|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | Daffodil International University | | | | |  |  | |
|  |  | | |  | | |  | |  | |
|  | | | | | | | | | | |
| http://4.bp.blogspot.com/-SotLIkzmscA/UqQLE9BjdPI/AAAAAAAAARc/-SF46qFfVE0/s1600/daffodil-international-university-logo.png | | | | | | | | | | |
|  | | | | | | | | | | |
| Inventory Management System  Course Project  Software Engineering Project 2: Web Programming  Course Code: SWE331 | | | | | | | | | | |
|  | | NAME | | | **:** | Md. Sajib Mulla | | | |  |
| ID | | | **:** | 143-35-816 | | | |
| DEPARTMENT | | | **:** | Software Engineering | | | |
| Faculty of Science and Information Technology | | | | | | | |
|  | | |  |  | | | |
| December 2017 | | | | | | | | | | |

Table of Contents

[Chapter 1 1](#_Toc497901973)

[1. Introduction 2](#_Toc497901974)

[1.1 About the System 2](#_Toc497901975)

[1.2 Purpose 2](#_Toc497901976)

[1.3 Scope 2](#_Toc497901977)

[1.4 Vision 2](#_Toc497901978)

[1.5 Why this system is necessary? 2](#_Toc497901979)

[1.6 Proposed Solution 2](#_Toc497901980)

[Chapter 2 3](#_Toc497901981)

[2. System Analysis 4](#_Toc497901982)

[2.1 Use Case Model 5](#_Toc497901983)

[2.2 Actor Goal List 5](#_Toc497901984)

[2.2.1 Assumptions 5](#_Toc497901985)

[2.3 Use Case Model 5](#_Toc497901986)

[2.4 Use Case Description (Brief) 5](#_Toc497901987)

[2.4.1 Register Accounts (Example) 5](#_Toc497901988)

[2.5 Use Case Description (Detailed) 6](#_Toc497901989)

[2.5.1 Register Account (Example) 6](#_Toc497901990)

[2.6 System Sequence Diagrams 6](#_Toc497901991)

[2.6.1 Register Account 6](#_Toc497901992)

[2.7 Domain/Conceptual Model 7](#_Toc497901993)

[2.8 Activity diagram 8](#_Toc497901994)

[Chapter 3 9](#_Toc497901995)

[3. System Design 10](#_Toc497901996)

[3.1 Sequence Diagrams 10](#_Toc497901997)

[3.1.1 Register Account 10](#_Toc497901998)

[3.2 Class Diagram 11](#_Toc497901999)

[3.3 Entity Relationship Diagram 13](#_Toc497902000)

[Chapter 4 15](#_Toc497902001)

[4. Implementation 15](#_Toc497902002)

[4.1 Tools &Technologies 16](#_Toc497902003)

[4.2 Project Link 16](#_Toc497902004)

[Chapter 5 17](#_Toc497902005)

[5. System Testing 19](#_Toc497902006)

[5.1 Why Software Testing is Essential? 19](#_Toc497902007)

[5.1.1 Black box testing 19](#_Toc497902008)

[5.1.2 White box testing 19](#_Toc497902009)

[5.2 Test Cases 19](#_Toc497902010)

[5.2.1 Test case: Sign up 19](#_Toc497902011)

[Chapter 6 20](#_Toc497902012)

[6. Conclusion 21](#_Toc497902013)

[6.1 Good Features of the System 21](#_Toc497902014)

[6.2 Limitations of the System 21](#_Toc497902015)

[6.3 Future Enhancements 21](#_Toc497902016)

[Appendices 22](#_Toc497902017)

[7. User Manual 23](#_Toc497902018)

[7.1 Log in Screen of Federal Shariat Court Case Flow Automation System. 23](#_Toc497902019)

[8. References 24](#_Toc497902020)

Chapter 1

Introduction

# Introduction

## About the System

* In this proposal, we introduce a software for a shop that Calculate, Store Data, Shop Product items, Customer Info, Lender Info Etc. We developed this software for easily and safely Store Data and calculate total shop calculation. This project is complete system that help a shopkeeper for daily activity.

