****

Asia Pacific College

School of Computing and Information Technology

3 Humabon Place, Magallanes Vlg., Makati City

Inventory Management System

By:

Ylaya, Owen James

Romero, Kenneth

Legaspi, Ron

BSIT-MI143

Prof. Manuel Sebastian Sanchez

CSPROJ Instructor

December 07, 2016

1. **Introduction**

**Executive Summary**

This Document was created to let the user know more about the system. The project was created for our current client, which is Tapsibog, was a small eatery before but grew to have many departments. The Project team is comprised of 3 students from Asia Pacific College. This three were tasked to do system for tapsibog inventory system after seeing how the existing system was. They have found some problems that need attention, this three-people continued developing the system. So far, they are close to finishing it.

The Project team’s objective in creating this project was generally spin around the issue of viability and productivity. They made the framework to determine the issue our customer is confronting. This study is constrained to what they can create and come up for them. It concentrates on how they can help them improve their school much. The system features a function that helps the client organize its files and record. It also has a feature that lets the staff member access and input data in a database system.

**Project Context**

Tapsibog is an eatery located at J.P. Rizal Makati. The owner told us that they want to expand their business and create more branches and of the reasons why the business is not slowly growing, is the manual way to track inventory, record reports.

Inventory management software is a computer-based system for tracking inventory levels, orders, sales and reports. Companies use inventory management software to avoid product overstock and outages. It is a tool for organizing inventory data that before was generally stored in hard-copy form or in spreadsheets.

**Purpose and Description**

The purpose of this project is to create a web-based inventory system that features:

1. Item Tracking: Every product in the eatery will be tracked.
2. To keep inventory at sufficiently high level to perform production and sales activities smoothly.
3. To minimize investment in inventory at minimum level to maximize profitability.

**Objectives**

1. To ensure that the supply of raw material & finished goods will remain continuous so that production process is not halted and demands of customers are duly met.
2. To minimize carrying cost of inventory.
3. Minimize manual work and papers that is time consuming and costly for the eatery

**Scope and Limitation**

The Scope of this project system are the item tracking, inventory managing, logs, and reporting.  
Item tracking will record all the transactions that have been inputted in the project system. Inventory Managing will edit product details easily. Reporting will show all the records and logs of the transactions. The project will reduce manual works for the employees.

**Software Environment**

**Programming Languages**

* + - * + PHP
        + MySQL

**Specific Software**

* + - * + Yii Framework
        + Internet Browser (IE, Edge, Mozilla Firefox, Chrome, etc.)
        + Notepad++, Sublime 3 or any PHP IDE
        + MySQL Workbench CE 6.3

**Web Server Applications**

* + - * + XAMPP Control Panel
        + IBM Blue mix

**Operating System**

* + - * + Microsoft Windows 8.1 or Later

1. **Technical Background**

Web-based Server System Requirements

1. Minimum Hardware Requirements

Processor: Dual Core CPU 2.0GHz

Instruction Set: 64-bit

RAM: 4GB

HDD: 250GB

1. Operating System

Windows 7 SP1 (32-bit/64-bit); OR

Microsoft Windows Server 2008 SP2 (32-bit); OR

Microsoft Windows Server 2012 (64-bit)

1. Database Requirements

XAMPP Control Panel

1. Server

Apache

Notes:

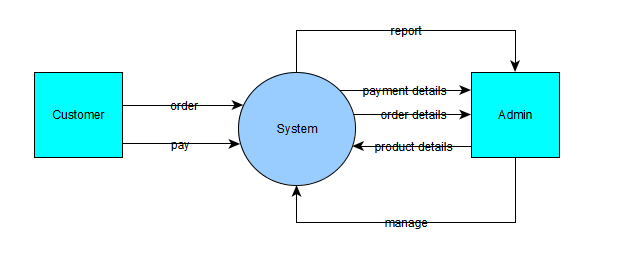
The system must be accessible by desktop computers and laptops (connected through LAN) within the campus.

The system will utilize Apache as its Web-Based Server System.

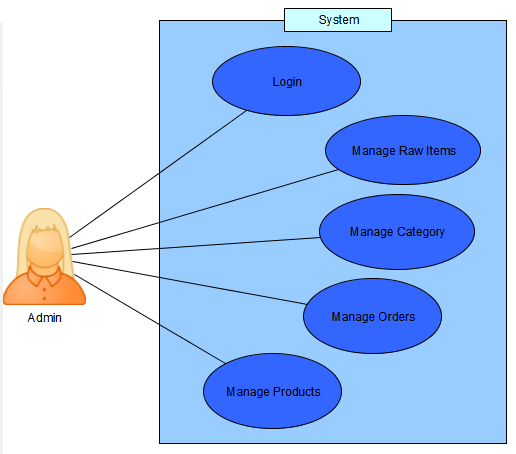
The system must be able to retrieve, store and view data.

1. **Methodology, Results and Discussion**

**Context Diagram:**



**USE CASE DIAGRAM**

****

**Event Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event | Trigger | Source | Use Case | Response | Destination |
| Admin inputs supply in inventory | Input and record supply details | Admin | Input Supply count in inventory | Recorded supply details in system | System |
| Customer orders a product | Receive and record order details | Customer | Record order details | Order details recorded in the system | System |
| Customer pays for his/her order | Receive payment | Customer | Get Payment | Compute payment from order price | System |
| New item quantity | Diminished product count | System | Diminish product quantity | Product quantity update | System |
| Admin manages inventory system | Manage/View  Products | Admin | Manage/View  Inventory System | Changes are Saved | System |

**Use Case: Full Description**

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Input supply count in inventory | |
| Scenario: | Replenished goods are recorded in the system | |
| Triggering Event: |  | |
| Actors: | Admin | |
| Related Use Cases: | Create Order | |
| Stakeholders: | Admin | |
| Preconditions: | Exact number of raw Items must be inputted | |
| Postconditions: | Number of raw items are recorded inside the system | |
| Flow of Activities: | Actor | System |
| 1. login  2. “Add Raw Items”  3. Input item count | 3.1 Admin page will be shown |
| Exception Conditions: | No new raw items | |

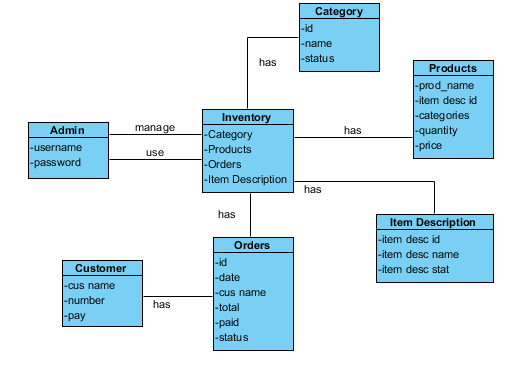
|  |  |  |
| --- | --- | --- |
| Use Case Name: | Order details recorded in the system | |
| Scenario: | Customer’s available requests are recorded | |
| Triggering Event: | Receive and record order details | |
| Brief Description: | New order will be added to the system | |
| Actors: | Customer, Admin | |
| Related Use Cases: | Get payment | |
| Stakeholders: |  | |
| Preconditions: | Products must be available | |
| Postconditions: | New order is added to the order list | |
| Flow of Activities: | Actor | System |
| 1. login  2. Admin inputs order | 2.1 System stores order  2.2 Computes required payment. |
| Exception Conditions: | Customer cancels order  Incorrect data inputted | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Get payment | |
| Scenario: | Customer pay the total amount of his/her order | |
| Triggering Event: | Customer pays order | |
| Brief Description: | Cashier receives payment | |
| Actors: | Customer, Cashier | |
| Related Use Cases: | New item quantity | |
| Stakeholders: | Customer, Admin | |
| Preconditions: | Cashier informs the customer the amount to be paid | |
| Postconditions: | Customer gives required amount payment | |
| Flow of Activities: | Actor | System |
| 2.1 Cashier informs the customer the amount to be paid  3. give change | 1. Compute for the amount  2. Display amount |
| Exception Conditions: |  | |

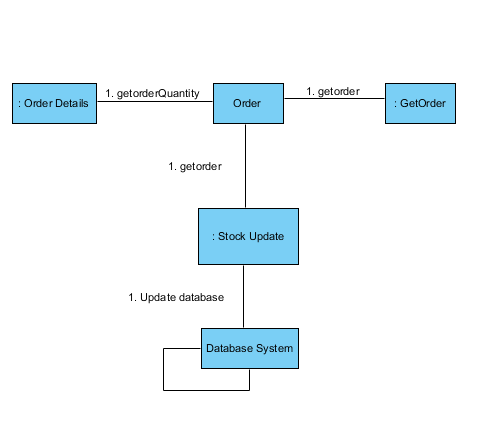
|  |  |  |
| --- | --- | --- |
| Use Case Name: | Diminish product quantity | |
| Scenario: | System will diminish product count by order details | |
| Triggering Event: | Update product quantity | |
| Brief Description: | System will diminish product quantity and raw items | |
| Actors: | System | |
| Related Use Cases: | Record order details | |
| Stakeholders: |  | |
| Preconditions: | Enter order details | |
| Postconditions: | Updated product quantity | |
| Flow of Activities: | Actor | System |
| 1. Input order details | 1. Record order details  1.1 Diminish product quantity |
| Exception Conditions: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Manage/View Inventory System | |
| Scenario: | Admin can Add, Delete, or Edit inside the system | |
| Triggering Event: | Manage Products, Categories, Raw Items | |
| Brief Description: | Admin can Add, Delete, or Edit the system and generate report | |
| Actors: | Admin | |
| Related Use Cases: |  | |
| Stakeholders: |  | |
| Preconditions: | Admin must be logged-in in the System | |
| Postconditions: | Admin can manage the inventory details | |
| Flow of Activities: | Actor | System |
| 1. login  2. Manage inventory  3. Save changes | 3.1 Changes are saved |
| Exception Conditions: |  | |

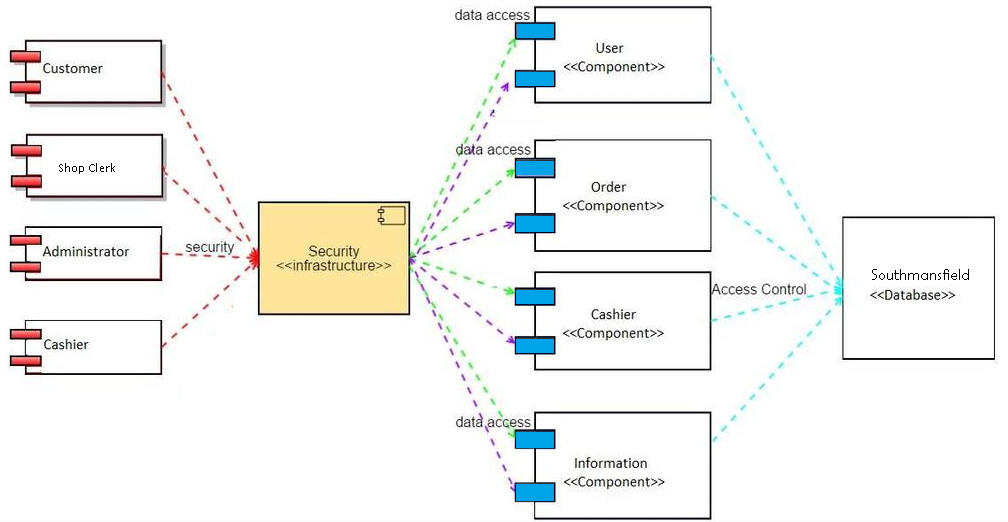
Class Diagram:



