

# **MOBILE BASED INVENTORY MANAGEMENT SYSTEM**

**Prepared By:**

**MD.SABBIR HASAN TONMOY**

**ID: 112-35-205**

**5<sup>th</sup> Batch**

Department of Software Engineering  
Daffodil International University

**Supervised By:**

**FARZANA SADIA**

**Lecturer**

Department of Software Engineering  
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**Dhaka, Bangladesh**

**April 2016**

## APPROVAL

This Report titled “**MOBILE BASED INVENTORY MANAGEMENT SYSTEM**”, submitted by Md. Sabbir Hasan Tonmoy ID No: 112-35-205 to the Department of Software Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Software Engineering and approved as to its style and contents.

### BOARD OF EXAMINERS

  
-----

**Dr. Touhid Bhuiyan**

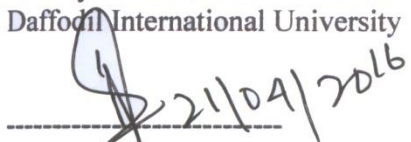
Associate Professor

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

**Head**

  
-----

**Dr. Md. Asraf Ali**

Associate Professor

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

**Internal Examiner 1**

  
-----

**Rubaida Easmin**

Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

**Internal Examiner 2**

-----  
**Dr. Nasim Akhtar**

Professor, Head of the Department

Department of Computer Science & Engineering


Faculty of Electrical and Electronic Engineering

Dhaka University of Engineering & Technology, Gazipur

**Examiner**

## DECLARATION

We hereby declare that this project report entitled “MOBILE BASED INVENTORY MANAGEMENT SYSTEM” is the record of my original work under the supervision of Farzana Sadia, Lecturer, Department of Software Engineering Daffodil International University. We also declare this project or any part of this report has never been submitted anywhere else for award of any degree or diploma.

 20.04.2016

---


**MD.SABBIR HASAN TONMOY**

**ID: 112-35-205**

**5<sup>th</sup> Batch**

Department of Software Engineering  
Daffodil International University

Certified By:

 20/4/2016

---

**FARZANA SADIA**

**Lecturer**

Department of Software Engineering  
Daffodil International University

## **ACKNOWLEDGMENTS**

First of all I would like to express my cordial gratefulness to Almighty Allah for HIS kindness, for which I successfully completed my project with in time. While working on this project I have received many invaluable help form a large number of people. I would like to take this opportunity to express my deepest gratitude to everyone have helped us. I fell grateful to express my boundless honor and respect to my supervisor, Farzana Sadia, Lecturer, Department Of Software Engineering, Daffodil International University for her Deep Knowledge & keen interest in the field of Software Development that influenced me to carry out of this project. Her endless patient help, friendly support, great enthusiasm and extensive knowledge, which have guided me throughout our work and showed the path of achievement. I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work. And at last but not the least I must acknowledge with due respect the constant support and patience of our family members for completing this project.

## ABSTRACT

This project entitled “**Mobile Based Inventory Management System**” is mobile based software that will work as a based on auto parts information of a small auto parts shop. The main purpose of the project is to store products, view stored products information, sale products and view sales information. The System store all products information and sales information. After that Shop Owner can be update product, delete product, search product by product id and product name and also sort product by product name. View information as per their needs. The modern technology is used to implement the software to make it more secured. Some client side scripts have also been used to make it user friendly.

## **Table of Contents**

<b>Chapter 1</b> .....	1
Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	2
1.4 Scope.....	2
<b>Chapter 2</b> .....	3
Pre-Analysis and Initiation.....	3
2.1Scenario.....	3
2.2Requirement collection.....	3
2.2.1 Client Requirements.....	4
<b>Chapter 3</b> .....	5
Analysis.....	5
3.1 Requirements analysis .....	5
3.1.1 Requirements Elicitation.....	5
3.2 Software Requirement Analysis.....	7
3.2.1 Objective/Goals .....	7
3.2.2. Software Deliverables.....	8
3.2.3. Software Requirement Specification.....	8
3.2.4Description of Specific Requirements.....	9
3.3 Software Development Plan (SDP).....	12
3.3.1Overview.....	12
3.3.1.1 Proposed System and Its benefits.....	12
3.3.1.2 Features of the system.....	12
3.3.2 Project Organizations.....	13
3.3.2.1 Development Methodology.....	13
3.3.2.2 Responsibilities .....	14
3.3.3 Managerial Process .....	14
3.3.3.1 Management Objectives .....	14
3.3.3.2 Monitoring and Controlling Mechanism .....	14
3.3.3.3 Risk Management .....	15

3.3.3.4 Tools and Technology.....	15
3.3.4 Work Schedule .....	16
3.3.5Grant Chart....	16
3.4 Software Test Plan.....	17
3.4.1 Introduction .....	17
3.4.2 Test Items .....	17
3.4.3 Features to be tested.....	18
<b>Chapter 4</b> .....	19
System Designing.....	19
4.1 High Level Design (HLD) .....	19
4.1.1 Component Diagram .....	19
4.1.2 Architecture diagram of the system.....	20
4.1.5Activity Diagram .....	21
4.1.6Context Diagram (DFD Level-0).....	25
4.2Detail Level Design.....	26
4.2.1 Data Flow Diagram Level-1.....	26
4.2.2Entity Relationship Diagram.....	28
4.2.3Sequence Diagram.....	28
<b>Chapter 5</b> .....	29
System Test and Completion.....	30
6.1. Unit Test Case.....	30
6.1.1Introduction.....	30
6.1.2Importance of Testing.....	30
6.1.3Unit Case Test (UTC).....	31
6.1.4 Unit Test Report.....	37
6.2 System Test Report.....	38
6.3 Software Development Completion Report (SDC).....	38
6.3.1BasicOverview.....	39
6.3.2 Project Outcomes .....	39
6.3.3 Project Achievements.....	39
6.3.4 Project maintenance .....	40
6.3.5Conclusion.....	40
6.3.5.1Remarks of the work.....	40
6.3.5.2 Limitation.....	40

6.3.5.3 Future work.....	40
<b>Chapter 7: References.....</b>	<b>41</b>
<b>Chapter 8: Appendixes.....</b>	<b>42</b>



## List of Figures

Chapter 3.....	5
Figure 01: Admin Login.....	9
Figure 02: Manage product.....	10
Figure 03: Manage Sale Product.....	11
Figure 04: Agile-scrum model.....	13
Figure 05: Grant Chart.....	16
 Chapter 4.....	 19
Figure 06: System component diagram.....	19
Figure 07: System Architecture Diagram.....	20
Figure 08: Activity diagram –Admin login.....	21
Figure 09: Activity diagram – Add new product.....	22
Figure 10: Activity diagram – Search, update and delete product.....	23
Figure 10: Activity diagram – Manage Sale product.....	24
Figure 11: System Context diagram.....	25
Figure 12: DFD 1.1.....	26
Figure 13: DFD 2.1.....	27
Figure 14: DFD 3.1.....	27
Figure 15: Entity Relationship diagram.....	28
Figure 16: Physical Database.....	28
Figure 17: Sequence diagram.....	29

## List of table

CHAPTER 2.....	3
Table 01: Description about user.....	3
CHAPTER 3.....	5
Table 02: Functional Requirement.....	7
Table 03: Non Functional Requirements.....	8
Table 04: Risk management.....	15
Table 05: Scheduling& Planning .....	16
Table 06: Feature to be tasted.....	18
CHAPTER 5.....	30
Table 08: Admin login.....	31
Table 08: Add new product.....	32
Table 09: Search product.....	33
Table 10: Delete product... ..	34
Table 11: Update product.....	35
Table 12: Display product.....	35
Table 13: Sale product.....	36
Table 14: Display sale product.....	37
Table 15: Unit test report.....	37
Table 16: System test report.....	38
Table 16: Software development completion report.....	38

# CHAPTER 1

## INTRODUCTION

---

### 1.1 Purpose

This project is aimed at developing mobile based Inventory Management System for a small auto parts shop. This system can be used to store the details of the inventory, update the inventory based on the stored details, Sale products and view sales information. This system used by the administrators for performing admin level functions such as adding new items to the inventory, update items to the inventory and delete items, search item by id and name, sorting product etc.

