul style="list-style-type: none"> sum - amount of money needed to transfer. Throws <ul style="list-style-type: none"> none
	deposit(double sum) <ul style="list-style-type: none"> Deposit amount of money according to parameter sum. Parameters <ul style="list-style-type: none"> sum – amount of money needed to deposit. Throws <ul style="list-style-type: none"> none
protected double	getAmount() <ul style="list-style-type: none"> This method returns value of ShopAmount variable. Parameters <ul style="list-style-type: none"> none Throws <ul style="list-style-type: none"> none
public void	toString()

Class Wallet

Abstract class Wallet

Class Description: Wallet class keeps information and attributes of each user's wallet.

Constructor	Description
Wallet()	Construct a new value for Amount variable with 0.
Wallet(double Amount)	Construct a new value for Amount variable with Amount parameter.

Modifier and Type	Description
public boolean	transfer(double sum) <ul style="list-style-type: none">This method is for transferring a sum of money to another account. Parameters <ul style="list-style-type: none">sum - amount of money needed to transfer. Throws <ul style="list-style-type: none">none

Class UserAccount

public class UserAccount

Class Description: UserAccount class is used for containing attributes and information of each users.

Constructor	Description
UserAccount()	Construct a new user account with default variable
UserAccount(String Name, String Pass, Double distance)	Construct a new user account with specific name, password, and distance.

Method	Description
Public boolean	makePayment() <ul style="list-style-type: none">This method is for deducting money from user's wallet. Parameters <ul style="list-style-type: none">None Throws <ul style="list-style-type: none">None
Private boolean	ChangeName(String name) <ul style="list-style-type: none">This method changes username of the account.Return 1 if success, return 0 if not.

	Parameters <ul style="list-style-type: none"> name – new username to change Throws <ul style="list-style-type: none"> None ChangePass(String Old, String New) <ul style="list-style-type: none"> This method is for changing password. Return 1 if success, return 0 if not. Parameter <ul style="list-style-type: none"> Old – current password for verification. New – new password to change. Throws <ul style="list-style-type: none"> None Static Verification(String Pass1, String Pass2) <ul style="list-style-type: none"> This method compares between 2 passwords If 2 passwords are the same, return 1. If not, return 0. Used in ChangePass method. Parameter <ul style="list-style-type: none"> Pass1 – first password input Pass2 – second password input Throws <ul style="list-style-type: none"> None
Public double	getDistance()
Public String	Log()
Private void	toString()

Class Shop

public class Shop

Class description: Shop consists of any attributes for merchant account for instance name, wallet, and item.

Constructor	Description
Shop()	Construct a new shop with default variable
Shop(String name)	Construct a new user account with specific name

Method	Description
Public boolean	<p>addItem(String name, int stock, double price)</p> <ul style="list-style-type: none"> This method is for adding an item into the list Return 1 if success, return 0 if not. <p>Parameters</p> <ul style="list-style-type: none"> name – item name. stock – quantity of the item. price – item price. <p>Throws</p> <ul style="list-style-type: none"> None <p>removeItem(String name)</p> <ul style="list-style-type: none"> This method is for removing an item in the list. Return 1 if success, return 0 if not. <p>Parameters</p> <ul style="list-style-type: none"> name – name of the item to remove. <p>Throws</p> <ul style="list-style-type: none"> None. <p>changePrice(String name, double price)</p> <ul style="list-style-type: none"> This method is for changing price of the item. Return 1 if success, return 0 if not. <p>Parameters</p> <ul style="list-style-type: none"> name – name of the item to change price price – new price to change <p>Throws</p> <ul style="list-style-type: none"> None. <p>addStock(String name, int stock)</p> <ul style="list-style-type: none"> This method is for increasing quantity of the item. Return 1 if success, return 0 if not. <p>Parameters</p> <ul style="list-style-type: none"> name – name of the item to add stock. stock – quantity of the item to add. <p>Throws</p> <ul style="list-style-type: none"> None.