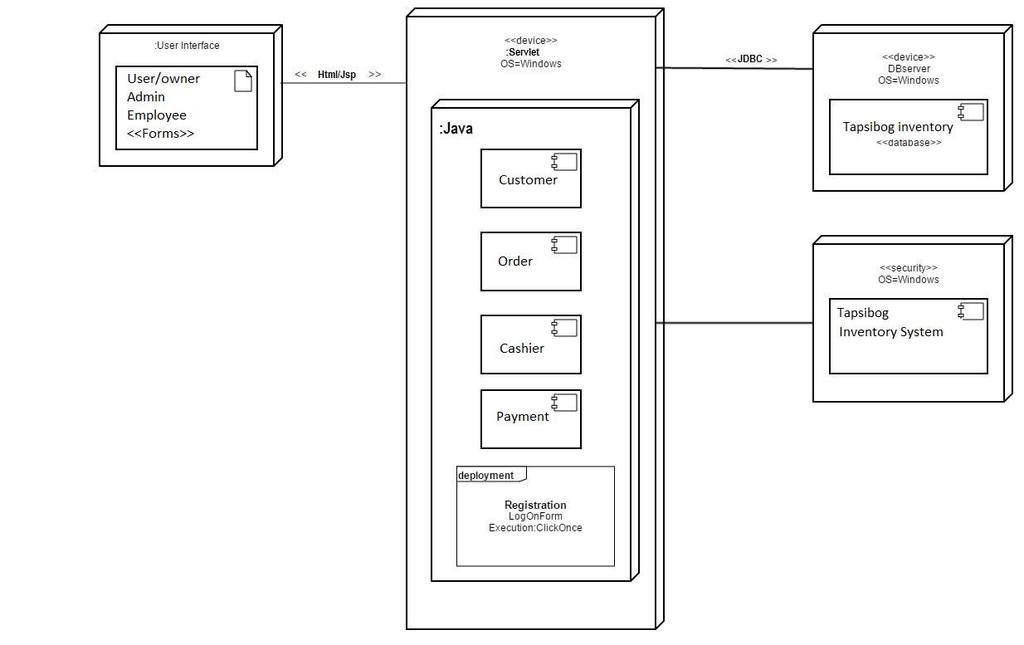
Communication Diagram:



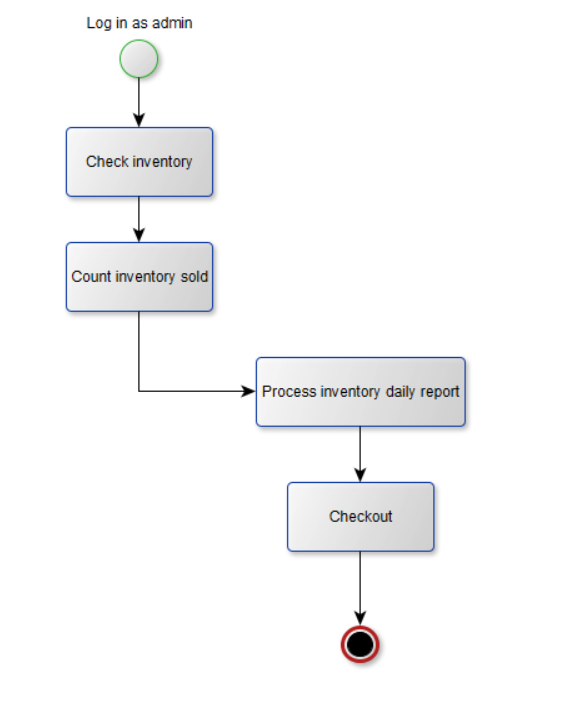
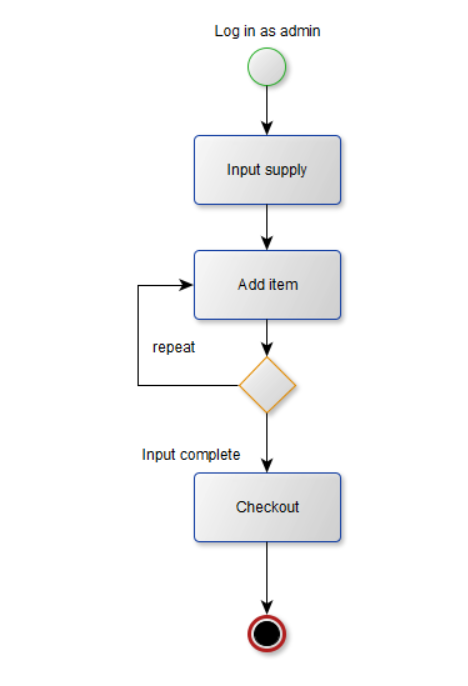
Component Diagram:

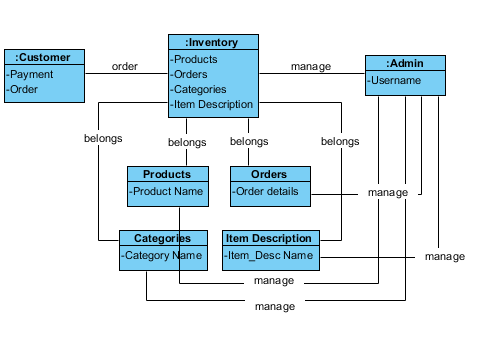


Deployment Diagram:



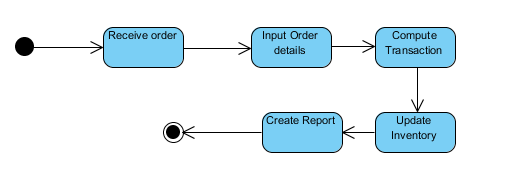
Interaction Overview Diagram:



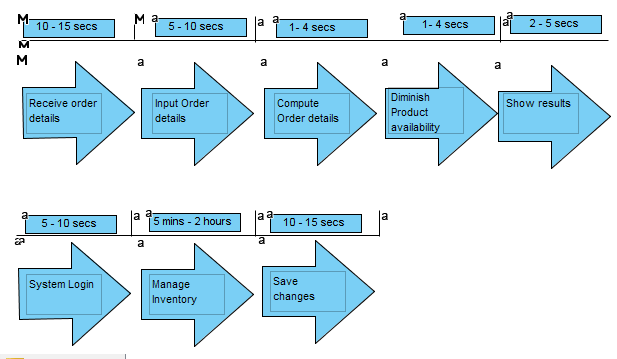


Object Diagram:

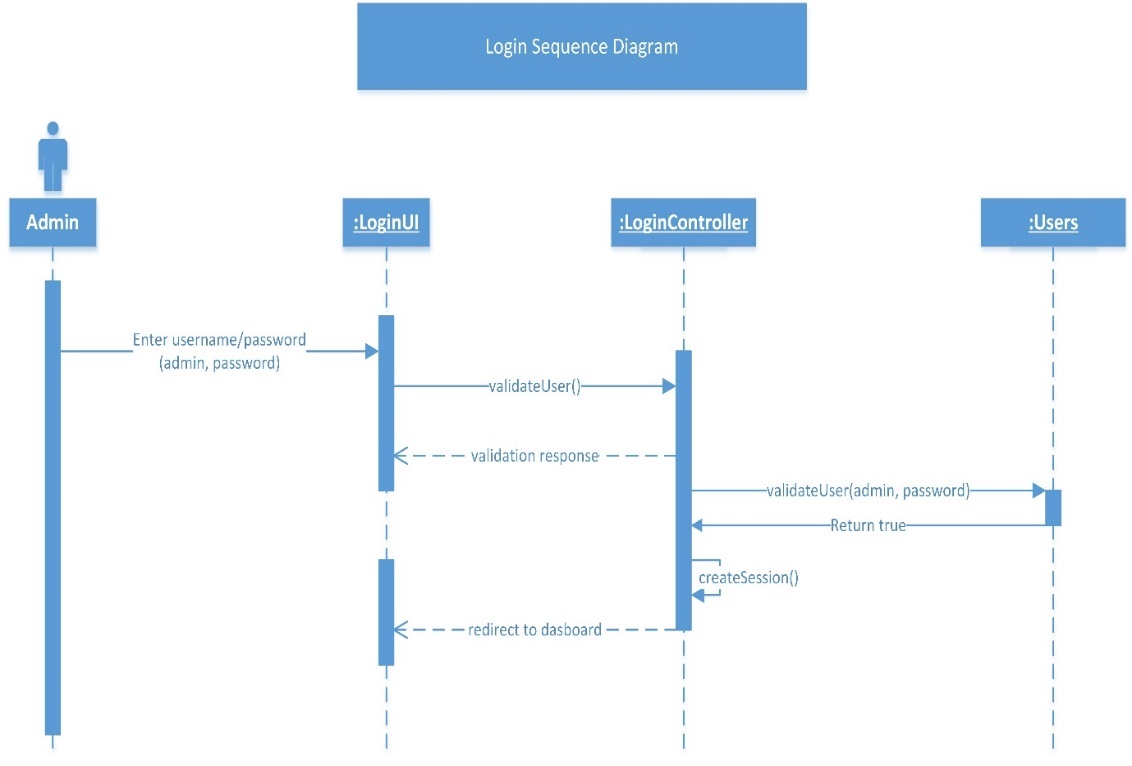
State Machine Diagram:

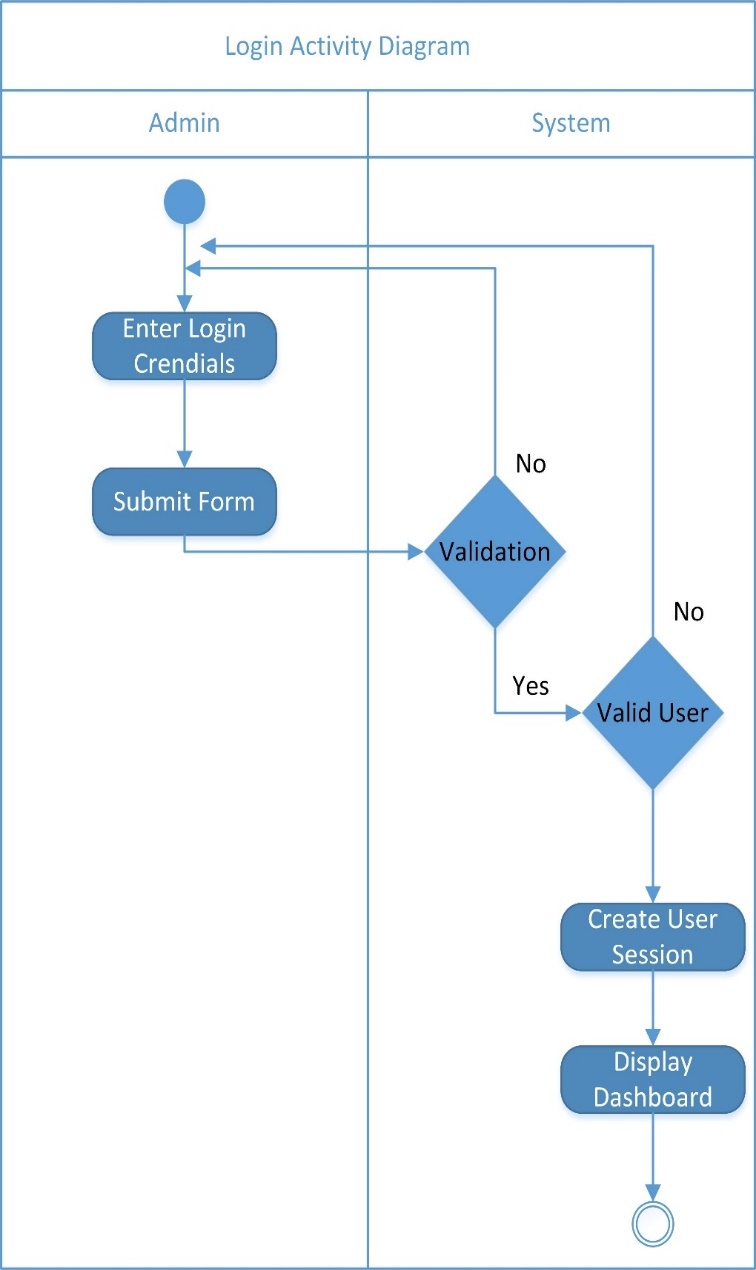


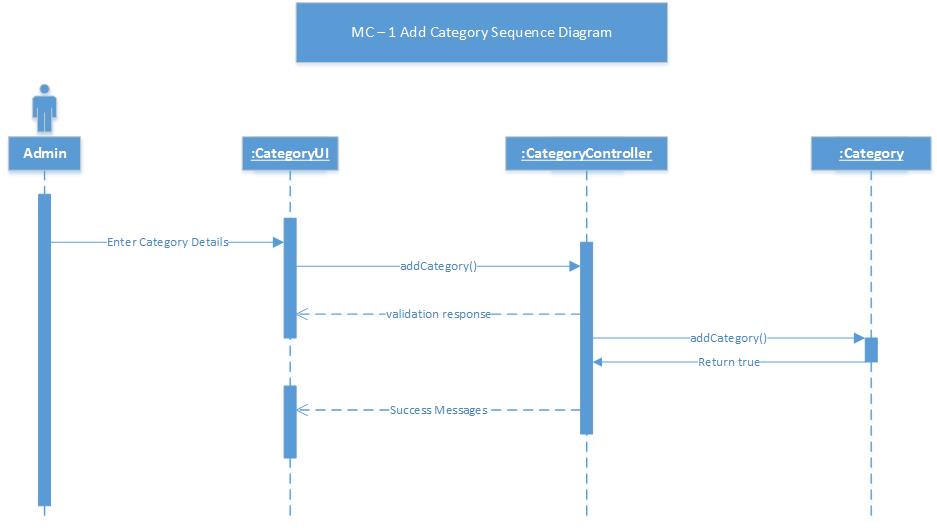
Timing Diagram:

