

Sri Lanka Institute of Information Technology

**Online Buying and Selling System**

Programming Applications and Frameworks (IT 3030)

2019

**Group ID: 402 – G8**

Submitted By:

* IT 17052498 Jayawardena J.K.S.L.
* IT 17005968 Karthikeyan J.
* IT 17022170 Perera P.S.R.
* IT 17071512 Ayesh Dissanayaka
* IT 17043342 Gunasinghe M.D.

Public VCS repo : Add link

**Workload Allocation**

|  |  |
| --- | --- |
| **IT Number - Name** | **Function** |
| IT17052498 - Jayawardena J.K.S.L. |  |
| IT17005968 - Karthikeyan J. |  |
| IT17022170 - Perera P.S.R. |  |
| IT17071512 - Ayesh Dissanayaka |  |
| IT17043342 - Gunasinghe M.D. |  |

1. **SE methodology**

Agile is an iterative approach to project management and software development that helps teams deliver faster and fewer headaches value to their customers. Instead of betting on a "big bang" launch, an agile team delivers work in increments that are small but consumable. Requirements, plans, and outcomes are continually evaluated so that teams have a natural mechanism to respond quickly to change. A set of ceremonies or specific development techniques does not define Agile. Instead, agile is a group of methodologies that show commitment to tight feedback cycles and ongoing improvement.

1. **Stakeholder Analysis (onion diagram)**

In a few circles, the onion diagram can display layers of online buying and selling system. In the 1st inner circle, it shows the stakeholders who are very close to the product. Which would be the new system or process. The stakeholders are consisting of the project leader, developer, QA, etc. In the 2nd inner circle shows the stakeholders whose work changes when the solution is defined as end users. In this diagram end, users are registered user, registered buyer, etc. The innermost circle represents the system, the next circle captures all of the users of the system. The outer circle captures all of the other employees within the organization who rely upon the system. Last circle or 4th circle display External Stakeholders such as customers, regulators, government, suppliers. Figure 1.1:Onion Stakeholder Diagram

1. **Requirements Analysis**

Functional Requirement

* Keeping records of admission of customers.
* Keeping the records of customers.
* Keeping the daily sell.
* Keeping details about the product it is delivered or not.
* Storing the items selected by the customer in the temporary storage.

Non-functional Requirement

* **Extensibility** – ensuring that the platform is extended in such a way to make future development feasible
* **Data integrity and retention** – defining how long data should be stored and reviewable
* **Testing** – defining how unit testing will be built into the solution
* **Compatibility** – ensuring that the platform can be easily integrated with 3rd party systems
* **Accessibility** – ensuring that the platform meets the basic accessibility standards throughout
* **Documentation and Usability** – ensuring that the platform is sufficiently documented and easy to understand with minimal or no training.

Technical Requirements

* Web Server
* Server Software
* Web Tools
* Database System
* Networking
* Browser Compatibility
* Ports
* Domain Names

1. **Use Case Diagram**

A Use Case diagram for the online buying and selling system consisting of mainly five Actors. They are Customer, Seller, System Administrator, Creditcard Company and Shipping Company. Customer can perform register and login to the system, manage cart, can view product details by selecting category or search product. Also, the user can add items to cart and make payment after the credit card verification done by the credit card company. User has to enter the valid credentials and after to use the credit card. User allowed to view orders and view receipt as well. As a seller, he can register and login to the system and after login to the system, he can update the product information. The seller can add new products to the system and update the products and also delete or remove the products from the system. Also, the seller can view the order details and manage details. The admin can login to the system, manage user accounts, activate or deactivate users and manage categories. The credit card company has to process the user’s credit card details and verify them and then allow the user to perform a transaction using the credit card. The shipping company shipping the products and manage shipping request. Figure 3.1:Use case Diagram

1. **System Architecture**

User login to the system. Then through the API, trying to create user accounts using account services. The users are mainly admin, buyer, and supplier. A database stores all the account details of all the users. Buyer and supplier use product inventory service and that details stored in the product inventory database. Other users use buyer order service, payment service, and shipping service. This details stored in buyer order database, payment database, and shipping database respectively. Figure 4.1:System Architecture

1. **Activity diagrams**

To register to the system user has to go to the user interface of the web application. After clicking the register button user can proceed to the registration page and then fill the user details. After submission of the registration form if the form validates user can go the login page. But the form is invalid, the user has to reenter the valid details and resubmit the registration form and login. Figure 5.1:Register to the system

Next step is login to the system. User can get the login page by clicking the login button. Then the user can submit the form using valid credentials. If the login credentials are matching with inserted data in the database user can proceed to the home page. Otherwise, the user has to reenter the valid login credentials to log in. Figure 5.2:Login to the system.

After proceed to the home page,the user searches a product or click a category . Then the product will be display. If the searched product is not available the system displays an error message. After selecting the product, the user can add it to the cart. Then enter the card details to do the payments. Then the system checks the validation of entered details and if the details are invalid show the error message and user have to reenter the card details. If the details are valid, then check the available balance in the account. If the balance is insufficient, show the error message and ask the user to reenter another card. If the user disagrees, the user can terminate the process. Otherwise,the user can submit the valid card details. If the balance is sufficient for the payments to deduct the amount from card, send balance notification and invoice to user. After that process, order request sends to the seller and the seller prepares the package and sends it to the shipping company. Finally, the shipping company ships the package to the customer. Figure 5.3:Make Order

1. **Any Other Related Diagrams**

**Class Diagram**

There are nine classes in the diagram. They are Customer, Category, Supplier, Admin, Product, Order Details, Shipping, Payments, and Orders. Many Customers can have many Products. Therefore Customer Class and Products class have many to many Also many Products can have many Customers. Customer class has a composition relationship with the order detail class. Order details cannot exist without the Customer class. Admin can add one or more categories to the system. Admin class and category class connected with aggregation relationship. Category can have many Products. The product cannot exist without Category and Supplier. Order and Order Details are connected with the composition relationship. Order Details always depend on the Order. Payments and Orders are connected with the aggregation relationship. Also, one Shipping class can have many Orders.

Figure 7.1:Class Diagram

**ER Diagram**

In the ER diagram, entities are Customer, Category, Buyer, Product, Shipping, cartLine, and Payment. Order is the week entity. Many Customers can search for many Products. Category can store many Products.Many Buyers can add many products to the system. Customer can add many products into the cart. Buyer can arrange much Shipping. Cart must be checked out to prepare the Order. Order is week entity. Therefore Order will be meaningless without cartLine. One certain can prepare many Orders. There will be one Payment for one Order.

As previous the Payment will be useless without Order. Figure 7.2:ER Diagram

1. **Gantt chart**