Public List<item>	getItems() <ul style="list-style-type: none"> Return list of items Parameters <ul style="list-style-type: none"> None Throws <ul style="list-style-type: none"> None
Public Item	Static CompareCheaper(Item item1, Item item2) <ul style="list-style-type: none"> Compare price between items Parameters <ul style="list-style-type: none"> Item1 Item2 Throws <ul style="list-style-type: none"> none
Public String	getShopname() log()
Private void	toString()

Class Item

public class Item

Class description: Item class is used to represent attributes of each item used in the system.

Constructor	Description
Item()	Construct a new shop with default variable
Item(String ItemName, String tag, int Stock, double price)	Construct a new user account with specific name

Method	Description
Public boolean	setPrice(double price) <ul style="list-style-type: none"> This method is for setting price of the item. Return 1 if success, return 0 if not Parameters <ul style="list-style-type: none"> Price – amount of money to set price Throws <ul style="list-style-type: none"> None IsAvailable() <ul style="list-style-type: none"> This method is for getting availability of an item If the item has more than 0 return 1. Else, return 0. Parameters <ul style="list-style-type: none"> None Throws <ul style="list-style-type: none"> None

Public int	getStock()
Public double	getPrice()
Public String	getName() getTag()
Public void	toString()

Class Cart

public class **Cart**

Class description: Cart class represents the basket which each user will hold on to. Items can be added into the basket along with the method to calculate final sum of all items ordered.

Constructor	Description
Cart()	Construct a new Cart object with default variable

Method	Description
Private boolean	addItem(String itemName) <ul style="list-style-type: none"> This method is for adding an item to the list Return 1 if success, return 0 if not Parameters <ul style="list-style-type: none"> itemName – Throws <ul style="list-style-type: none"> removeItem(String itemName) <ul style="list-style-type: none"> This method is for removing an item to the list Return 1 if success, return 0 if not Parameters <ul style="list-style-type: none"> itemName – Throws <ul style="list-style-type: none"> None
Public List<item>	getItems() <ul style="list-style-type: none"> This return list of items Parameters <ul style="list-style-type: none"> None Throws <ul style="list-style-type: none"> None
Public Item	<u>CompareCheaper(Item item1, Item item2)</u> <ul style="list-style-type: none"> Compare price between items Parameters <ul style="list-style-type: none"> Item1 Item2

	Throws <ul style="list-style-type: none"> • none
Public double	CalPrice() <ul style="list-style-type: none"> • This method is for calculating all items price Parameters <ul style="list-style-type: none"> • None Throws <ul style="list-style-type: none"> • None CalTotalPrice(double Distance) <ul style="list-style-type: none"> • This method is for calculating all items price with shipping cost Parameters <ul style="list-style-type: none"> • Distance – use for calculating shipping cost Throws <ul style="list-style-type: none"> • None
Public String	Log()
Private void	resetCart() <ul style="list-style-type: none"> • This method is for resetting cart and remove all items Parameter <ul style="list-style-type: none"> • None Throws <ul style="list-style-type: none"> • None
Public void	toString()

Class <interface>Loggable

public class Loggable

Class Description: Turns information in the implemented class to string in order to print out.

Method	Description
Public String	Log()

Class <interface>getItems

public class getItems

Class Description: getItem interface has a method of getting list of items and a method to turn information into string.

Method	Description
Public List<item>	getItems() - This return list of items
Public void	toString()

Class <interface>Wallet

public class Wallet

Class Description: Wallet interface has a method of getting amount and turns into string in order to print out.

Method	Description
Protected double	getAmount() - This method returns value of variable.
Public void	toString(f

Shopify

As of today, the technology has improved a lot and it is now in "Internet of thing" era. People yearn for more conveniences and as of that there are increment in online shopping needs. Our website consists of online marketplace along with many features including selling and buying in one platform, E-wallet, search for name, store and tags, and much more!!

Figure 1 : Login in page, since the user has to login before making any payment.