### 1.2 Document Conventions

#### Main Chapter

- Font: Times New Roman
- Face: Bold
- Size: 18

#### Main Section Titles

- Font: Times New Roman
- Face: Bold
- Size: 16

#### Sub Section Titles

- Font: Times New Roman
- Face: Normal
- Size: 14

#### Other Text Explanations

- Font: Times New Roman
- Face: Normal
- Size: 12

## **1.3 Intended Audience and Reading Suggestions**

This document describes the project scope for software developers, users and testers to understand the basic system architecture.

Developer: A developer must have read the documents to modify, update any valid or necessary requirements. Project developer has an opportunity to understand the system.

## **1.4 Scope**

This project helps to improve the current sales and inventory management process of the shop thus eliminating the usage of so much manual process. This system can easily keep track of how much products in on hand real time. This system can also sale products and view sales information the requested parameters or based on the users criteria.

## 2.1 Scenario

Scenario is a process used during requirement analysis to describe the use of proposed system. Scenario is a story about proposed System scenario explains an early stage of proposed system.

The scenario of this system is given below:

The system gives **Mobile Based Inventory Management System** documentation. There are only one users that is Admin.

User	Description
Admin	Admin can login to the system through username and password. Admin can add new product to the inventory, update product and delete product from inventory. Search for a particular product by product id and product name. Admin can view the status of inventory. Sale products and view sales information. Sort product by product name.

Table 01: Description about user

## 2.2 Requirement Collection

Requirements gathering are an essential part of any project and project management. Understanding fully what a project will deliver is critical to its success. This may sound like common sense, but surprisingly it's an area that is often given far too little attention. Many projects start with the barest headline list of requirements, only to find later the customers' needs have not been properly understood. One-way to avoid this problem is by producing a statement of requirements. This document is a guide to the main requirements of the project. It provides:

- ☐ A succinct requirement specification for management purposes.
- ☐ A statement of key objectives - a "cardinal points" specification.
- ☐ A description of the environment in which the system will work.
- ☐ Background information and references to other relevant material.
- ☐ Information on major design constraints.

Finally, ensure you have cross-referenced the requirements in the statement of requirements with those in the project definition report to ensure there is no mismatch. The functional requirements describe the services provided by system in detail, the inputs and outputs, expectations and so on. The functional requirements define the facilities to be provided by the system. The non-functional

requirements are not directly concerned to the system. They may define constraints on the system and specify system performance, security, availability and emergent properties.

### **2.2.1 Client Requirements**

#### **Interviewing:**

This Inventory management system has some collected requirements from Auto parts shop. Major requirements collect from interviewing of shop owner, I have visited Rahim Parts in Bangla Motor name of shop owner Mr.kamal, and he is not using automated system for his shop. His using system just like as whole things are recorded into paper documents. Every day they recorded their transaction in paper documents but manually it's really hard to specific total revenue.

#### **Observation:**

This technique is used when attempting to document an existing process or when a project's goal is to improve a process. Observation is a great way to understand what the end user goes through in their job and can provide some instant requirements for how a process can be improved. I decide that, I will need to spend some time in the parts shop, observing and taking notes about what the shop owner does. So, I do. I take a few days and hang out in the parts shop. I discover that the shop owner talk with the customer and take an order from him. Then find out those product which the customers want. Then the shop owner Bargain with customer and fixed a price for those product. Finally, the shop owner write this entity in his register and write a receipt forthe customer. As I suspected, I'm just writing code for entry product, but not for sale product.

#### **Brain Storming:**

Brainstorming is a very effective technique for idea generation. Brainstorming is most effective when there are two phases – creating ideas and validating ideas. The key to making brainstorming work is to not criticize, remove, or prioritize any ideas during the creation phase. After I have done creating ideas, I can move into the validation phase, where bad ideas are removed and good ideas are prioritized. There I have been studies that creative thought is more effective when there's a time limit.

#### **Supervisor feedback:**

Some requirements are selected by analysis requirements and weekly meeting with project supervisor Farzana Sadia, Lecturer Department of Software Engineering.

After collecting requirements, the next phase is to analysis on requirements. In this stage, software engineers work with customers and system end-users to find out about what services the system should provide, the required performance of the system, hardware constraints and so on.

### 3.1 Requirements analysis

The requirement analysis is divided into three phases. The phases are given below:

#### 3.1.1 Requirements Elicitation

Requirement discovery is the process to gathering requirements about the proposed system. The source of the information is to learn related system, identifying the stakeholders and interviewing them. There are many techniques to gather requirement. Some common techniques are used to gather proper requirements. Techniques that are used to collection requirements are:

**Interviewing** –The requirements of clients are very important. The major requirements are collected by interviewing clients with different types of questions.

**Idea sharing** - Some requirements are elicited by analysis on collected requirements and weekly meeting with project supervisor & other teacher.

All functional and non-functional requirements are listed in requirement classification.

Basically, requirements are classified into two categories – functional and non-functional.

**Functional Requirements:** The following requirements are functional requirements of proposed system:

1. Login of Admin account
2. Admin can Update default password after successful login.
3. Admin can add new product.
4. Admin can search product by product id and product name.
5. Admin can update product.
6. Admin can delete product.
7. Admin can sort product.
8. Admin can view store product.
9. Admin can sale product.
10. Admin can view sales information.

This is a module based project. So that the overall **Agile-Scrum** methodology followed here are as following features priority:

- ❖ Admin Login
- ❖ Manage Product
- ❖ Manage Sale Product

**Non-functional Requirements:** The following requirements are non-functional requirements of the proposed system:

- ❖ Portability: As the system is chosen to develop using eclipse IDE and Java language (for front-end) and SQLite for back-end database, So SQLite is required to run the software.
- ❖ Reliability: As the software has many important information of product data, the system must be run on secure hardware. It may have backup of every day database to recover any disaster.
- ❖ Efficiency: This software is used to store products and generate product sales report. It is working with a large number of products data. The performance of server must be better to process data to authorized users in time.
- ❖ Usability: The software has many features. The Graphical User Interface must be user friendly, so that users feel easy to use.
- ❖ Dependability: The system must be dependable. It should deliver accurate report when submitted data by users. System must not be vulnerable and login process should be very secure.

## 3.2 Software Requirement Analysis

### 3.2.1 Objective/Goals

This system is generate different types of reports so that we can,

- To keep inventory at sufficiently high level to perform production and sales activities smoothly.
- To minimize investment in inventory at minimum level to maximize profitability.
- 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.
- To minimize carrying cost of inventory.
- To reduce the losses of theft, obsolescence & wastage etc.
- To make arrangement for sale of slow moving items.