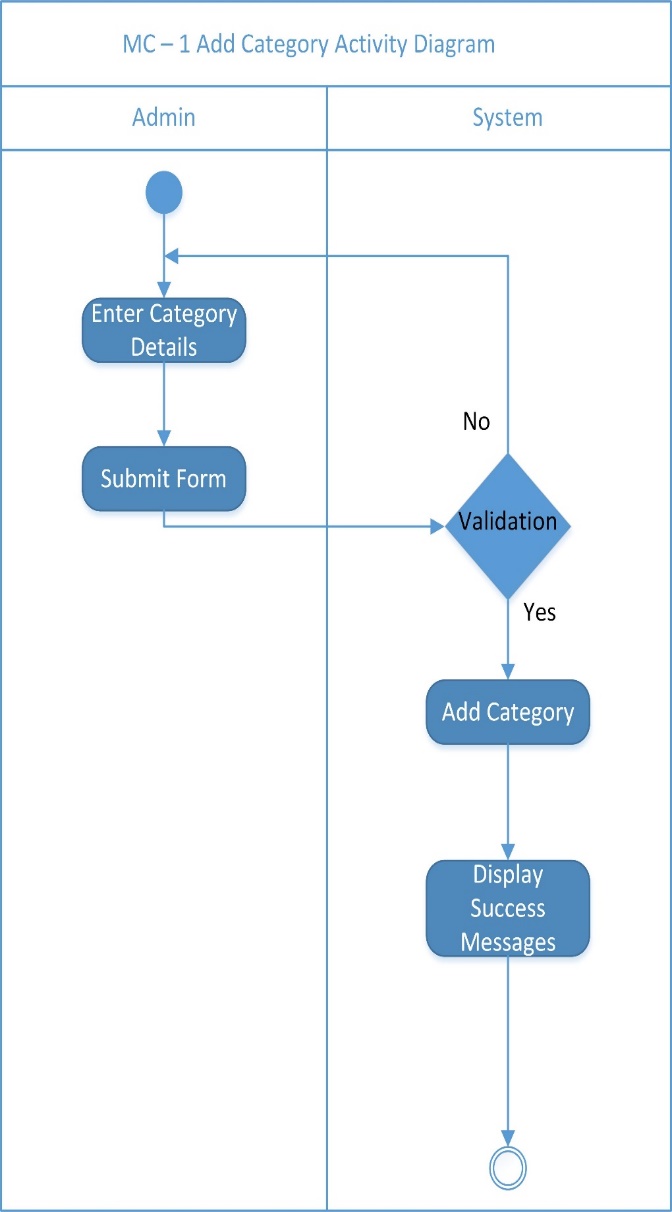


Login Sequence Diagram:

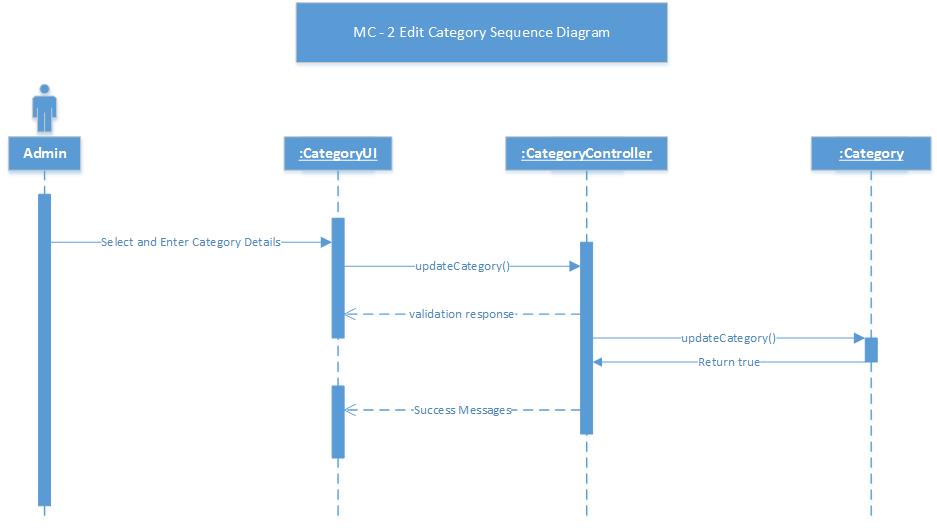
  
Login Activity Diagram

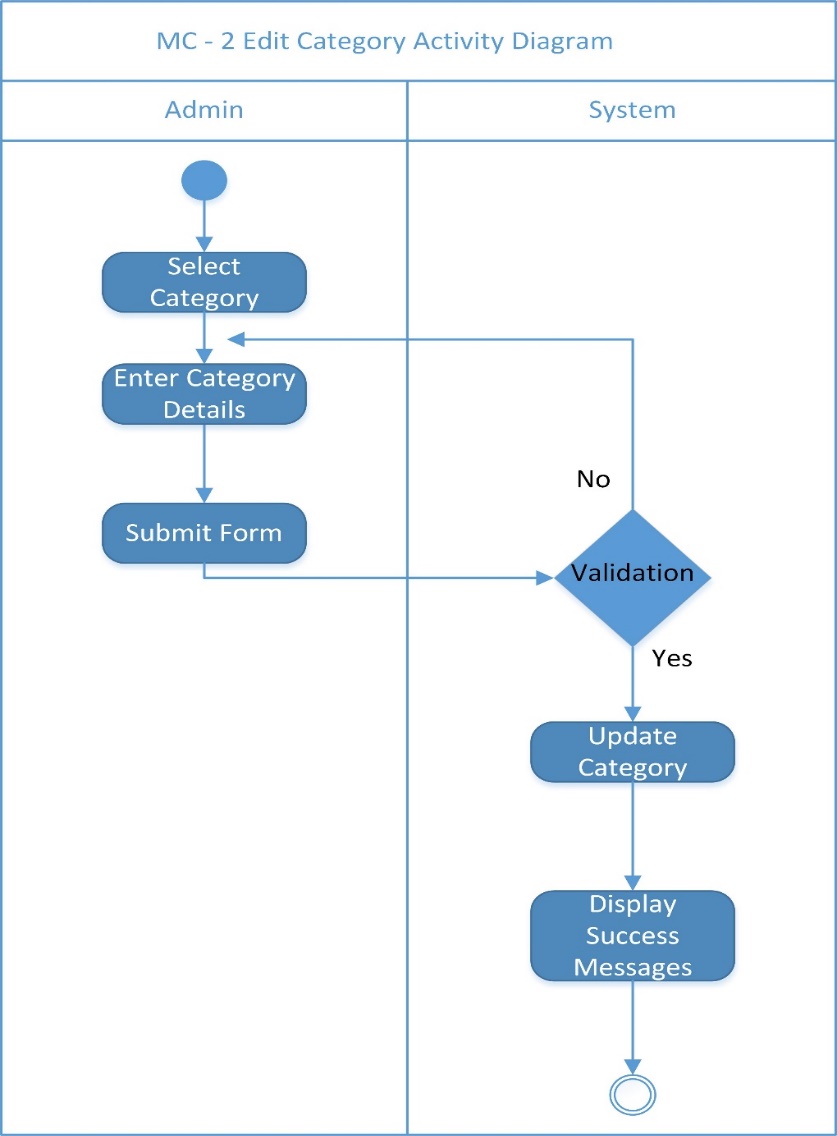


Add Category Sequence Diagram:  