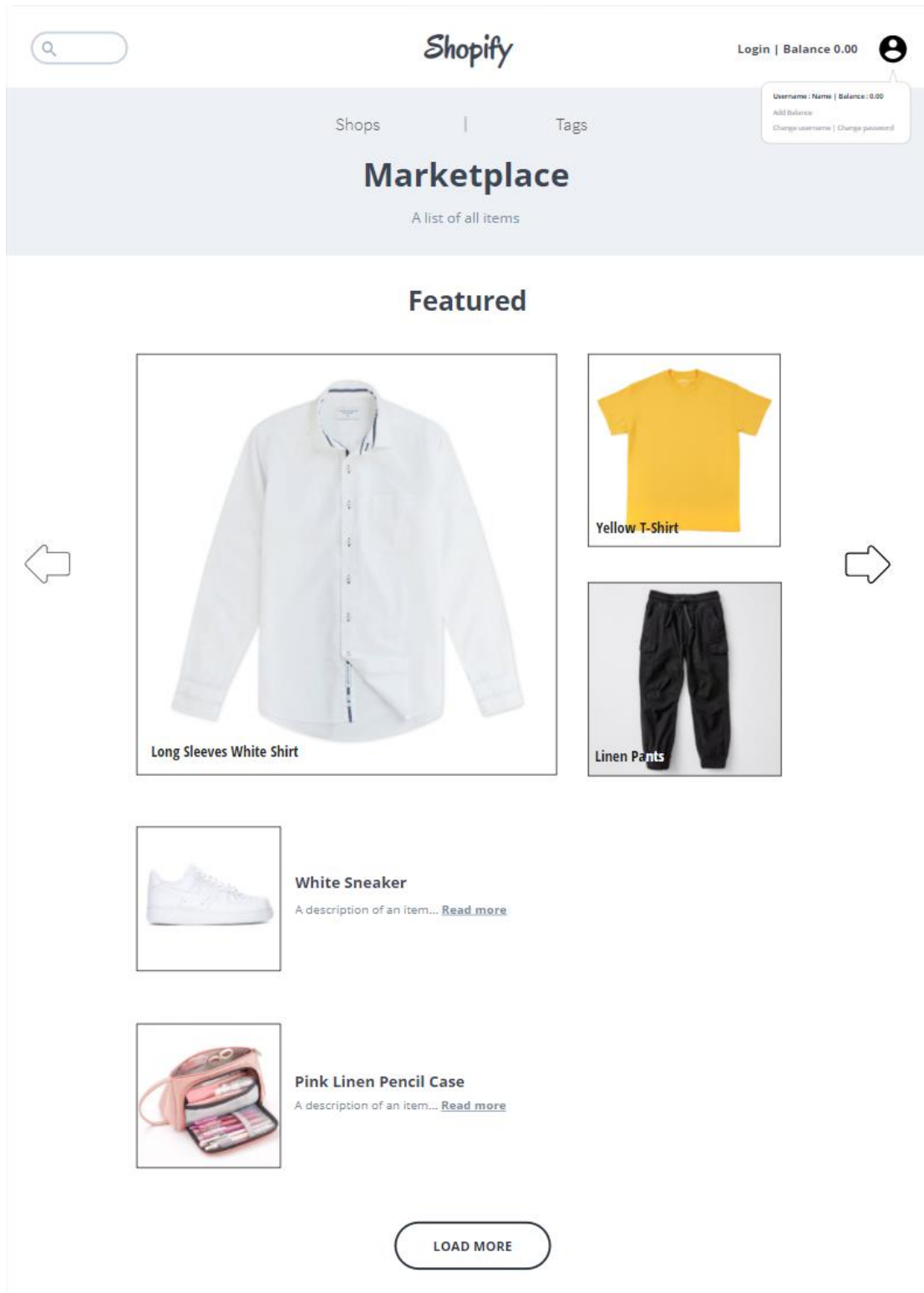


Figure 2 : Homepage, for when the user first access the website, this page will provide a list of all items with a featured items order by rating. User can add balance from the top right pop up.