### 3.2.2. Software Deliverables

The proposed system deliverables are following:

- a. Project Plan
- b. Design Document
- c. Code List

### 3.2.3. Software Requirement Specification

A software requirement is a description of software to be developed with functional and non-functional requirements. SRS describes the essential behavior of software from user's point of view.

The specified requirements are shown in following table based on module. The functional requirements are following:

Module #	Requirement ID	Requirement Title	Priority
<u>Module 1</u> Login	FR_001	Login of Admin account	High
<u>Module 2</u> Manage Product	FR_002	Add new products.	High
	FR_003	Search products information.	High
	FR_004	Update product information.	High
	FR_005	Delete product information.	High
	FR_006	Sorting product.	
	FR_007	View store product information.	High
<u>Module 3</u> Manage Sale Product	FR_007	Sale products.	High
	FR_008	View sales information.	High

Table 02: Functional Requirements

Non-functional Requirements: The non-functional requirements are shown in following table:

<b>Module #</b>	<b>Requirement ID</b>	<b>Requirement Title</b>
NFR_001	Portability	As the system is chosen to develop using eclipse juno IDE and Java language (for front-end) and SQLite for back-end database, So SQLite is required to run the software.
NFR_002	Reliability	As the software has many important information of product data, the system must be run on secure hardware. It may have backup of every day database to recover any disaster.
NFR_003	Efficiency	This software is used to store products and generate product sales report. It is working with a large number of products data. The performance of server must be better to process data to authorized users in time.
NFR_004	Usability	The software has many features. The Graphical User Interface must be user friendly, so that users feel easy to use.
NFR_005	Dependability	The system must be dependable. It should deliver accurate report when submitted data by users. System must not be vulnerable and login process should be very secure.

Table 03: Non Functional Requirements

### 3.2.4 Description of Specific Requirements

The description of each module with sub-modules is given below with necessary diagram:

#### 1. Module 1 (Login)

**FR\_001**

**Requirement Title:** Login of Admin Account

**Description:** Admin can login to this system with valid username & password.

Module 1(Login) Use Case: The use case of module one is following:

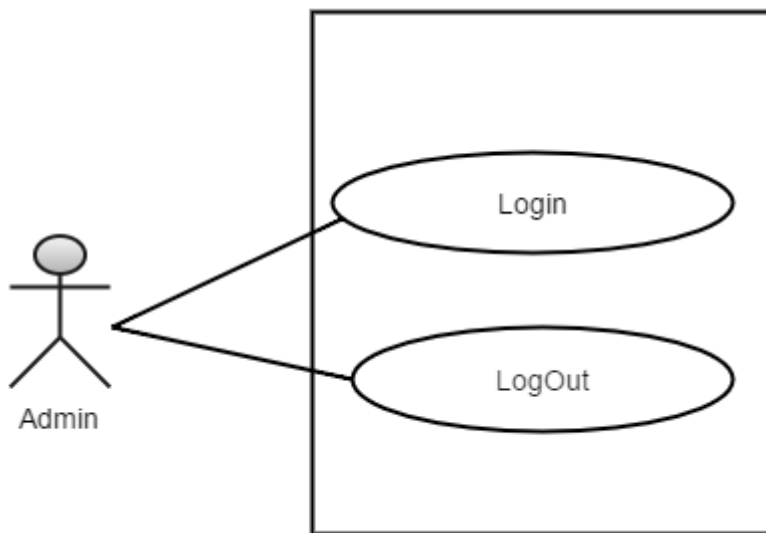


Figure 01: Admin Login

## 2. Module 2 (Manage Product)

### **FR 003**

**Requirement Title:** Add new products.

**Description:** After login into system admin add new products.

### **FR 004**

**Requirement Title:** Search products information.

**Description:** Admin can search a stored product.

### **FR 005**

**Requirement Title:** Update products information.

**Description:** The admin must be able to modify/update stored product information.

### **FR 006**

**Requirement Title:** Delete products information.

**Description:** The admin must be able to delete products from the store.

### **FR 007**

**Requirement Title:** View store products information.

**Description:** Admin can delete product information using product id.

Module 2(Manage Product) Use Case: The use case of module one is following:

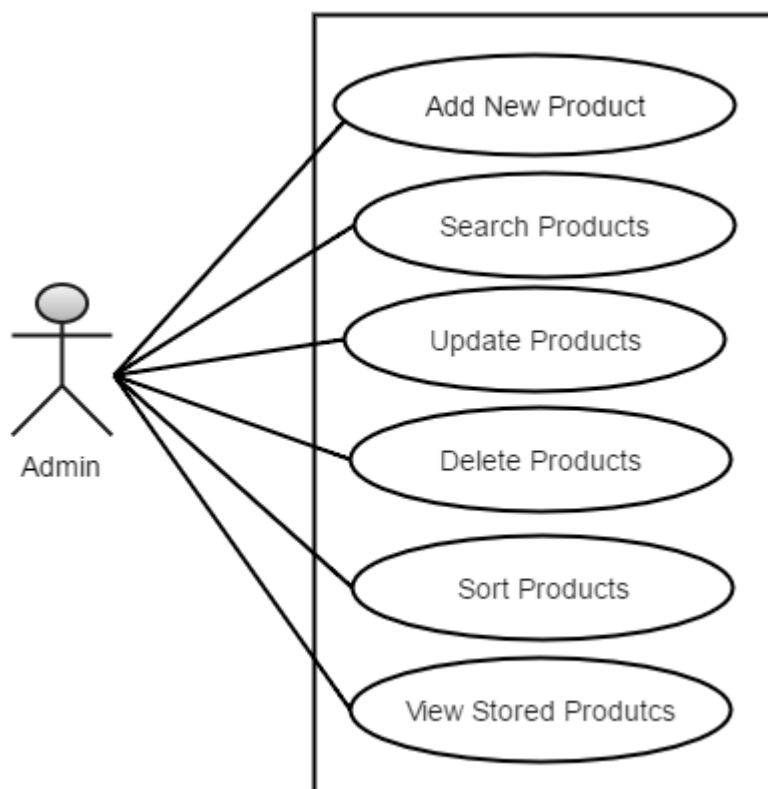


Figure 02: Manage product

### 3. Module 3 (Manage Sale Products )

**FR 008**

**Requirement Title:** Sale products.

**Description:** Admin can sale products.

**FR 009**

**Requirement Title:** View sales information.

**Description:** Admin shall be able to view the entire history of the Selling product.

Module 3(Manage sale products) Use Case: The use case of module one is following:

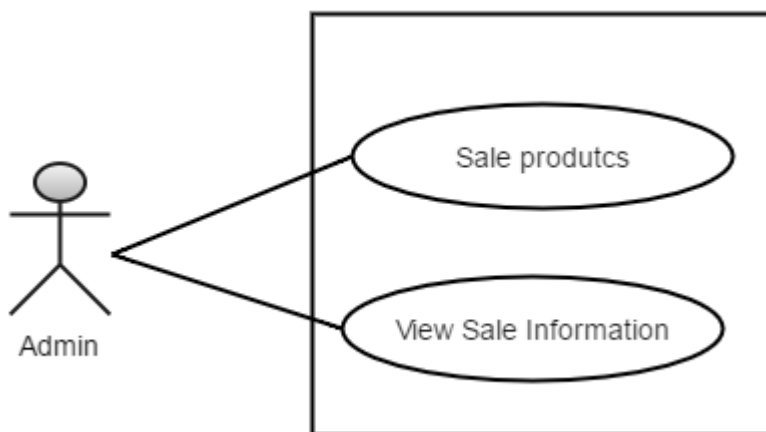


Figure 03: Manage sale product

## **3.2 Software Development Plan (SDP)**

Software Development Plan is a comprehensive that includes all information required to manage the project. The purpose of SDP is to describes the approach to development of software and generate Top-level Plan.