Add Category  
  
Activity Diagram

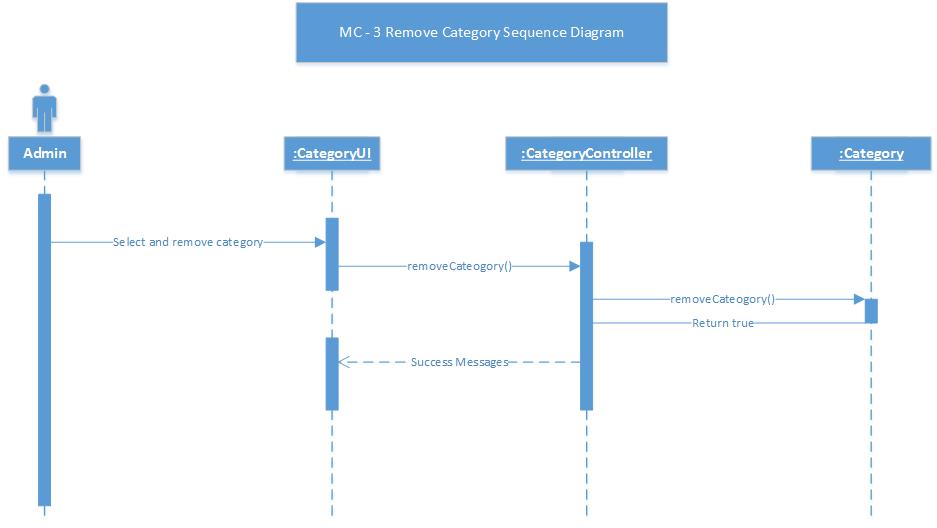
Edit Category Sequence Diagram

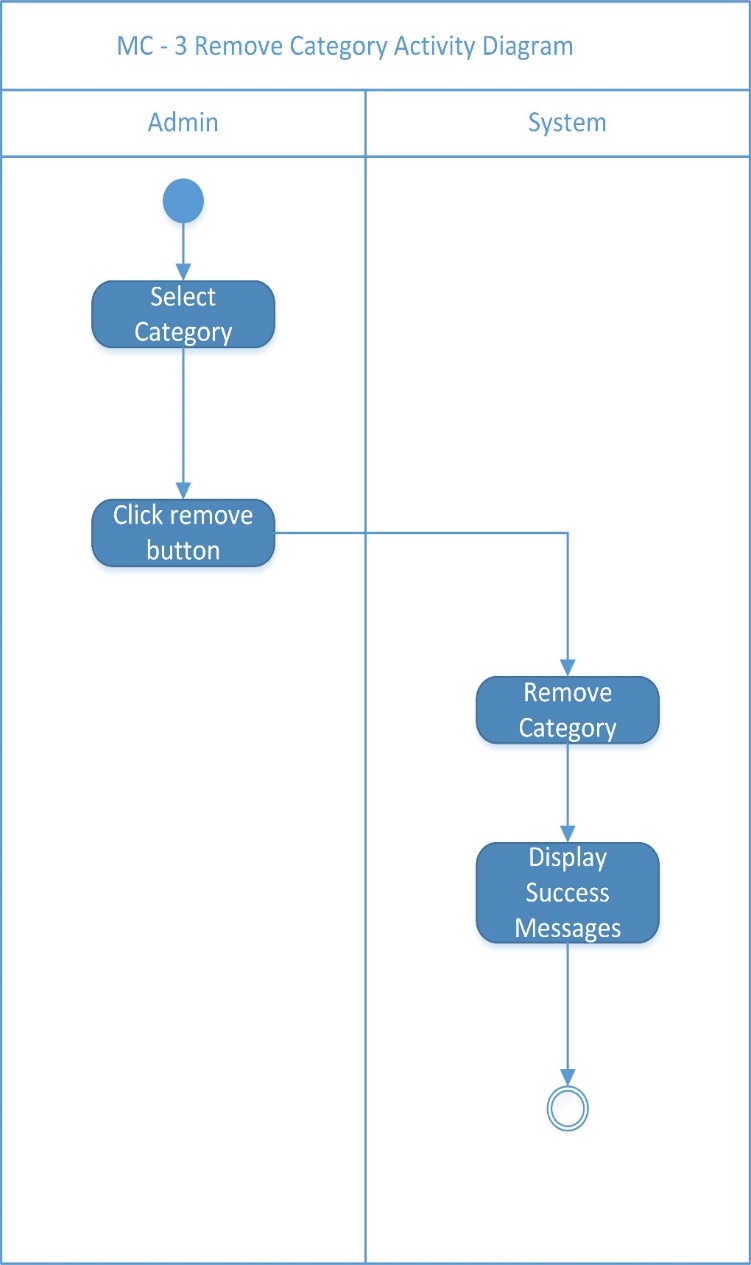




Edit Category  
  
Activity Diagram

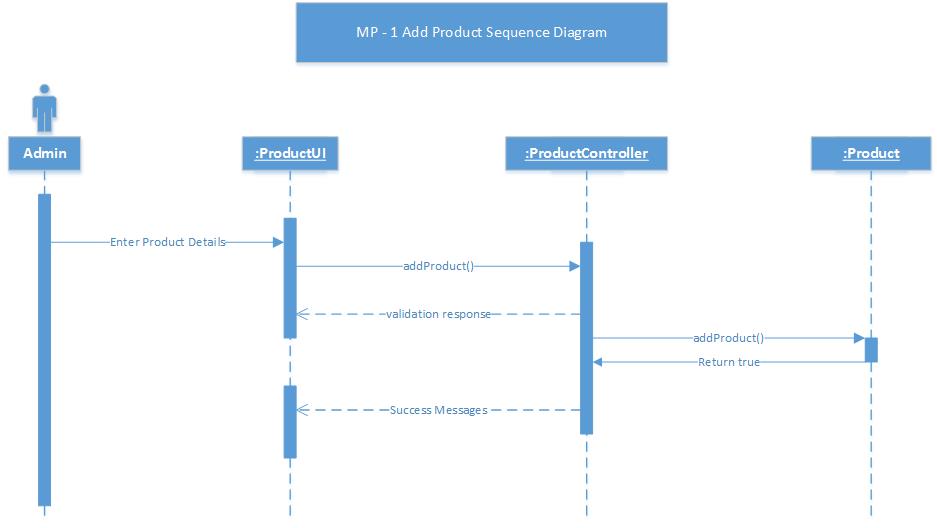
Remove Category Sequence Diagram

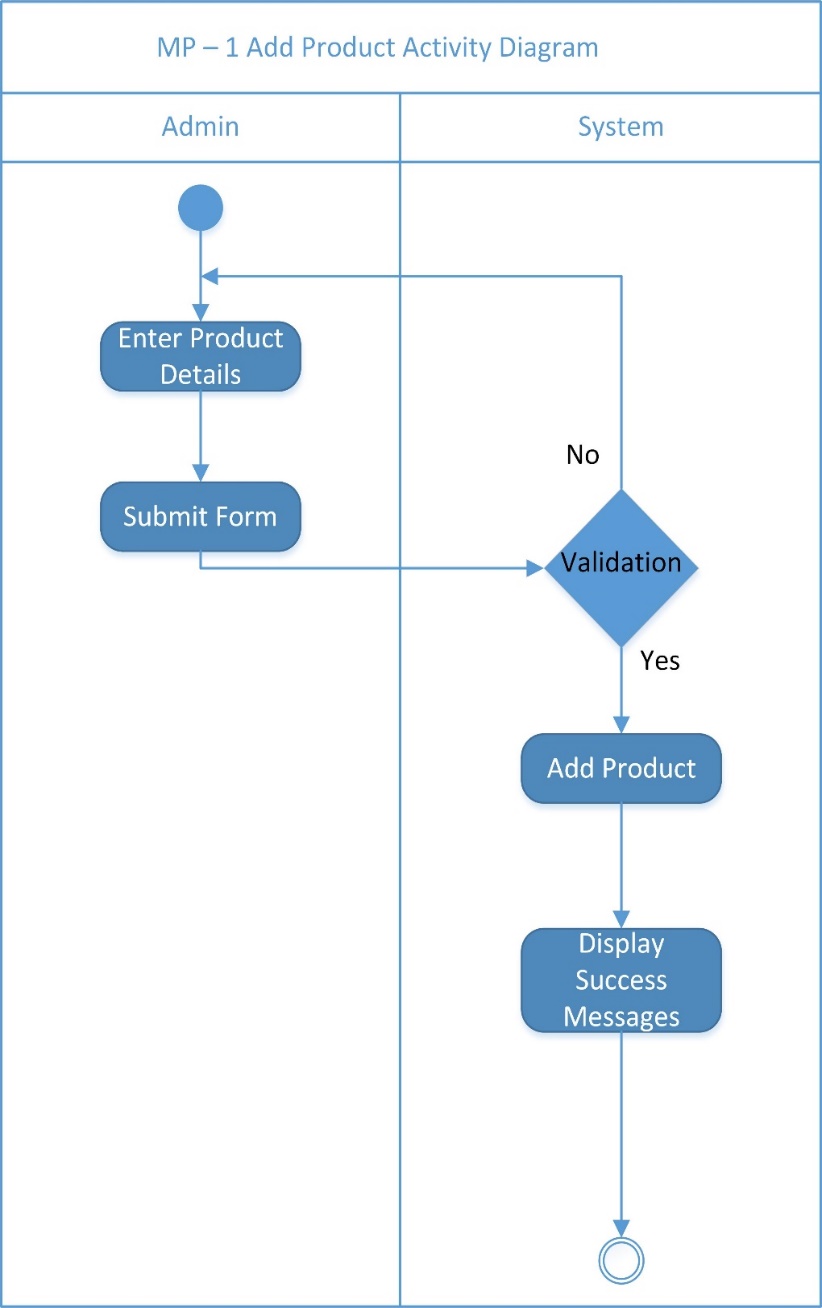