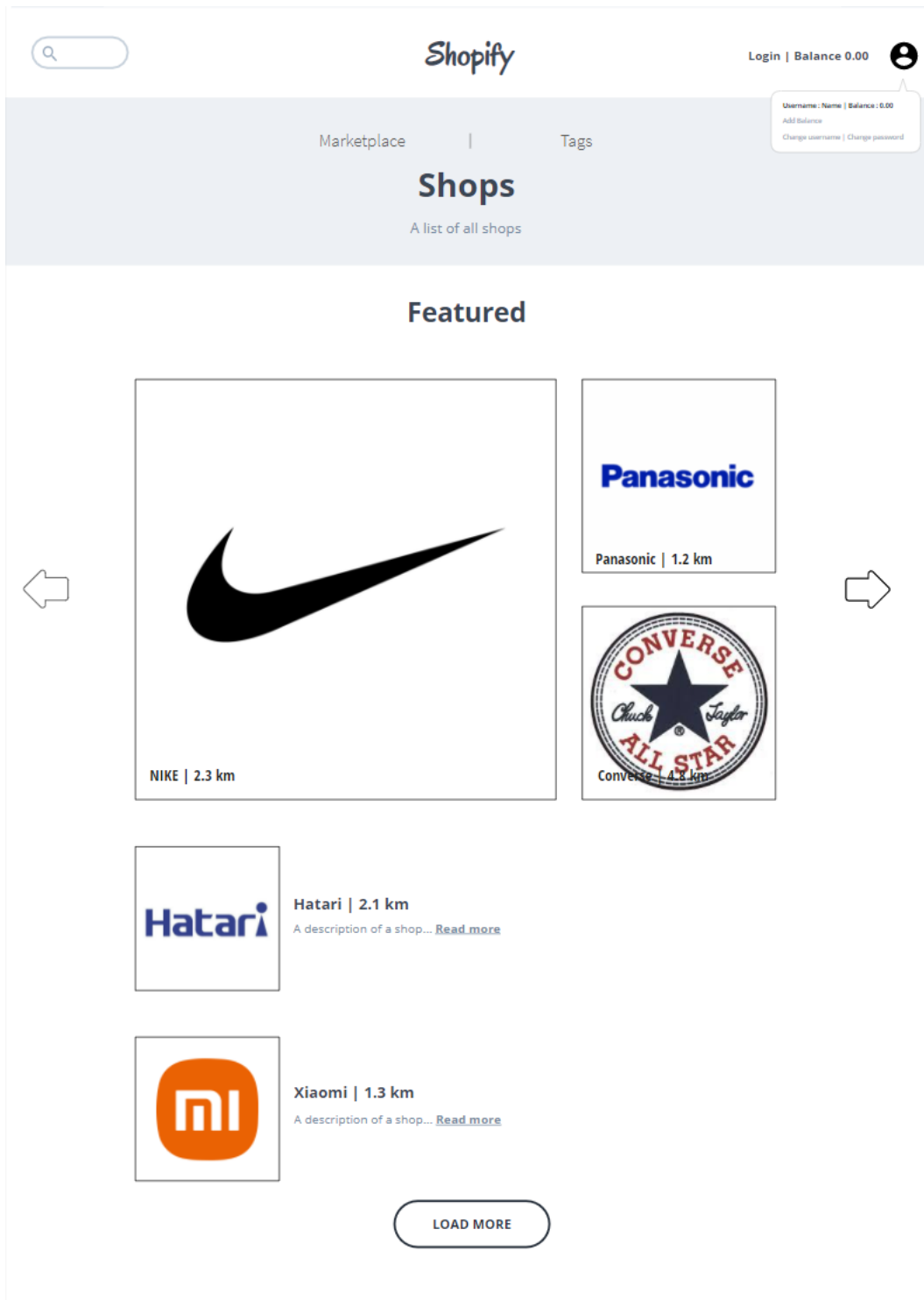


Figure 3 : User can switch category to list of all shops, this will not only express the name but also the distance away from the user. User can add balance from the top right pop up.