### **3.3.1 Overview**

#### **3.3.1.1 Proposed System and Its benefits**

This project is aimed at developing mobile based Inventory Management System for a small auto parts shop. This system can be used to store the details of the inventory, update the inventory based on the stored details, Sale products and view sales information. This system used by the administrators for performing admin level functions such as adding new items to the inventory, update items to the inventory and delete items, search item by id and name, sorting product etc. Mobilizing inventory management applications increases efficiency across the board, especially when it comes to work orders and paperwork. Mobile apps can actually eliminate most inter-department paperwork. It's easy to find out how many units of products in store and how many products are sold.

#### **3.3.1.2 Features of the system**

The features of this system are following:

1. Login
2. Add new products.
3. Search products.
4. Update products information.
5. Delete products.
6. View store products information.
7. Sort product.
8. Sale products.
9. View sale products information.

### 3.3.2 Project Organizations

This section describes the process model of proposed system and responsibilities of individuals on this system.

#### 3.3.2.1 Development Methodology

The proposed system is developed using Agile-Scrum methodology [8][9][10]. We choose scrum because this is the module based system. We specify module as per client requirement. In scrum it is possible to develop system as per features priority & if anything misses in the development we can change our system as per client requirement in sprint cycle.

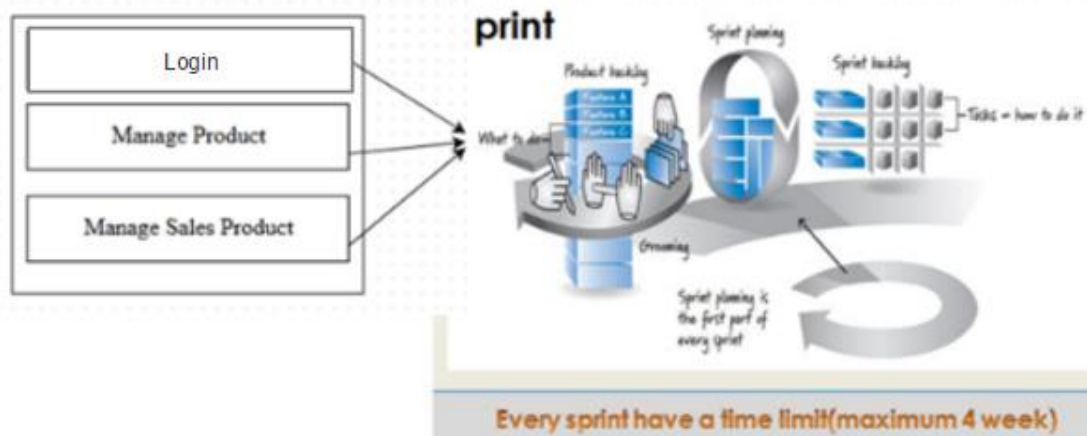
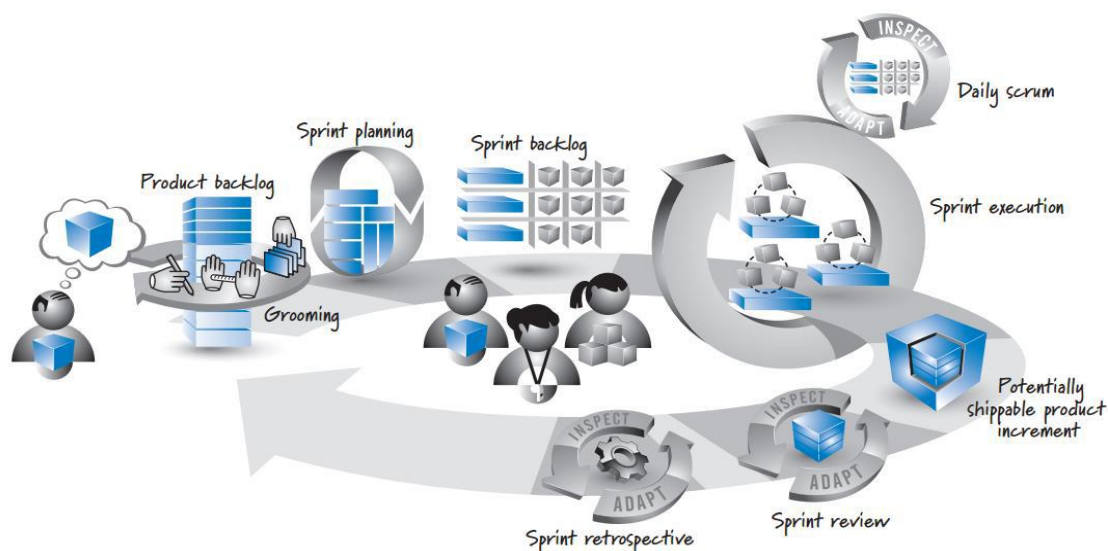


Figure 04: Agile-Scrum Methodology

### 3.3.2.2 Responsibilities

The organizational structure of team who are involved in the development of proposed system is shown in following figure with their contribution. The software is designed and developed by Md. Sabbir Hasan Tonmoy and all development process is verified by Farzana Sadia Supervisor of this project.

### 3.3.3 Managerial Process

#### 3.3.3.1 Management Objectives

The management objectives of this system are to complete each module task within schedule. The work is submitted to the project supervisor at the end of the week and makes a correction. When a module is done then starts the next module work. After complete a module if there need to change anything then again start working that module and update the rest of work.

#### 3.3.3.2 Monitoring and Controlling Mechanism

The working process of the software is divided into some module and makes a schedule to complete those modules. Project activities are monitored based on the following mechanism. The mechanisms are:

**User Interviewing:** System will be generating report so that firstly I finalize what types of data need. For that we are discussing our knowledge with our team. Then I discuss with auto parts shop owner what they need to know from this system. We also discuss my system goal with some software department senior students. The collected requirements are noted and sent to the supervisor at the end of the week. Finally, supervisor ensures a common standard to develop the system for useful report.

**Documentation:** When required data are discovered from client's requirements and some references, then documentation works started. End of week the documentation works are shown to the supervisor and the supervisor solved the problems.

**Internal Server:** The internal server to implement the proposed system is XAMPP server and IDE is used Eclipse Juno. End of the week the coding activities is shown to the supervisor and supervisor detects the errors which don't meet the software specifications.



### 3.3.3.3 Risk Management

Management and minimizing risk is the main responsibilities of proposed system. At each stage of the project there were a meeting at the end of the week, evaluated by the supervisor and corrected risks.

Unclear Understanding the requirements: manage this risk by holding weekly meetings with project supervisor.

Improper Management of Projects: Corrected ambiguous requirements and make a perfect list of requirements before development by regular weekly meetings with supervisor.

Loss of necessary project related documents: The documents uploaded to mail each meetings to get final overview of project documents.

Risks and their impact to the system is shown in following table:

Risk ID	Risk Description	Probability of Risk (0.1-1.0)	Impact (1-10)	Risk Exposure (P*C)	Priority
01	Unclear requirements. Some of the requirements is not fully clear to define the process of validation	0.3	7	1.4	Medium
02	Restrict the unauthorized users from login	0.6	4	2.4	High
03	Loss occurs in many forms, including physical loss of the product and errors during receipt of a product.	0.6	4	2.4	High

Table 04: Risk management

### 3.3.3.4 Tools and Technology

The tools and technology used to implement this software is described below:

Tools and IDE: Eclipse JUNO.