## Purpose

* Easily Know How much product have in a time
* Easily Calculate
* Save Customer Information
* Save lender information
* Save Daily Sale information
* Easily sale a product

## Scope

This system mainly create for shopkeeper. Shopkeeper easily can save daily activity on this system, like how was today condition on shop, how much sell, customer information, lander information, how much product have now in shop etc.

## Vision

## Why this system is necessary?

This system is necessary to reduce a shop information memories. For more easy to sale product and include product to shop. It’s necessary for easily control a shop.

## Proposed Solution

# 

Chapter 2

**System Analysis**

# System Analysis

## Use Case Model

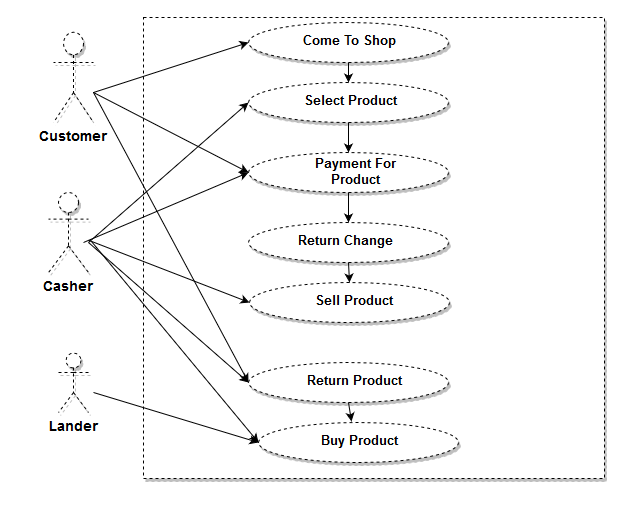


Fig 2.1:Use Case Diagram

## Actor Goal List

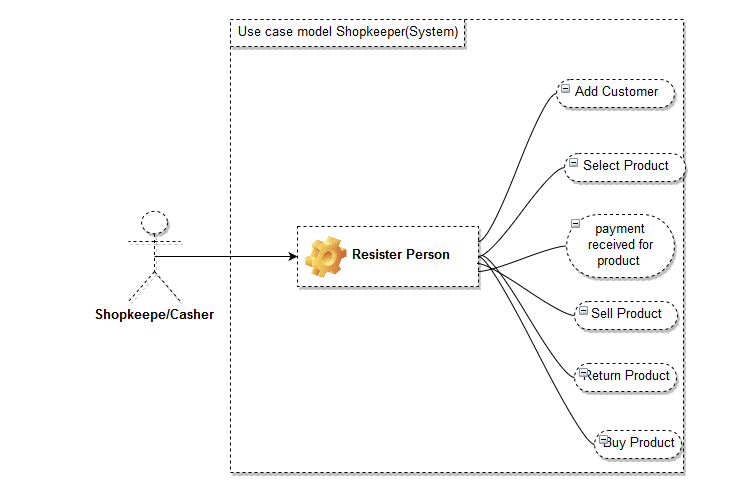
### Shopkeeper

* Login to this system by using User\_Name and Password.
* If customer come to buy product then shopkeeper cunt product which are selected by customer.
* Calculate the total product price and take payment for all product
* If customer back money return them
* If customer want to change or back product system can track it,
* Add product in shop which buy by Lander.

## Use Case Model

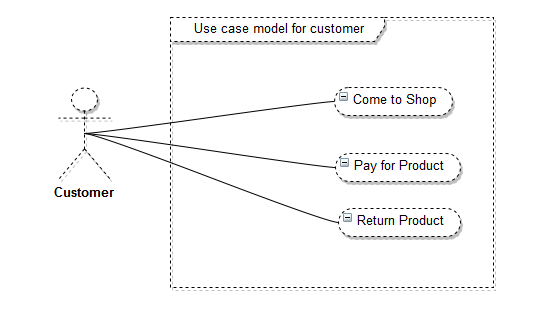
## Use Case Description (Brief)

### Shopkeeper/Casher



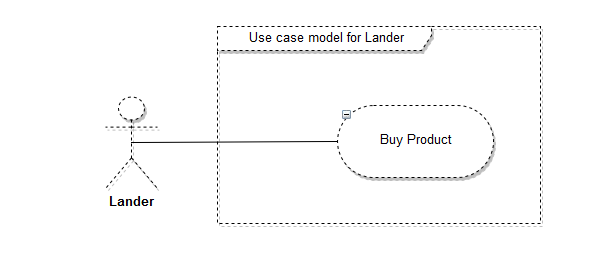
* In this phase who wants to use this system they should be resister by authority.
* Casher can add customer
* Select product for counting bill
* Take payment from customer
* Sell product which selected by customer
* Return product if customer want to back product
* Buy product for shop increase .