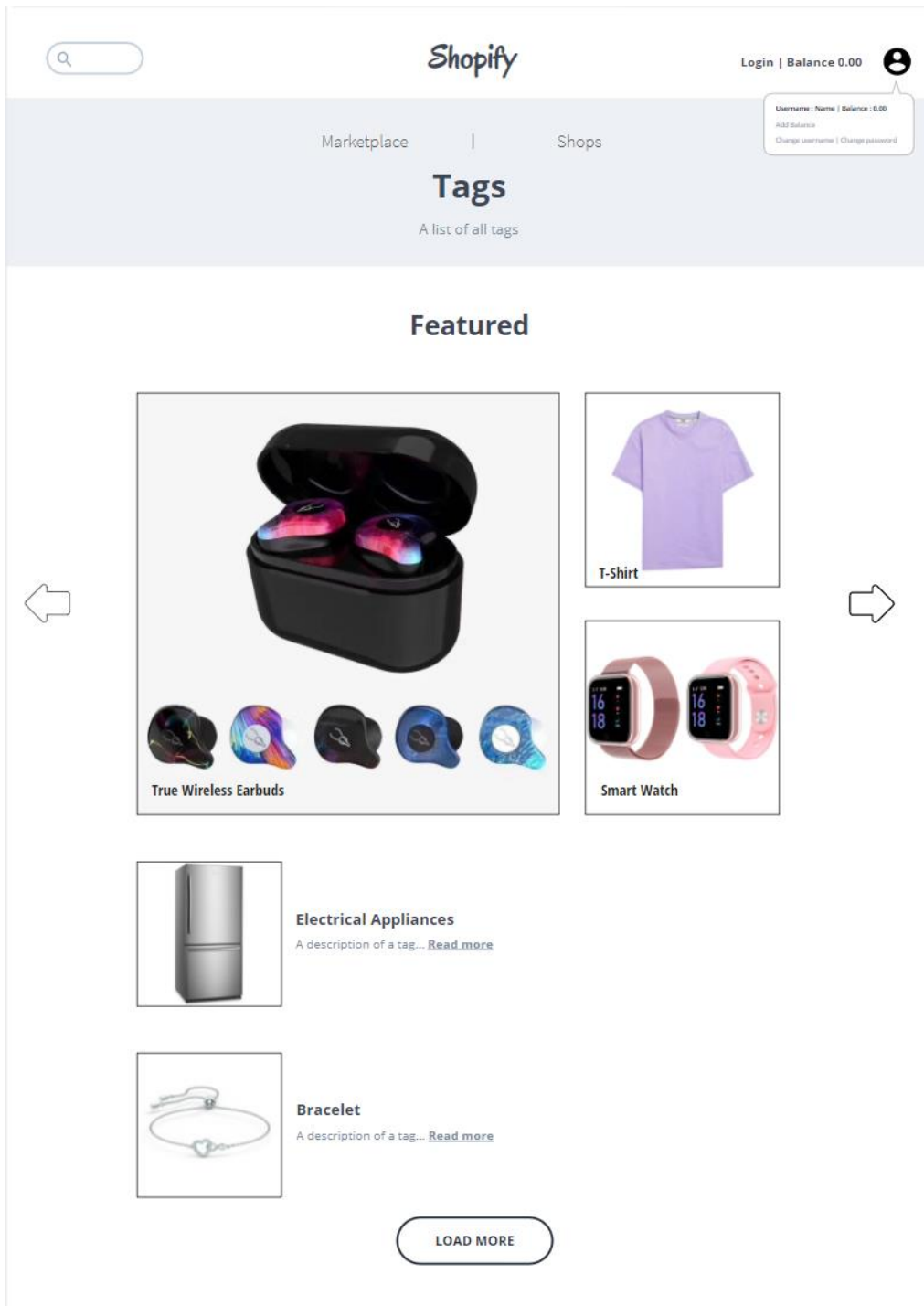


Figure 4 : User can switch to category to list of all tags, this will express a tag name. User can add balance from the top right pop up.