Language: JAVA

### 3.3.4 Work Schedule

The proposed system is implemented within 8 months. System development works are performed in each week. After weekly meeting works are done step by step. To make quality software within defined scope the works are done and repeated correction. The total project was completed in following phases:

**Scheduling & Planning:** Software development plan with target time is given below:

Phase	Start Date	Planned Submission Date	Working Days
Project Plan	10 June, 2015	18 June, 2015	9
Project Idea Sharing	22June,2015	28 June, 2015	7
Requirements collection	02 July, 2015	29 July, 2015	28
Proposal	02 August, 2015	10 August, 2015	08
Requirements Analysis	12 August, 2015	29August, 2015	18
Software Requirements Specification	02September, 2015	17September, 2015	16
Design	20September, 2015	1October, 2015	11
Develop and Prototype Implementation	3October, 2015	2 December, 2016	60
Testing & Result	3December. 2015	22 December. 2015	20
Total Working Days: 177 days			

Table 05: Scheduling & Planning

### 3.3.5 Grant Chart

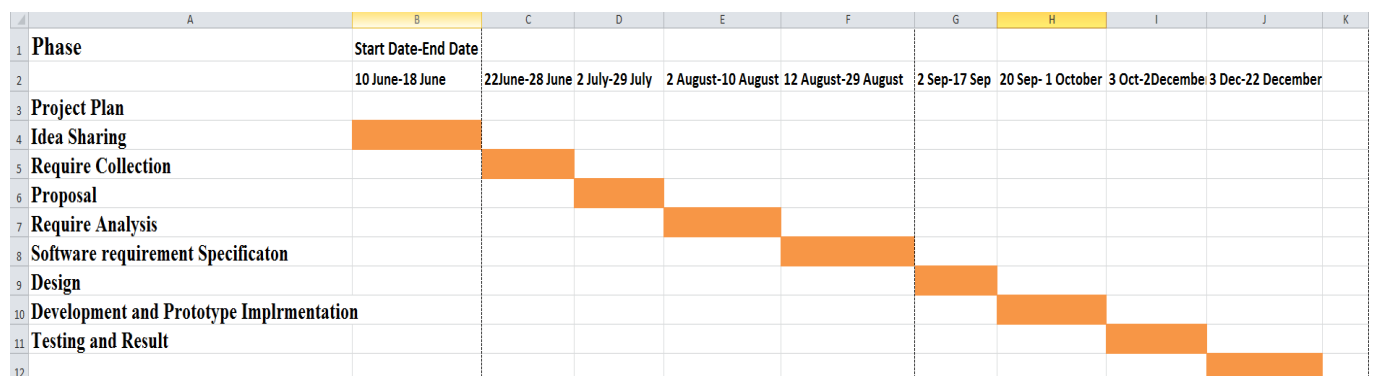


Figure 05: Grant chart

## **3.4 Software Test Plan**

This section describes about software test plan and follow IEEE standard template [5]. Test plan is the project plan for the testing work to be done.

### **3.4.1 Introduction**

According to ISTQB definition, Test Plan is a document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice and any risks requiring contingency planning. It is a record of the test planning process.

### **3.4.2 Test Items**

The test items are features to be tested that are delivered to the clients. This is the list of what to be tested. Test items for proposed system are as follows:

1. Login Process
2. Add new product
3. Search product
4. Update product
5. Delete product
6. View store product information.
7. Sale product
8. View sale status

### 3.4.3 Features to be tested

The features to be tested are as follows:

Feature ID	Title	Description	Involved User
001	Login Process	Login admin with valid credentials.	Admin
002	Add new product	Add new product to database	Admin
003	Search product	Admin search stored from database using product id.	Admin
004	Update product	Admin can update the information which was already stored into database.	Admin
005	Delete product	Admin can delete the information which was already stored into database.	Admin
006	View store product	Admin can view stored product.	Admin
007	Sale product	Admin can sale product.	Admin
008	View sale status	Admin can view all sales status	Admin

Table 06: Features to be tested

## CHAPTER 4

# SYSTEM DESIGNING

Design is the process that allows a software engineer to model the system that is to build. This model is accessible for quality and improves before code is generated, tests are conducted and end users are involved.

The High Level (HLD) and Detail Level Designs (DLD) of proposed system are given below:

### 4.1 High Level Design (HLD)

High Level Software Design is also called Software Architecture the first design step after analyzing all requirements for software. The goal is to define overall software structure which fulfill the requirements (mainly functional requirements) [6]. The diagrams that represent the High Level Design of Proposed system are given below:

#### 4.1.1 Component Diagram

The component diagram of proposed system is given below:

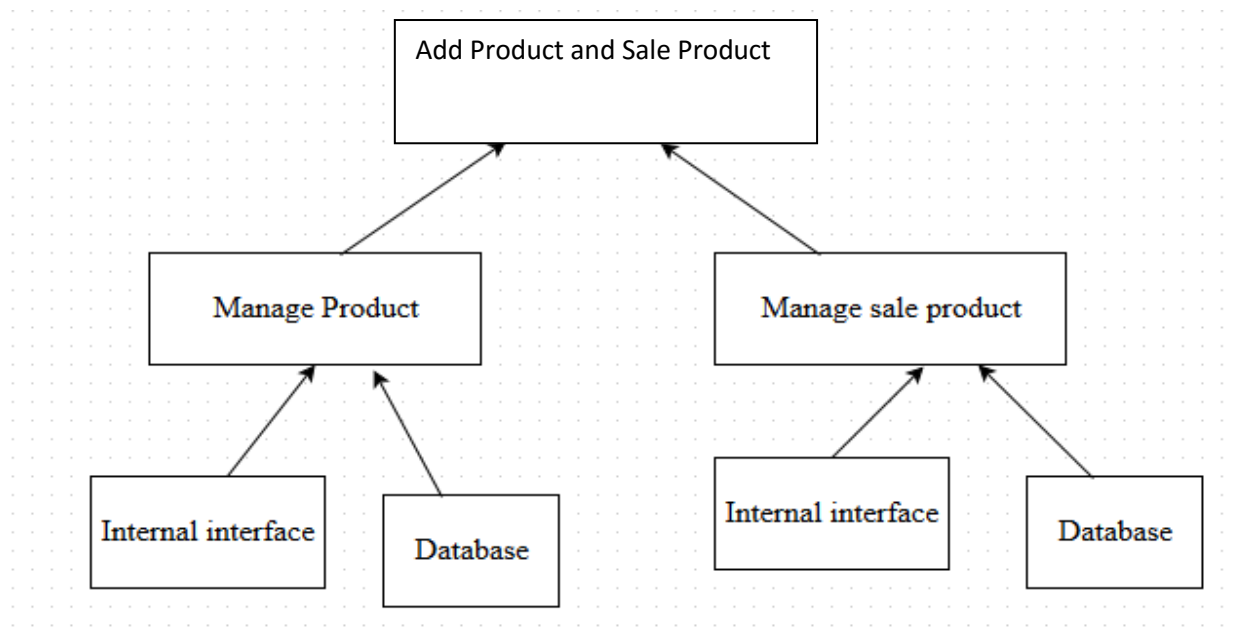


Figure 06: System component diagram

### 4.1.2 Architecture Diagram of the System

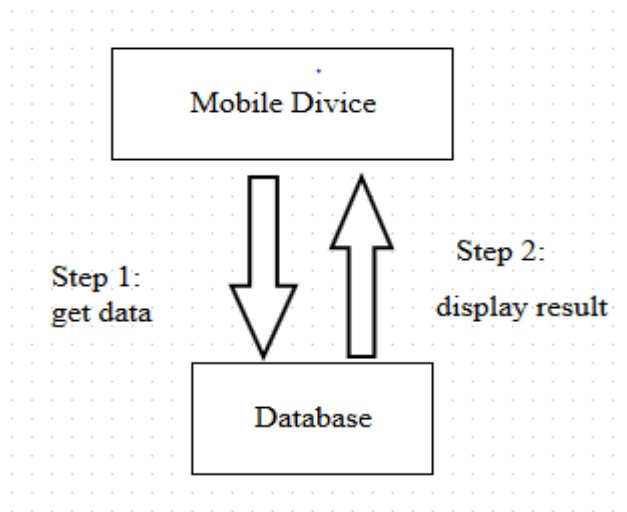


Figure 07: System Architecture Diagram

### 4.1.3 Activity Diagram

The activity diagram of proposed system describes the processing flow within each operation. The activity diagrams for each operation are as follows:

The activity diagram of each use case is given below:

#### Module 1: The activity diagram of module one are following

Admin Login: Activity diagram for admin login is following:

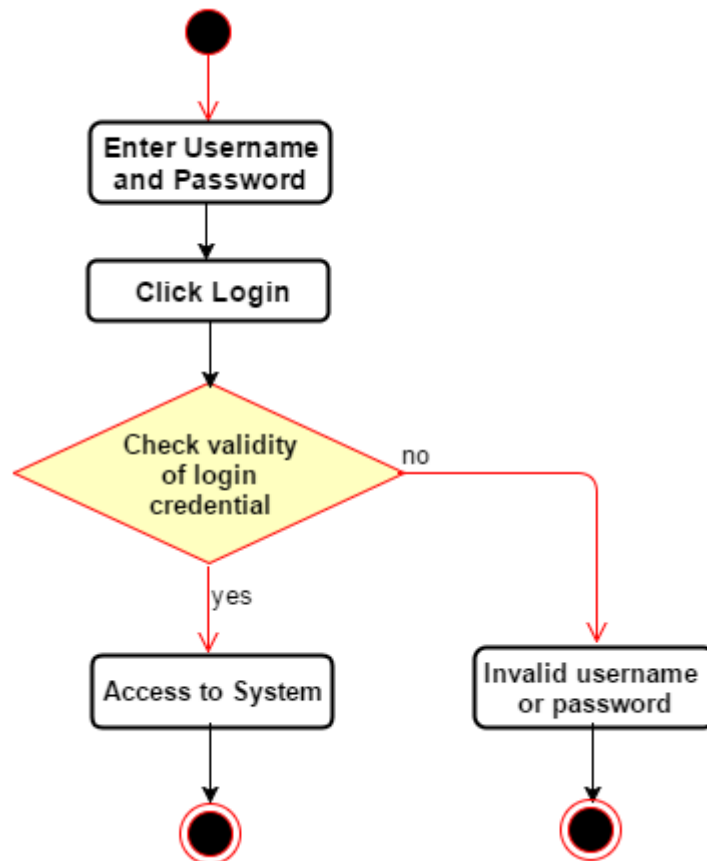


Figure 08: Activity Diagram – Admin Login

**Module 2.1: The activity diagram of module two are following**

Manage Product: Activity diagram for add product is following:

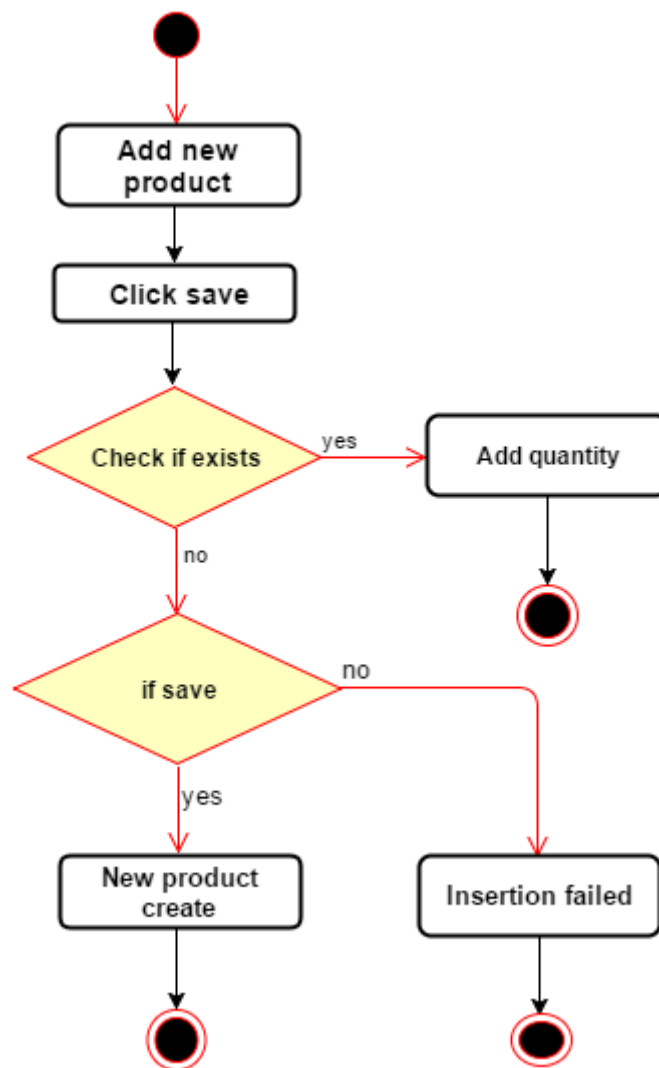


Figure 09: Activity Diagram – Add new product



## Module 2.2: The activity diagram of module three are following

Manage Product: Activity diagram for search, update and delete product is following:

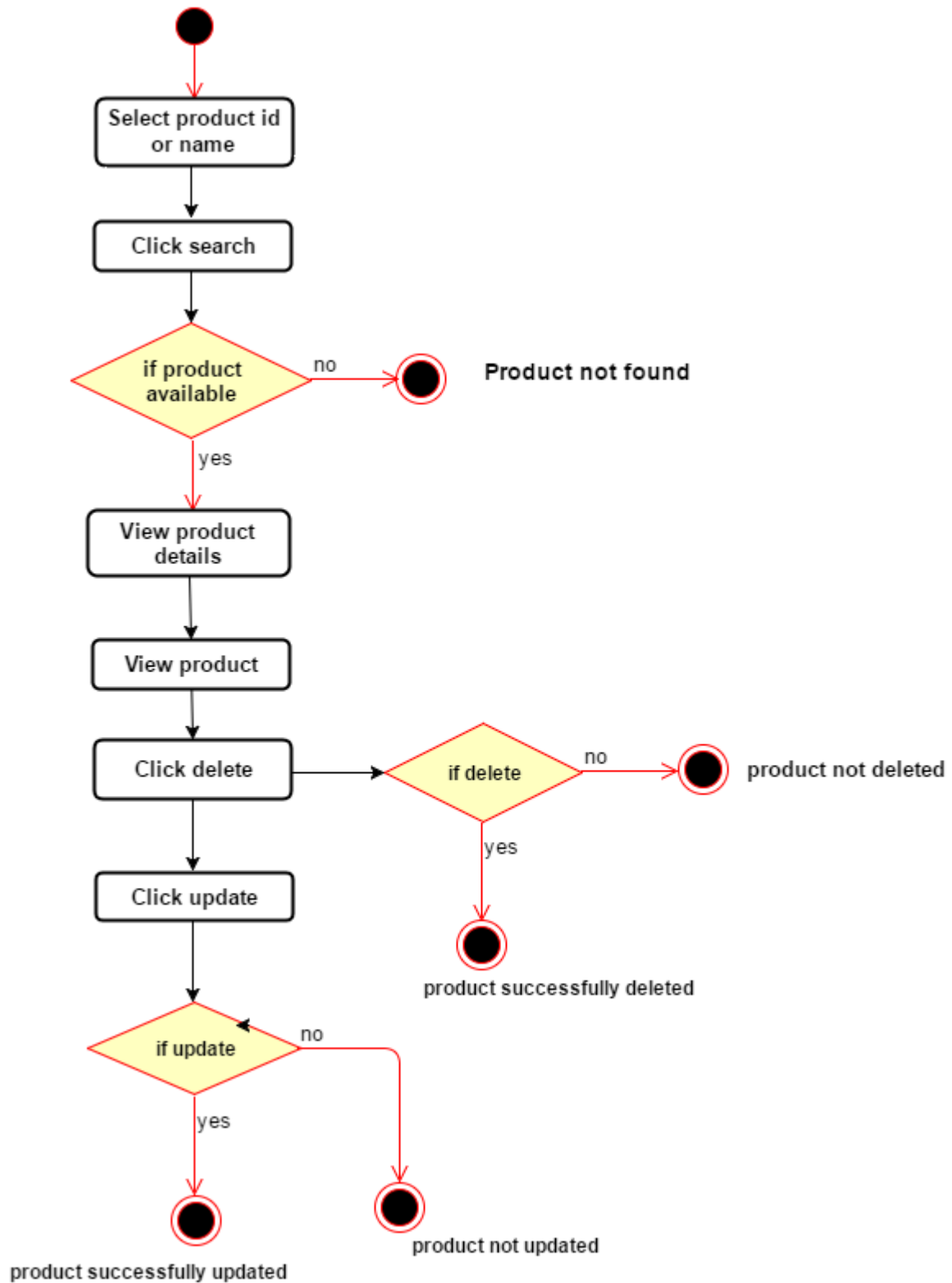


Figure 10: Activity Diagram – Search, update and delete product

**Module 3: The activity diagram of module three are following**

Manage sale product: Activity diagram for manage sale product is following:

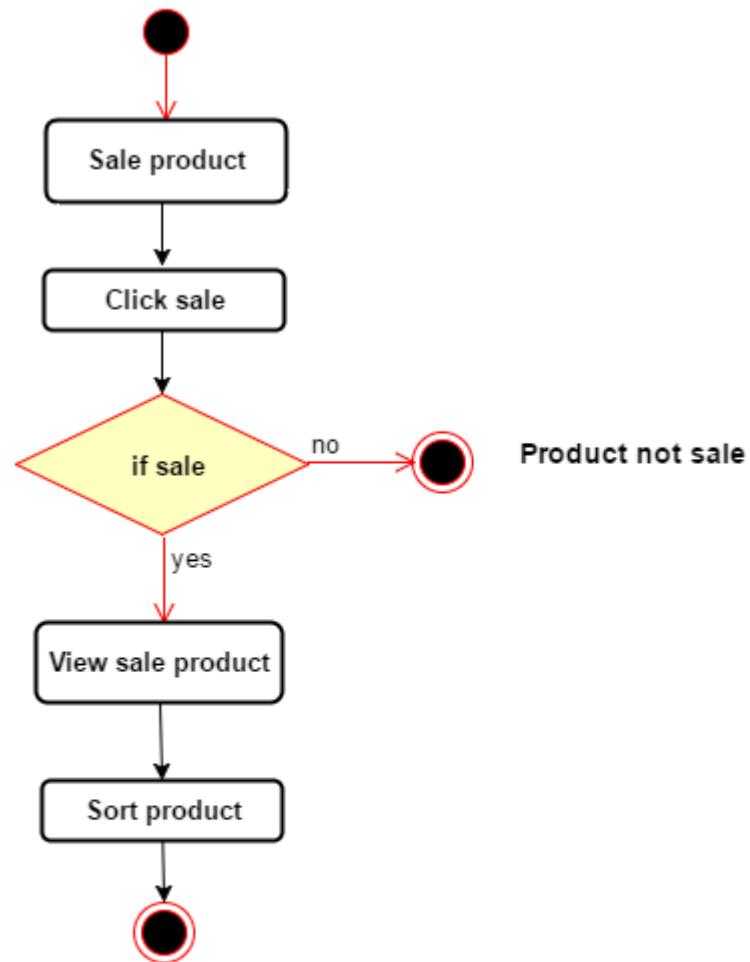


Figure 11: Activity Diagram – Manage sale product

### 3.1.4 Context Diagram (DFD Level-0)

The system context diagram of proposed system is following:

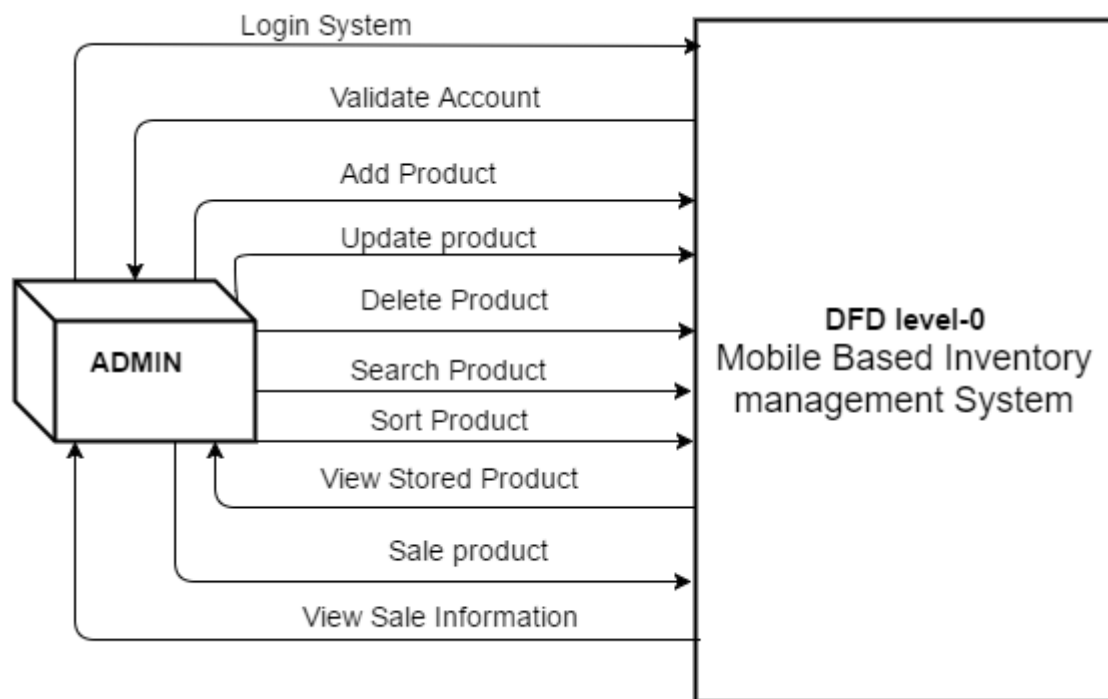


Figure 12: System Context Diagram

## 4.2 Detail Level Design

Detailed Level Design is the process of refining and expanding the preliminary design of a system or component to the extent that the design is sufficiently complete to begin implementation.