### Customer



* Customer have to come shop if they wants to buy somethings
* Then pay for selected product
* If product have any problem customer can change or return money.

### Lander



* Lander mainly he/she was the owner of this system. Who buy product for this shop.

## Use Case Description (Detailed)

### Register Shopkeeper

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | | 1 | |
| Name | | Shopkeeper | |
| Primary Actor | | Casher | |
| Goal | | Sale product and also add customer information. | |
| Precondition | | After Successfully join the system they fully control the system | |
| Post Condition | | Shopkeeper can change Customer information and shop inventory. | |
| Main Success Scenario | **Actor** | | **System** | |
| 1. After login the system shopkeeper can add customer 2. Calculate bill 3. Make recite for product 4. Able to see all customer 5. Able to buy product for shop and add to shop | | 1.1) System return successful message to user  2.1) System return message when new product add | |
| Failure Scenario | 1) Don’t able to login.  2) Don’t able to add customer.  3) Connection problem to database  4) Don’t save new product to database | | | |

### Customer

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | | 2 | |
| Name | | Customer | |
| Primary Actor | | Customer | |
| Goal | | Select product to buy. | |
| Precondition | | Customer have to come shop to buy product.  Pay for selected product | |
| Post Condition | | After buying product take recite for total information | |
| Main Success Scenario | **Actor** | | **System** | |
| 1. When come to shop customer can see which product available now 2. Can return product 3. Invoice copy | | 1.1) System show product now available  2.1)After complete buy system give a invoice copy . | |
| Failure Scenario | 1) Don’t able connect database  2) Don’t show available product.  3) system loss previous record | | | |

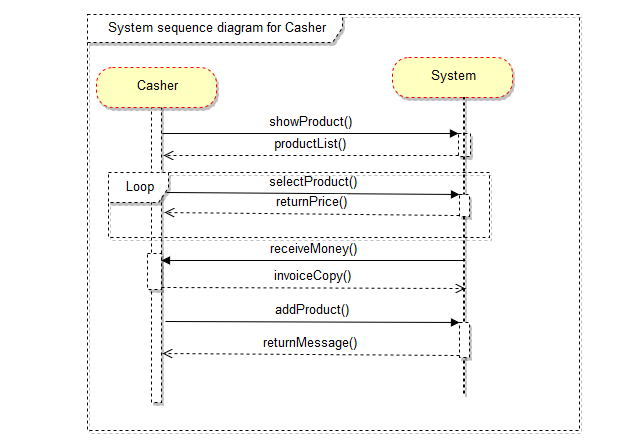
### Lander

|  |  |
| --- | --- |
| Use Case ID | 3 |
| Name | Lander |
| Primary Actor | Lander |
| Goal | To add product and monitor the shop. |
| Precondition | Have to buy product to add shop |
| Post Condition | Take a confirmation message that product add successfully |
| Main Success Scenario | 1. Product add system 2. System give information that which product below now |
| Failure Scenario | 1. Can’t add product to customer  2. Add product system can’t add product. |