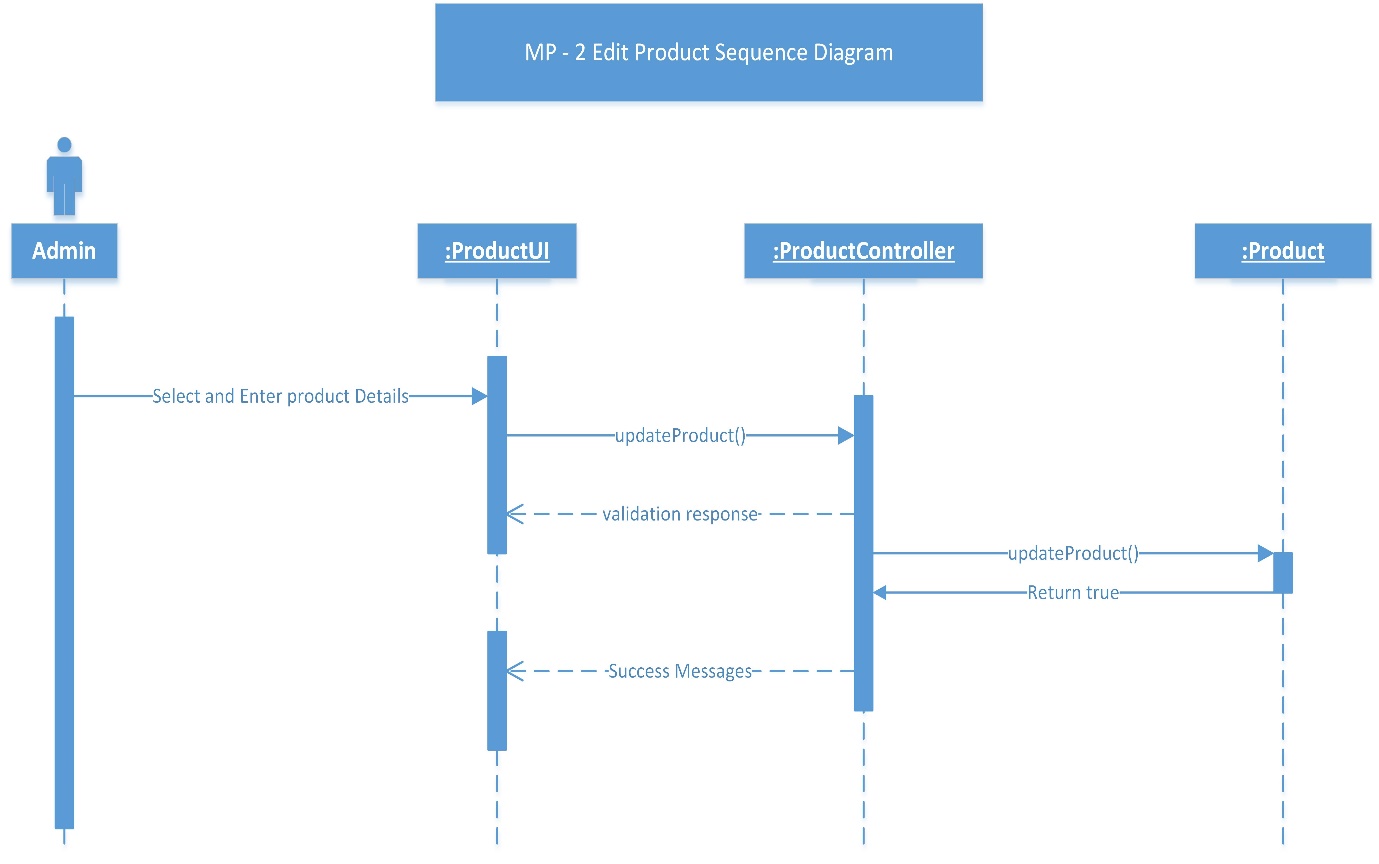
Remove Category  
  
Activity Diagram

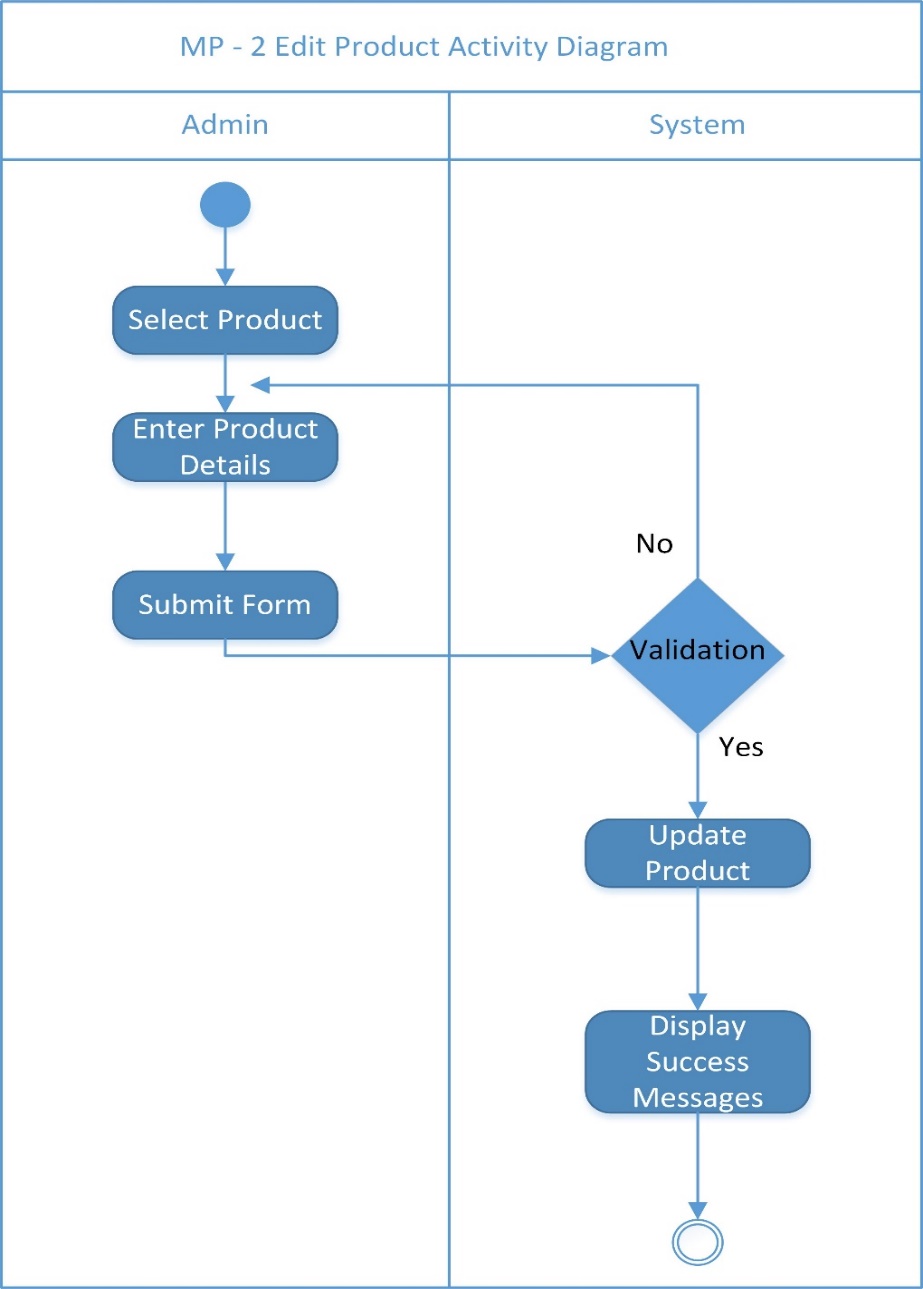
Add Product Sequence Diagram



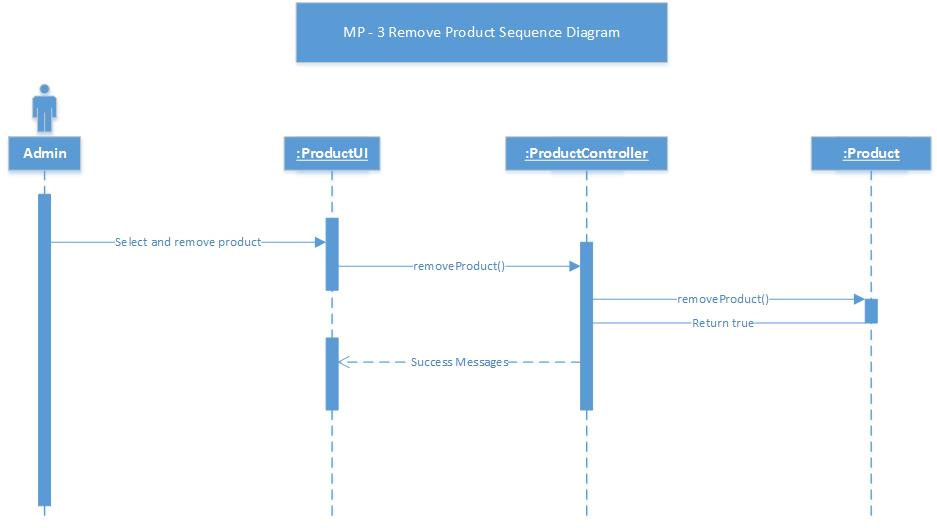


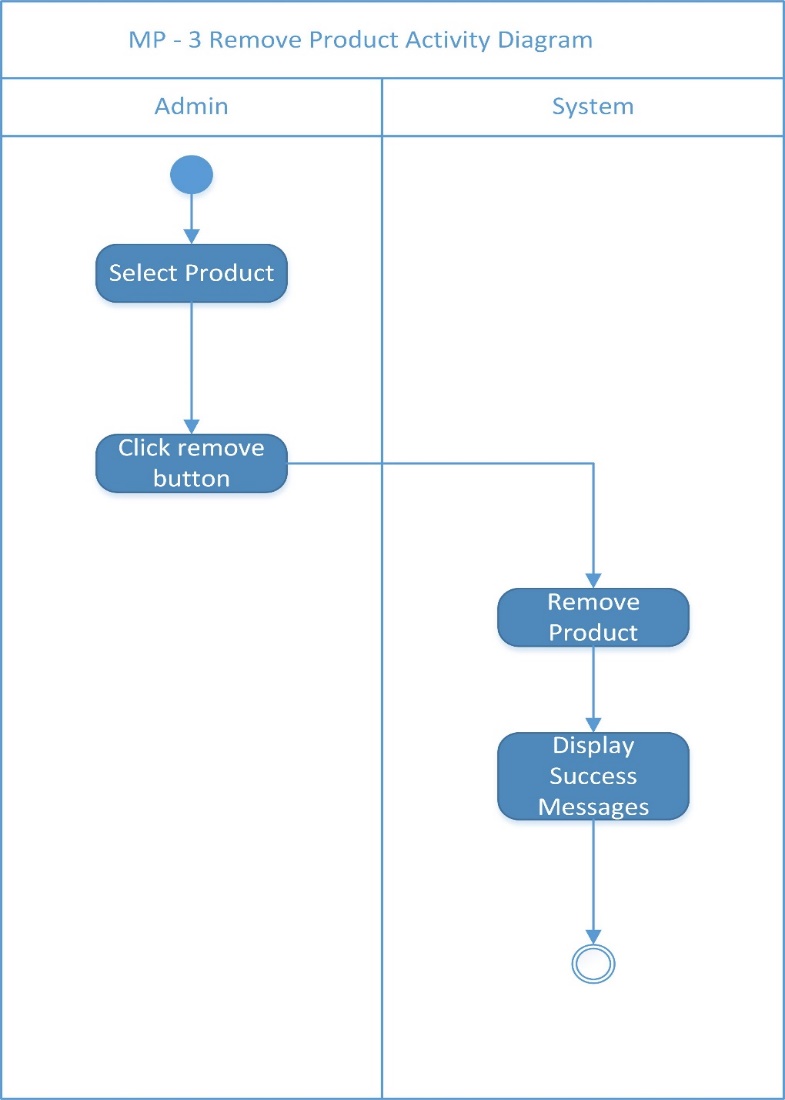
Add Product  
  
Activity Diagram

Edit Product Sequence Diagram  