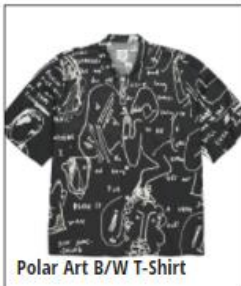
Humming Bird T-shirt

- Rating: 5.00/5.00
- Description: A humming bird t-shirt. Regular fit, round neckline, short sleeves. Made of extra long staple pima cotton.
- Price: 300.00 Baht
- Stock: 30 left
- Tags: T-Shirt, Short Sleeves, Clothing
- Distance : 0,6 km

Add to Cart

Qty: 1

Related Product



Polar Art B/W T-Shirt



Til' Death Do Us Part T-Shirt



Abstract Art White T-Shirt

LOAD MORE

Figure 5 : After the user access an item, it will direct to this page. This page consists of a description of an item along with related product to advertise the customer further.

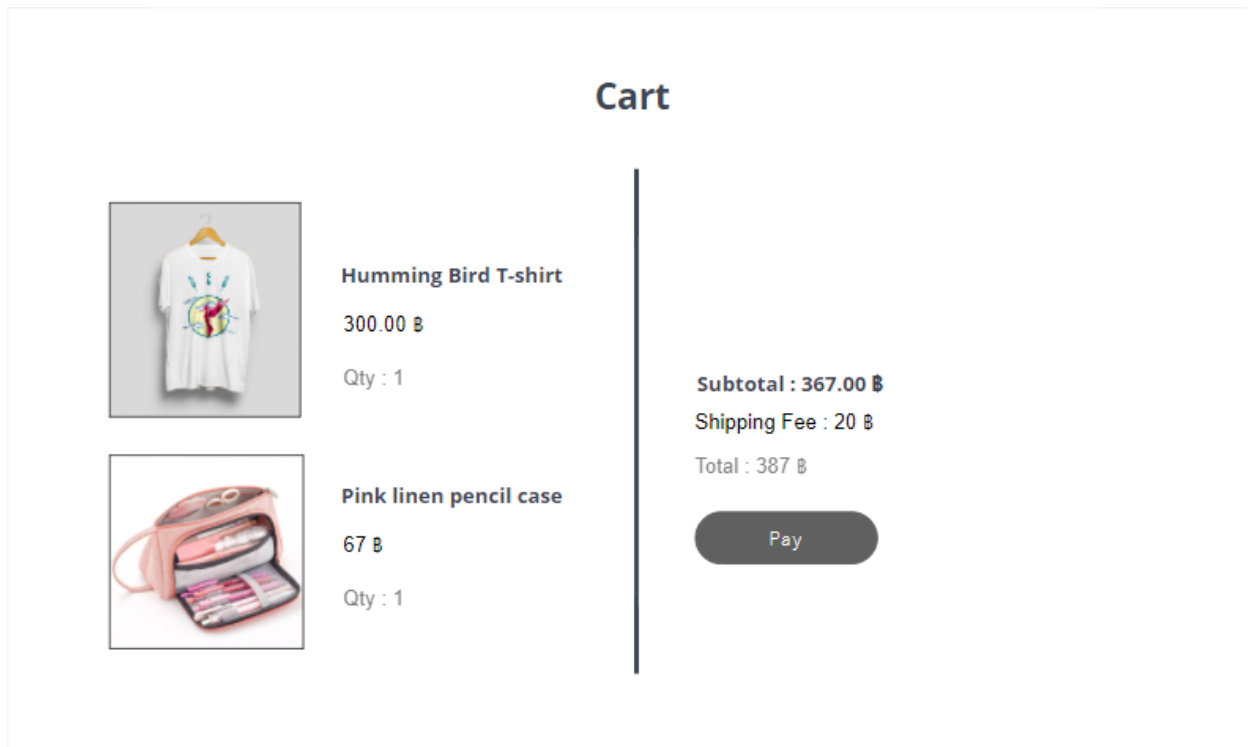


Figure 6 : After the user click add to cart, it will direct to this page and the user can make a payment from this page if user's wallet has enough balance.

Website's User Interface Wireframe