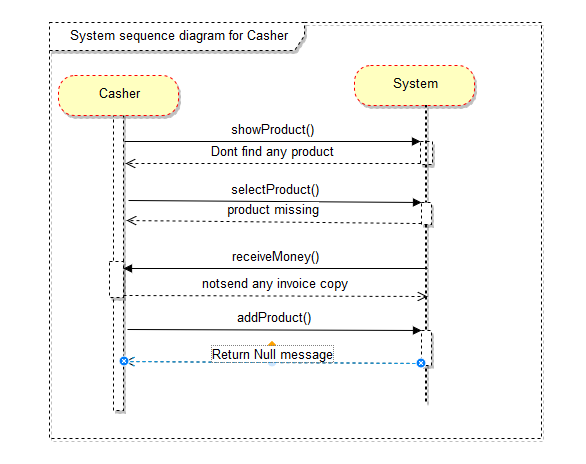
## System Sequence Diagrams

### Register Shopkeeper

#### Success scenario

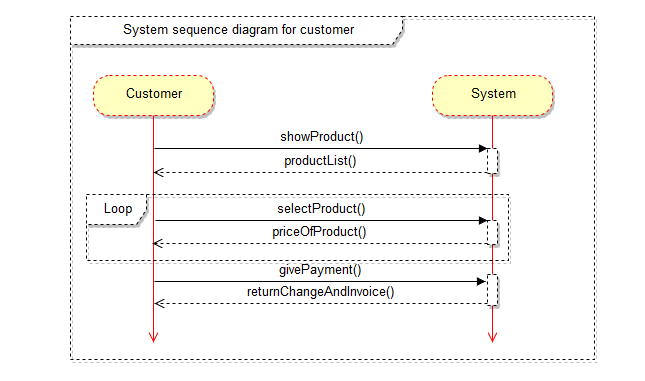


**2.6.1.2 Failure Scenario**

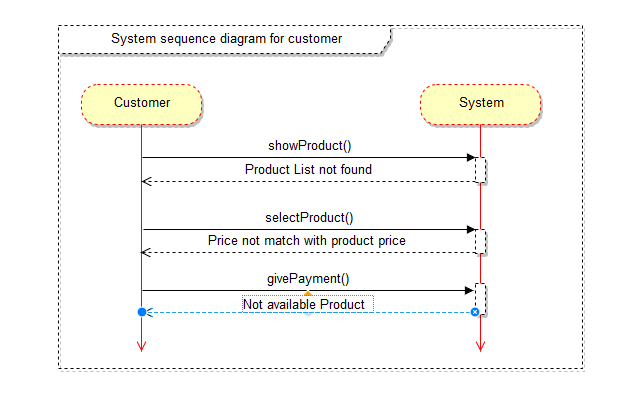


**2.6.2 Castomer**

**2.6.2.1 Success scenario**

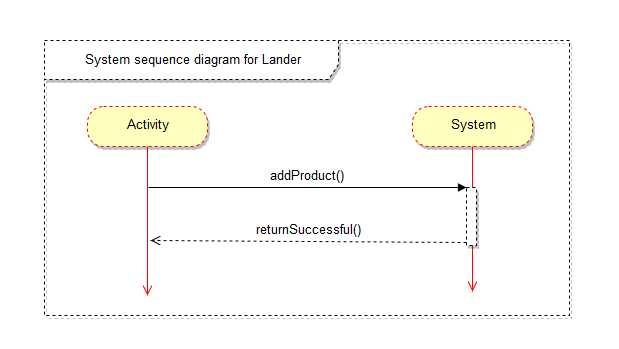


**2.6.2.2 Failure scenario**

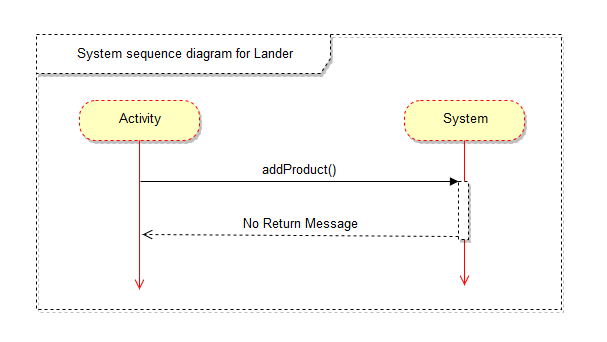


**2.6.3 Lander**

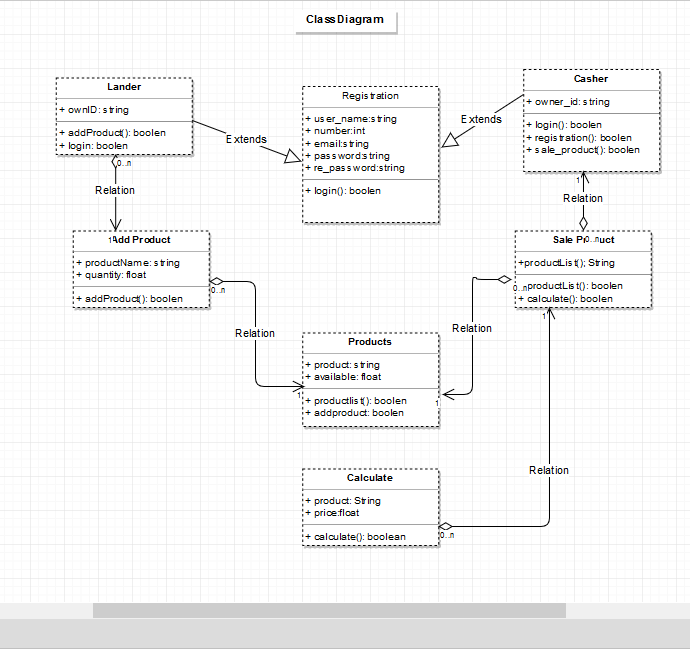
**2.6.3.1 Success scenario**



**2.6.3.2 Failure scenario**

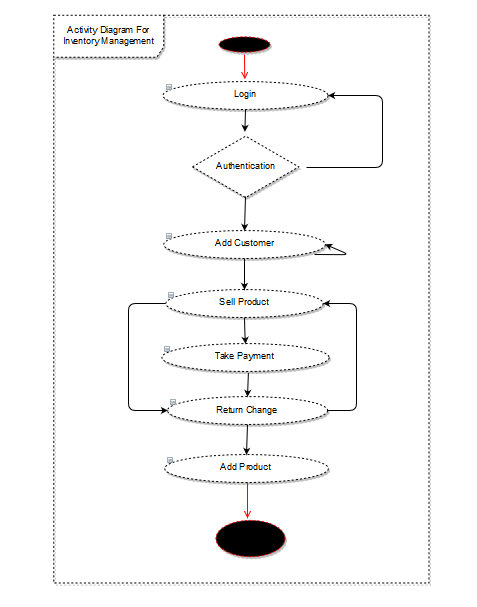


## Domain/Conceptual Model



**Figure 2.17 System Domain Model**

## Activity diagram

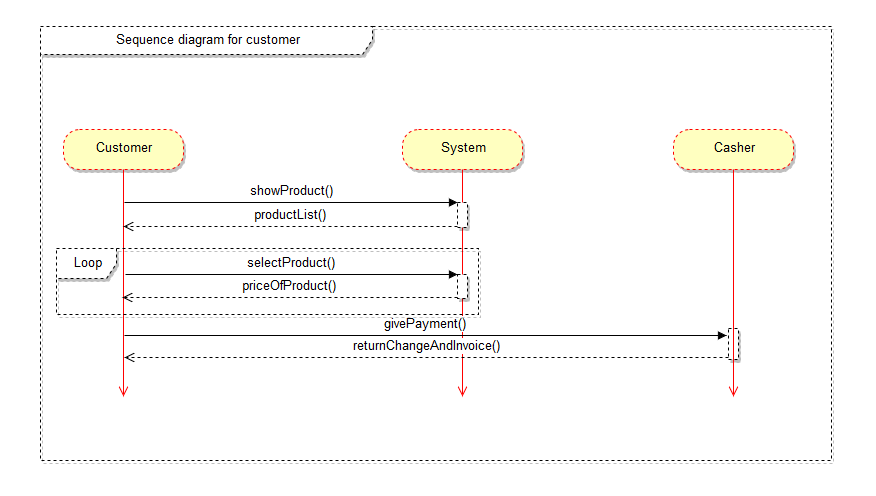


Chapter 3