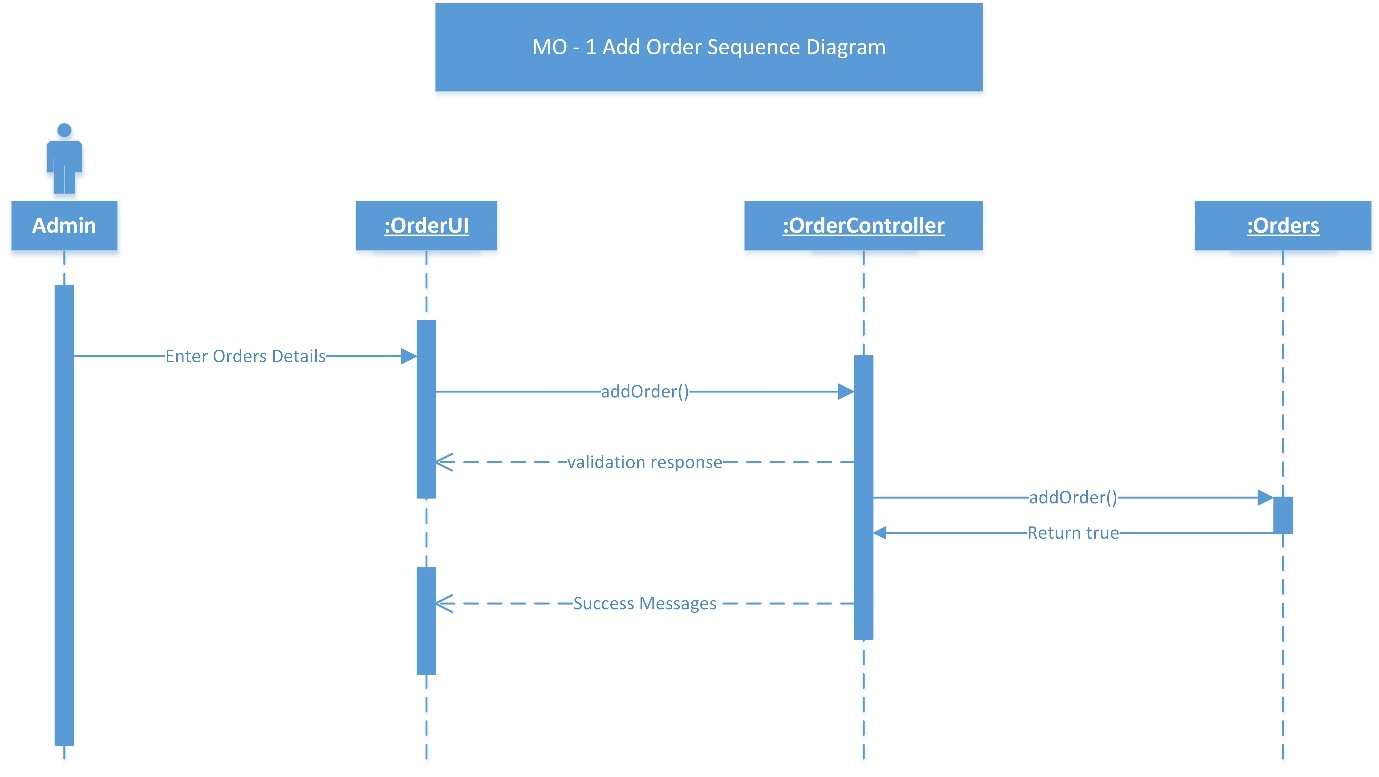
Edit Product  
  
Activity Diagram

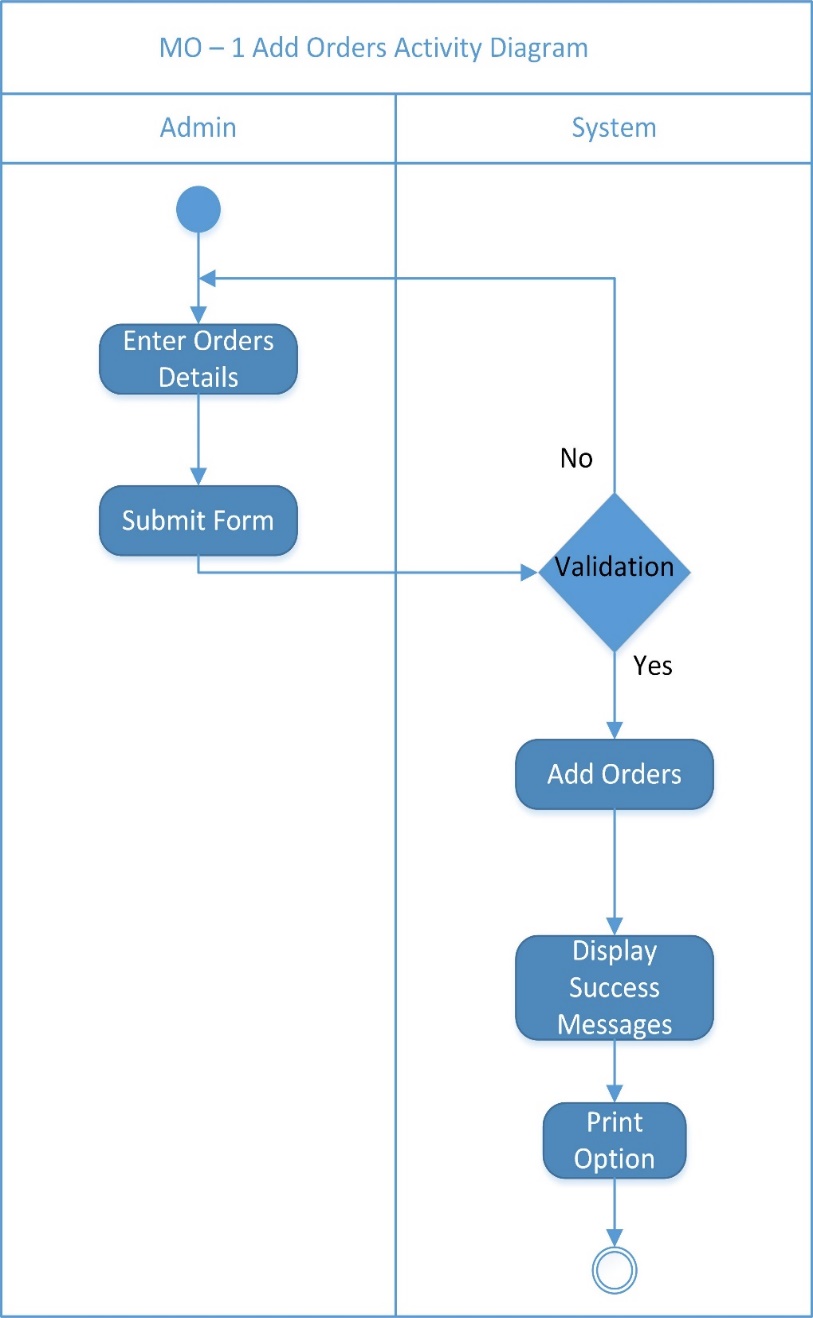
Remove Product Sequence Diagram  




Remove Product  
  
Sequence Diagram

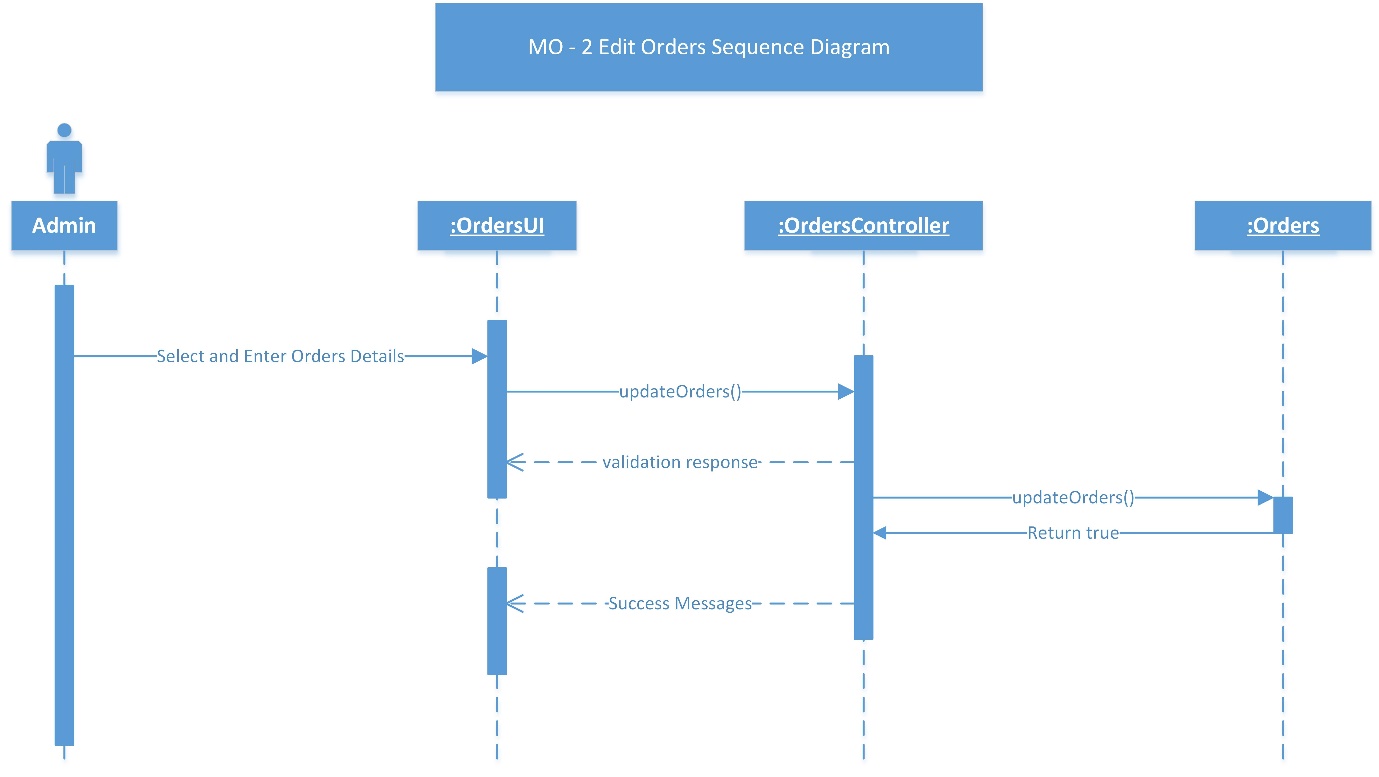
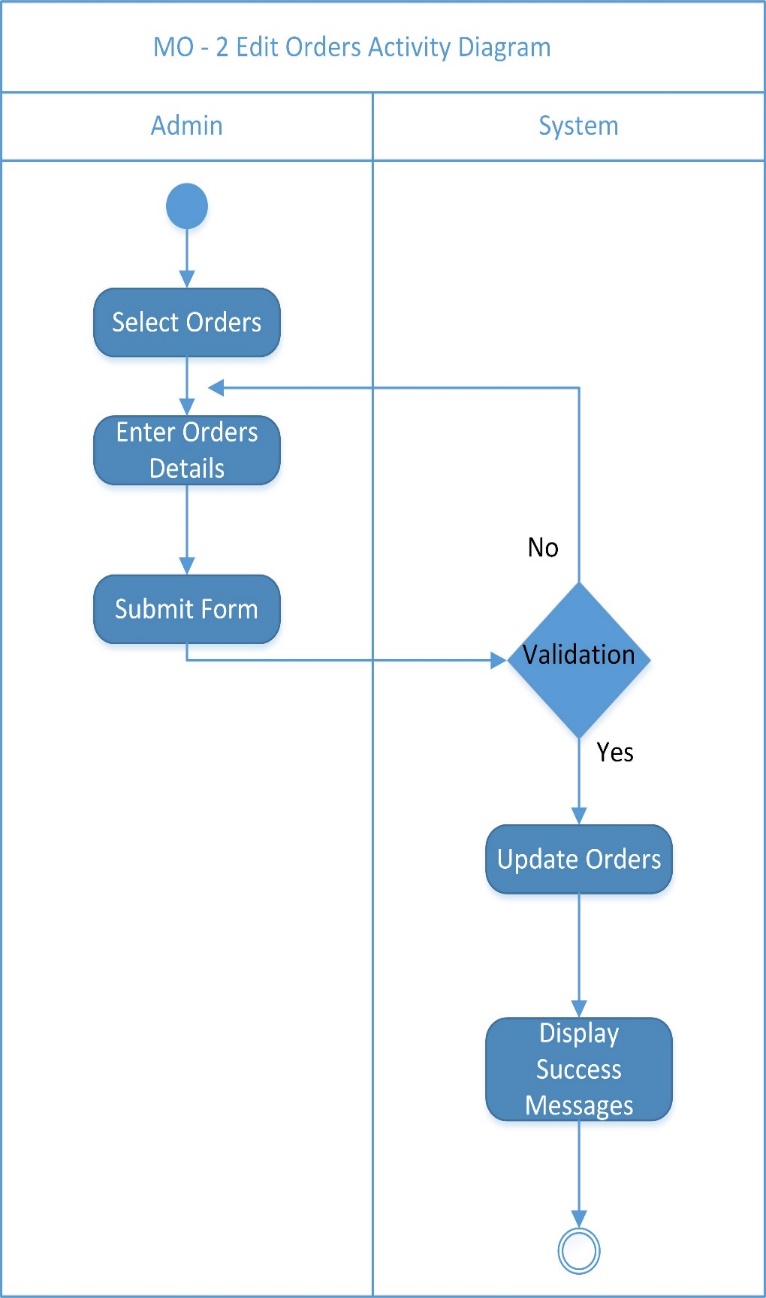
Add Orders Sequence Diagram





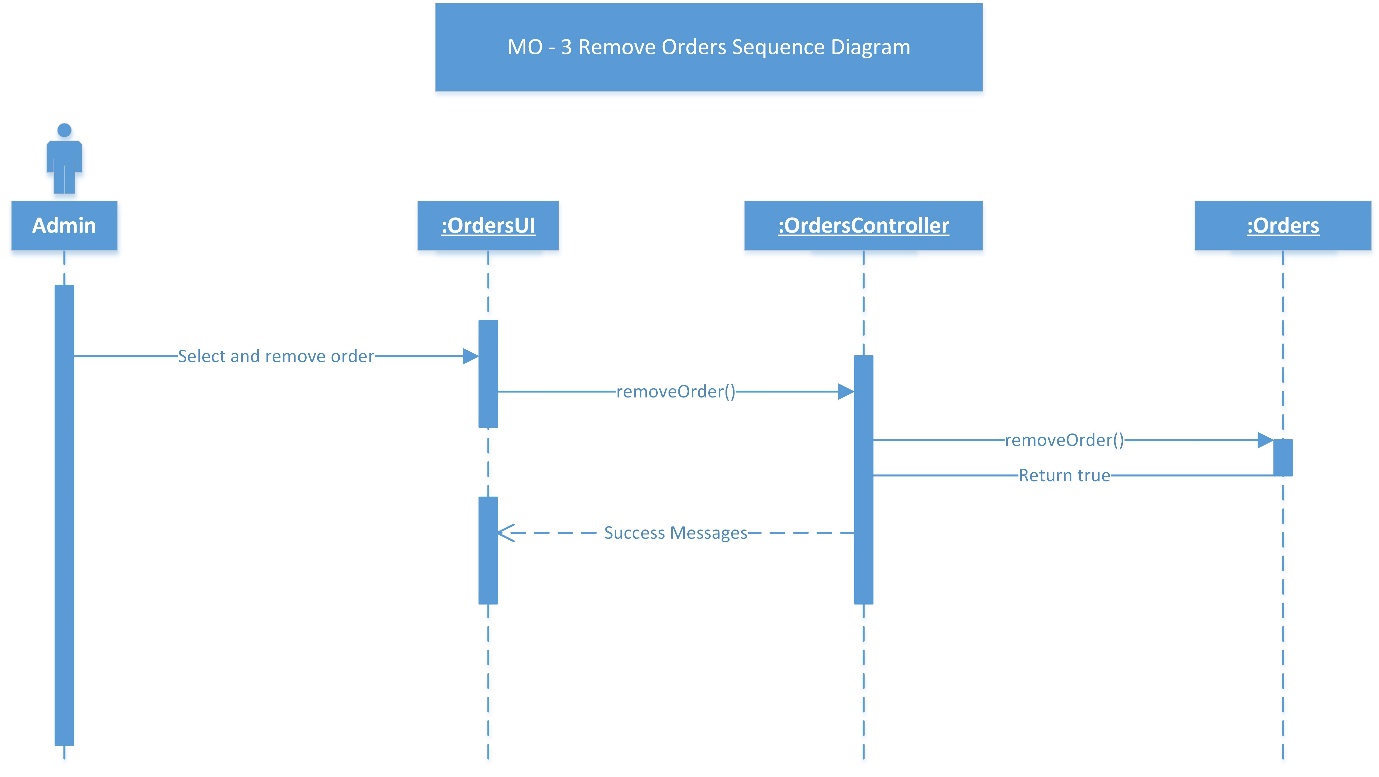
Add Orders  
  
Activity Diagram

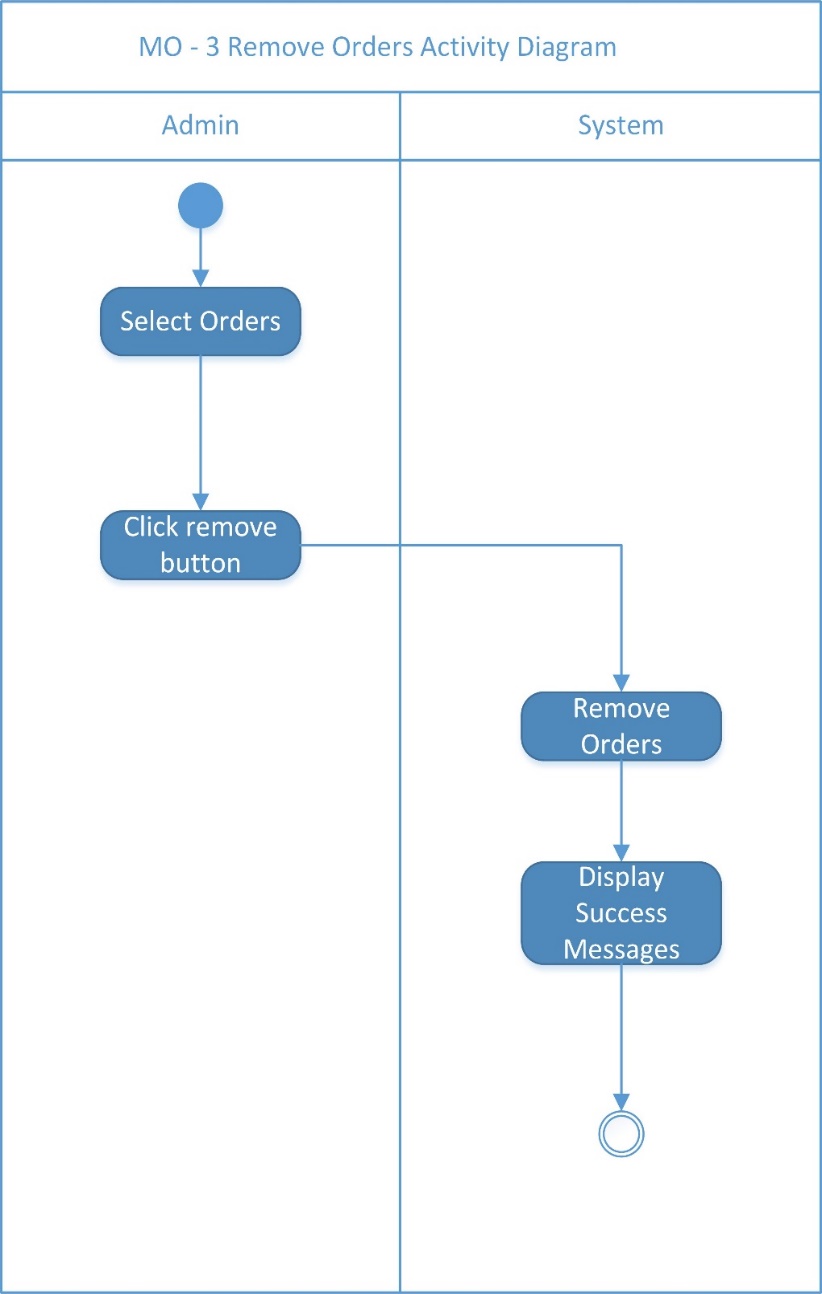
Edit Orders Activity Diagram



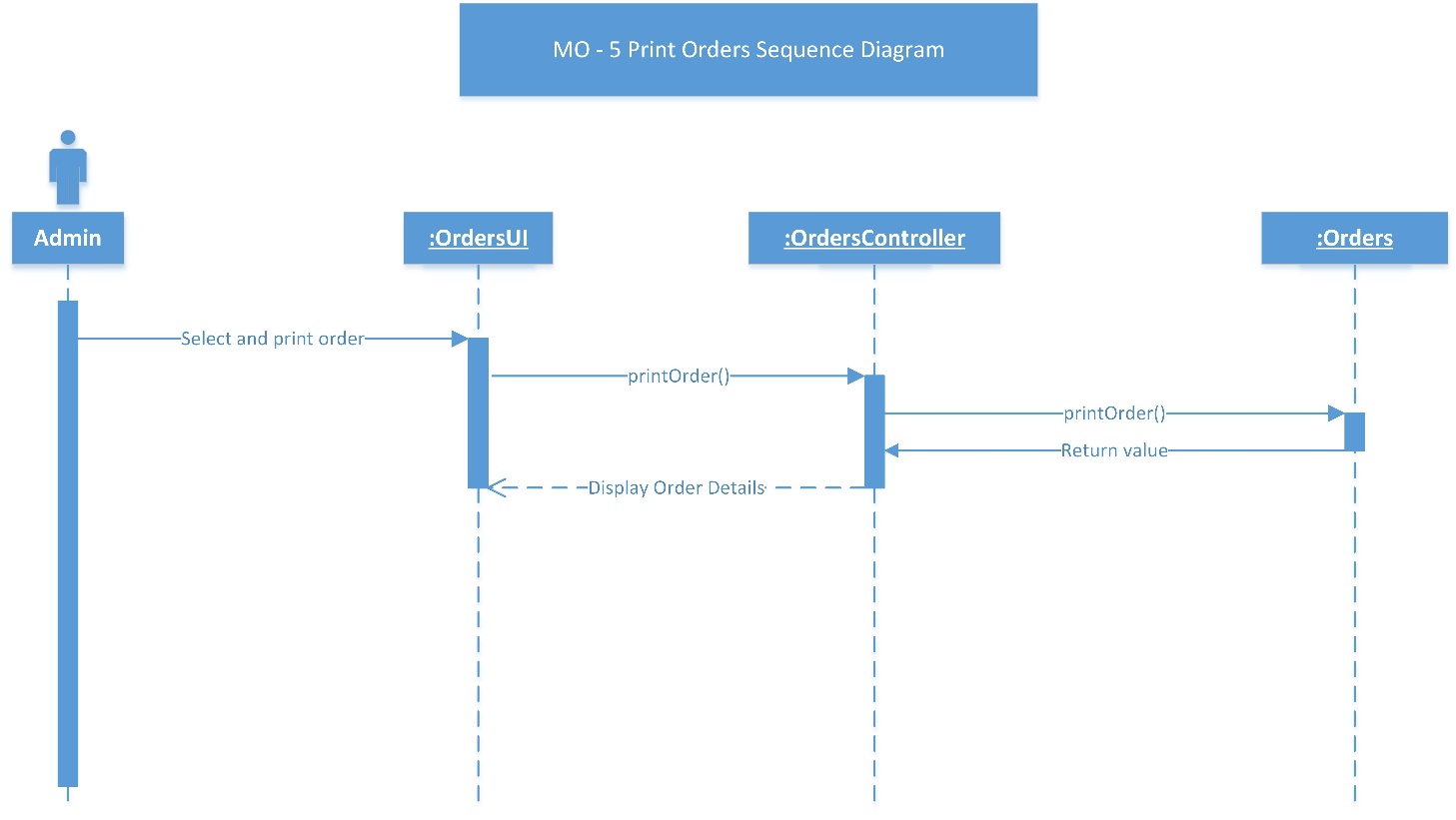
Edit Orders  
  
Activity Diagram

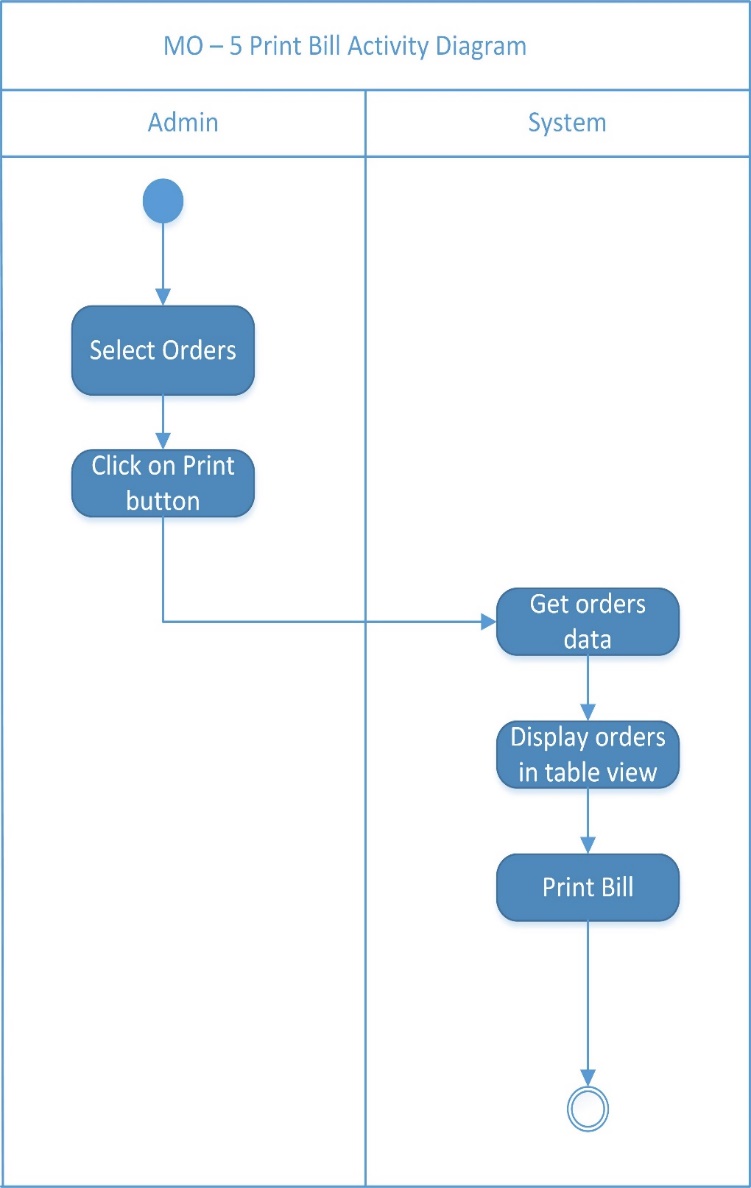
Remove Orders Activity Diagram



Remove Orders  
Activity Diagram

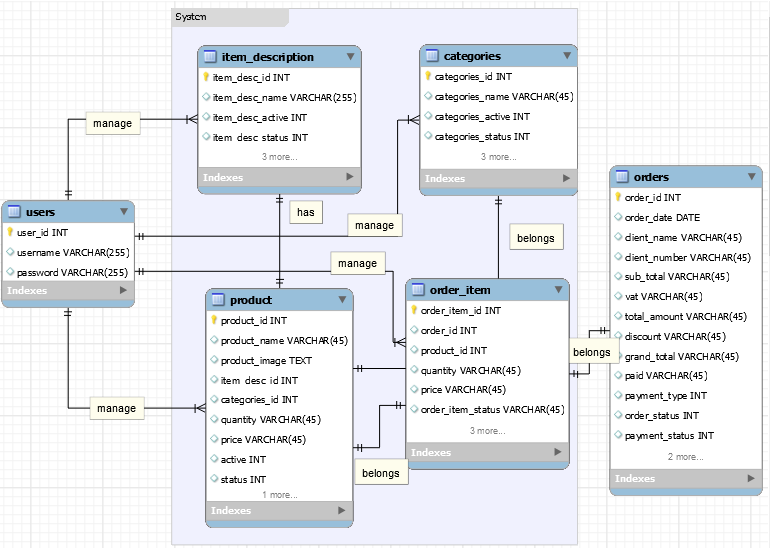
Print Order Sequence Diagram