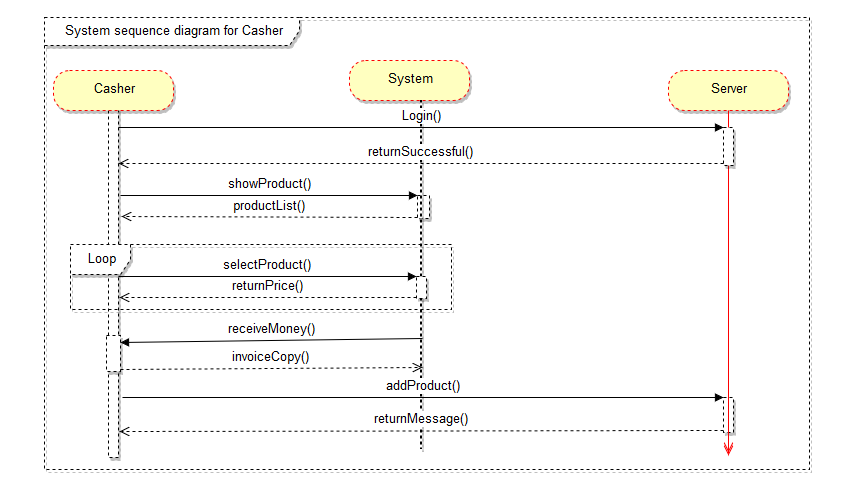
**System Design**

**3.1) Sequence Diagram**

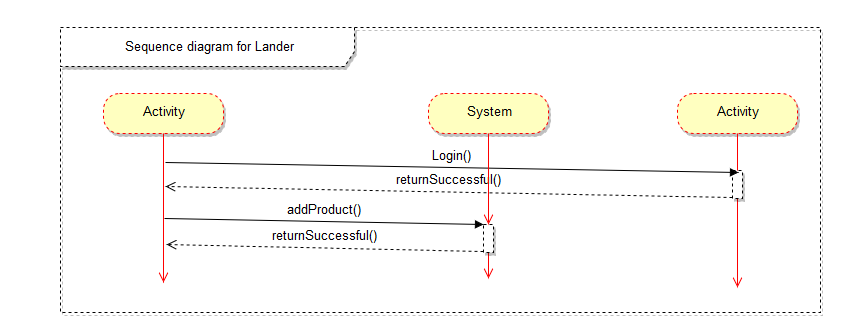
**3.1.1) Shopkeeper:**



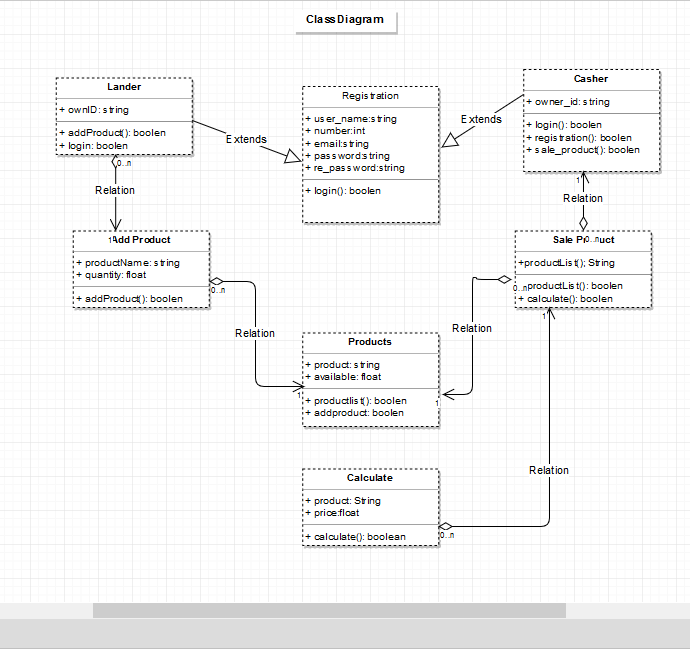
**3.1.2) Casher:**



* + 1. **Lander**



**3.2) Entity Relationship diagram:**



Chapter 4

**Implementation**

# Implementation

Implementation (software) perspective describes software implementations in a particular technology (such as C#). In the UP, Implementation means programming and building the system, not deploying it.

In the implementation phase, the developer builds the components either from scratch or by composition given the architecture document from the design phase and the requirement document from the analysis phase. The architecture document should give guidance. Sometimes, this guidance is found in the requirement document. The implementation phase deals with issues of quality, performance and debugging. The end deliverable of implementation phase is the product itself.

## Tools &Technologies

Following are the tools and technologies used in development of this project:

Microsoft Visual Studio 2012

ASP.NET Framework

Microsoft SQL Server 2008

Telerik reports

Microsoft Visio

Microsoft Architect

HTML5, CSS, JavaScript, J-query, Twitter bootstrap

## Project Link

They provide github repository link after uploading the project there.

**References:**

https://makebills.pro/