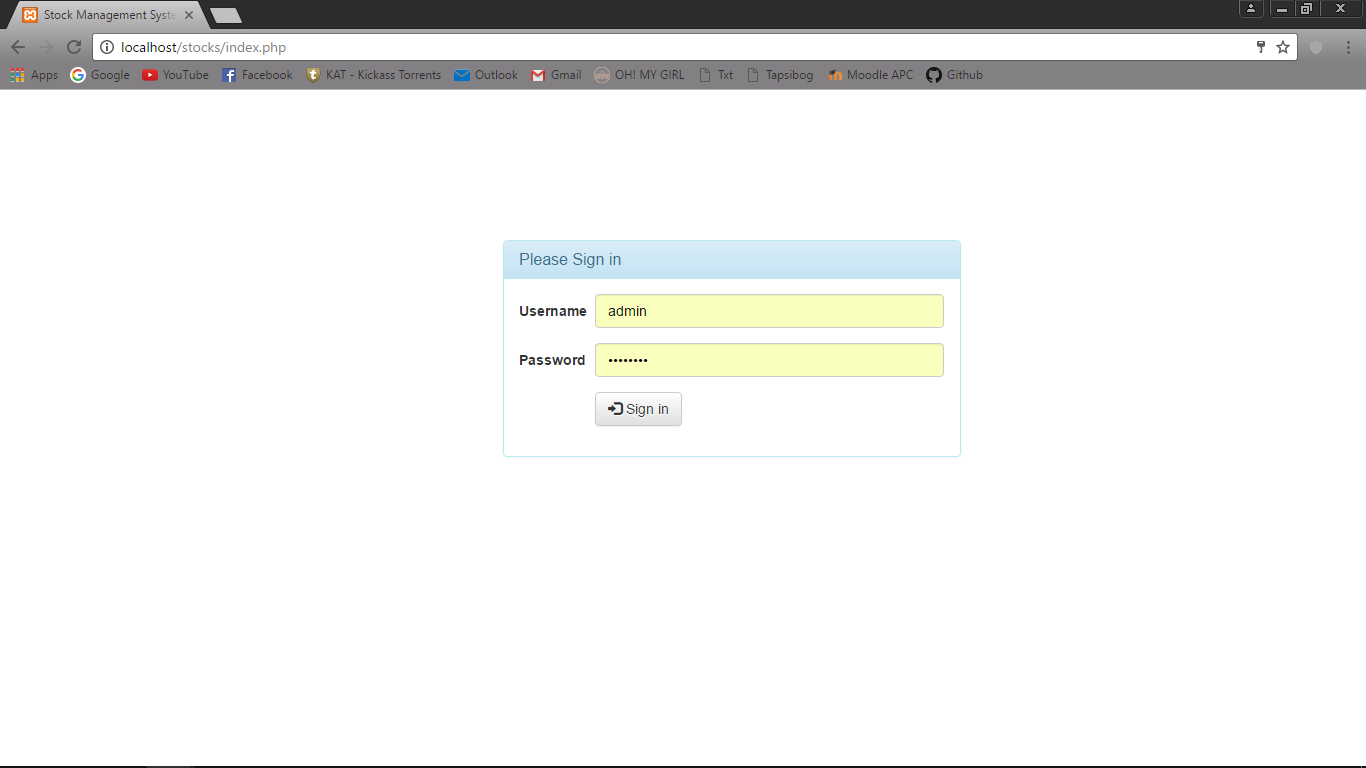
Print Order  
  
Activity Diagram

**ERD:**

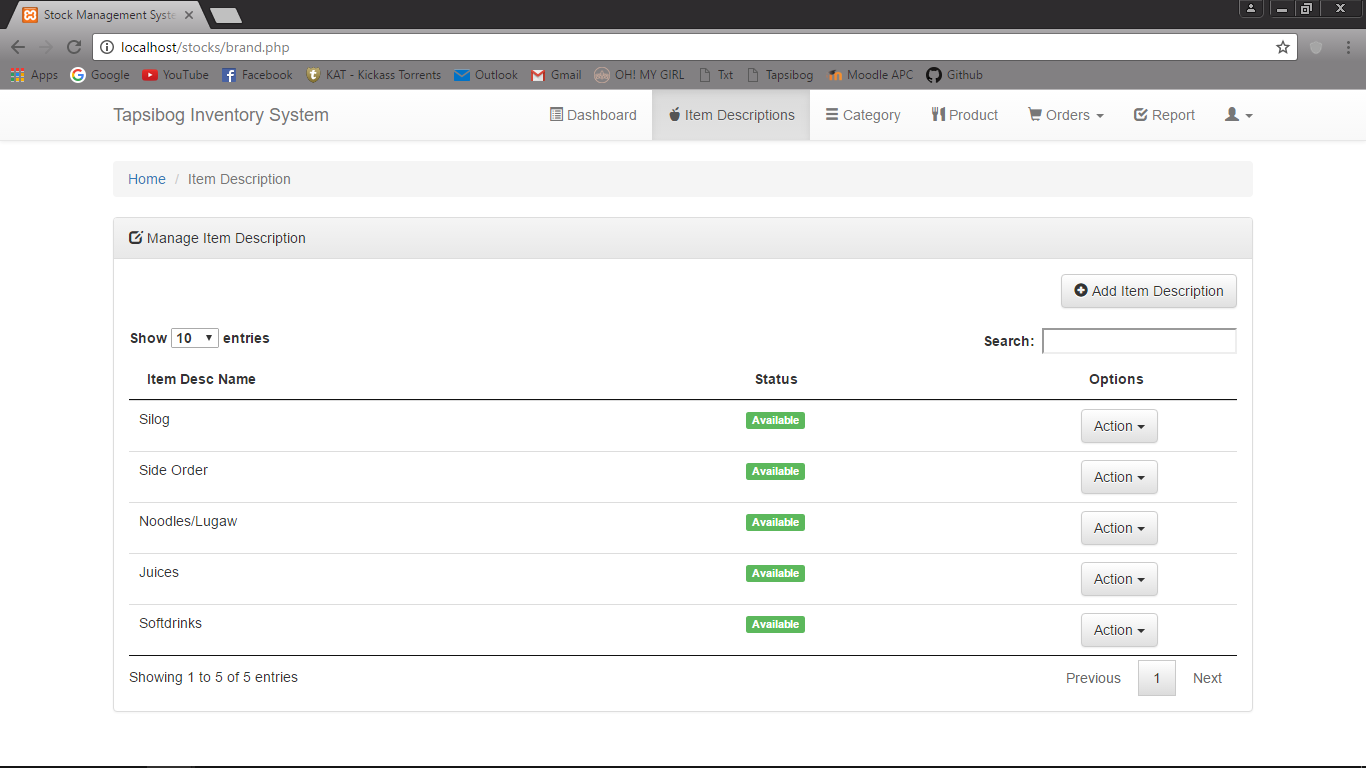


**SYSTEM SCREENSHOTS**

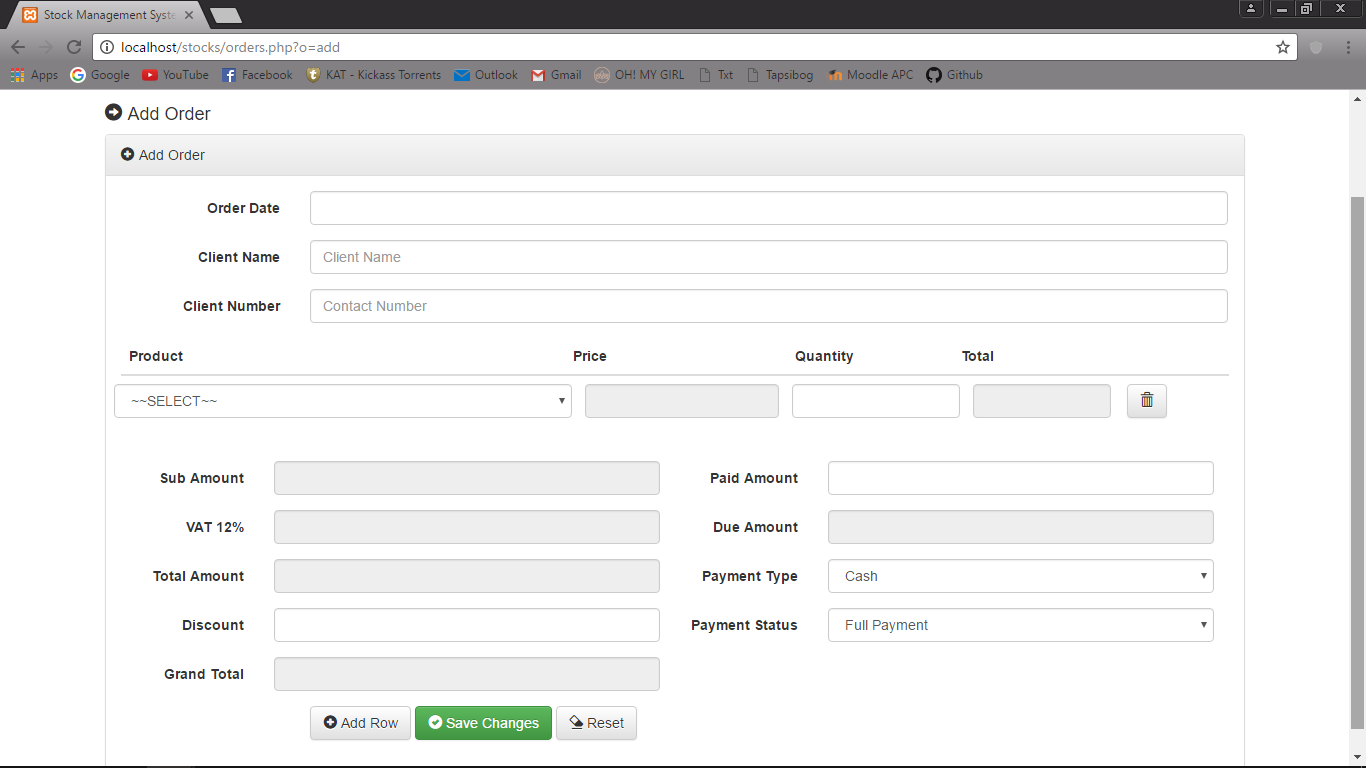
**Login**



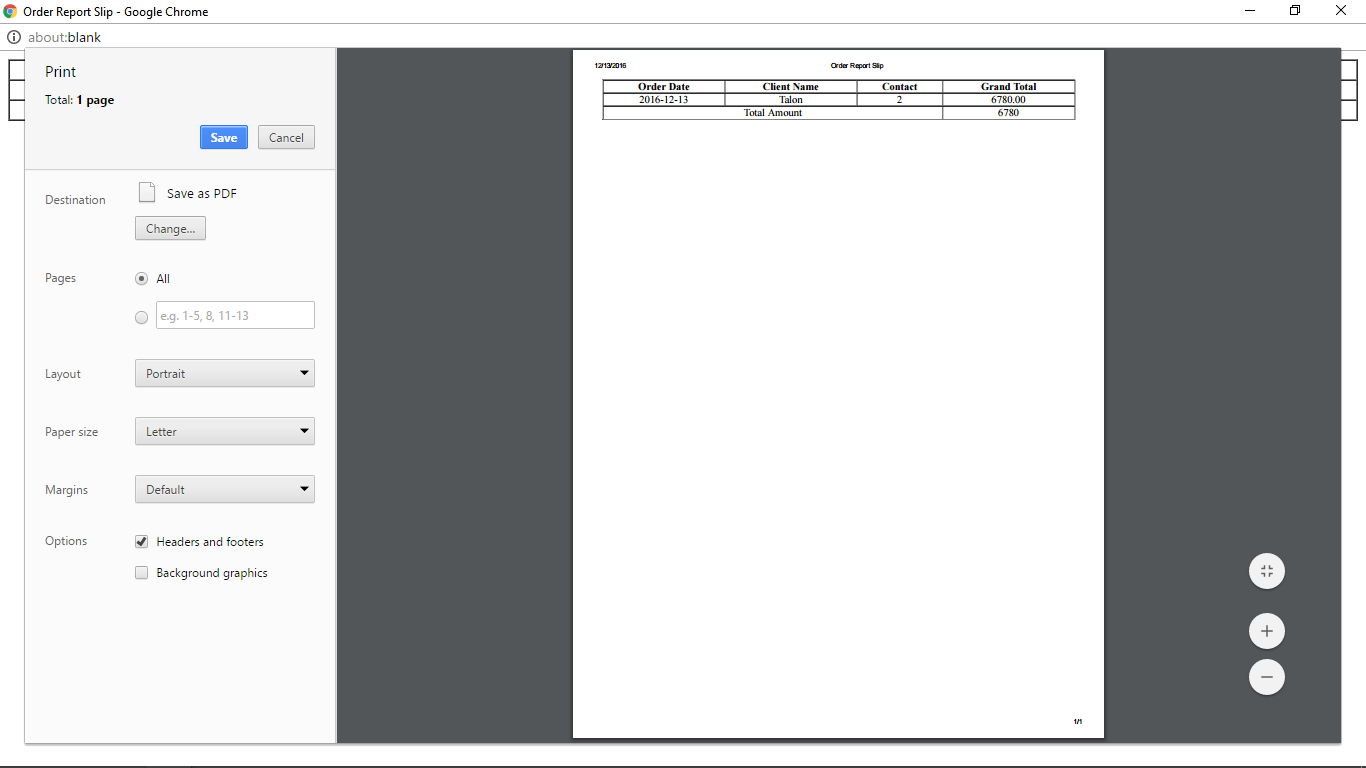
**Item Description page**



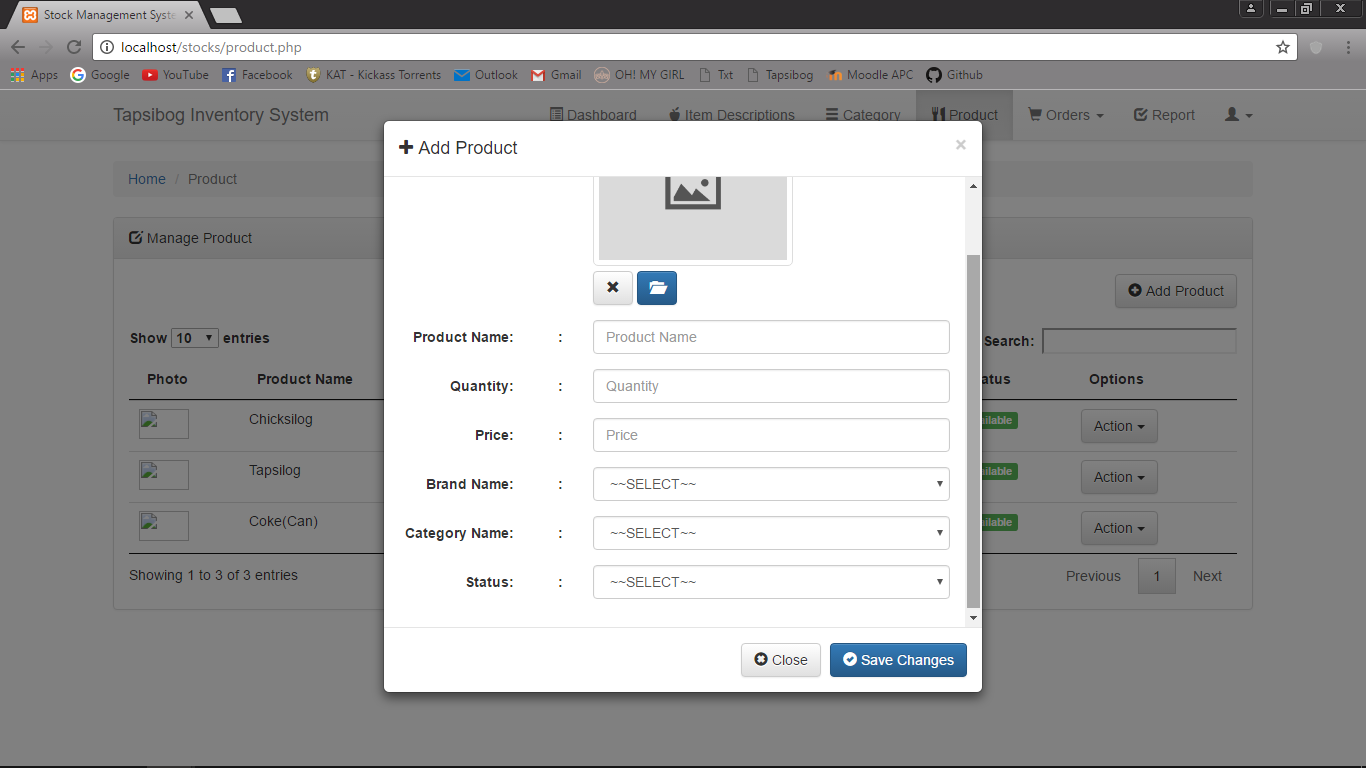
**Add Order**



**Print Report**



**Create New Product**



**Manage Orders**