The diagrams for detailed level design are following:

### 4.2.1 Data Flow Diagram Level – 1

The Data Flow Diagram Level 1 for proposed system are given below for each module.

#### Module 1 DFD 1.1

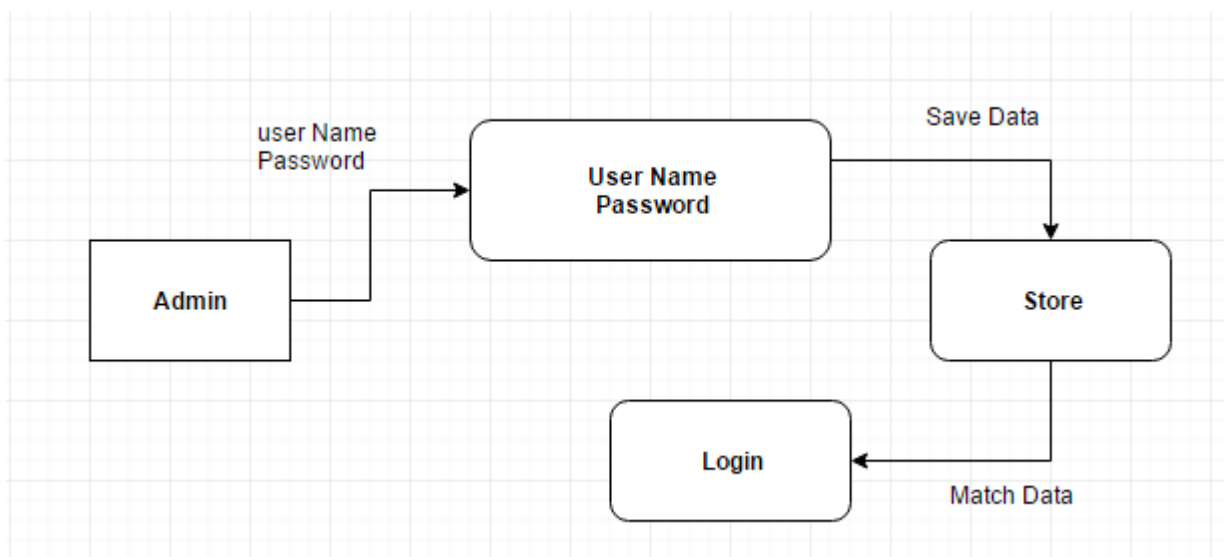


Figure 13: DFD 1.1

## Module 2 DFD 2.1

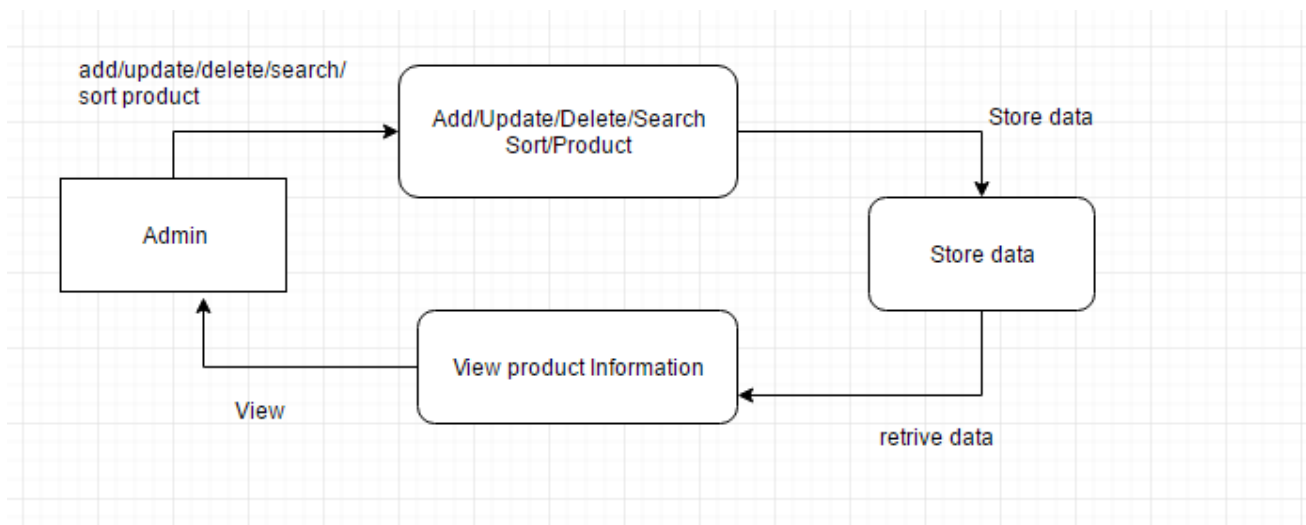


Figure 14: DFD 2.1

## Module 3 DFD 3.1

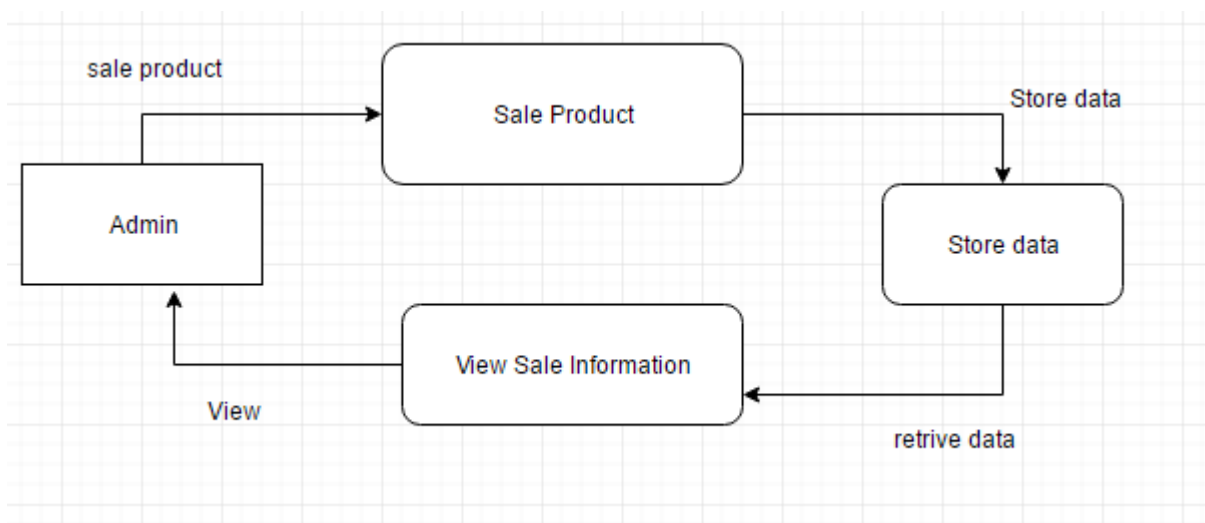


Figure 15: DFD 3.1

## 4.2.2 Entity Relationship Diagram

Entity Diagram also known as ER Diagram is a graphical representation of Entities of a system and relationship between them. There are three kind of relationship between entities are exist – 1. One to One relationship; 2. Many to Many relationship; 3. One to Many relationship.

The entity relationship diagram of proposed system (major entities and attributes) is following:

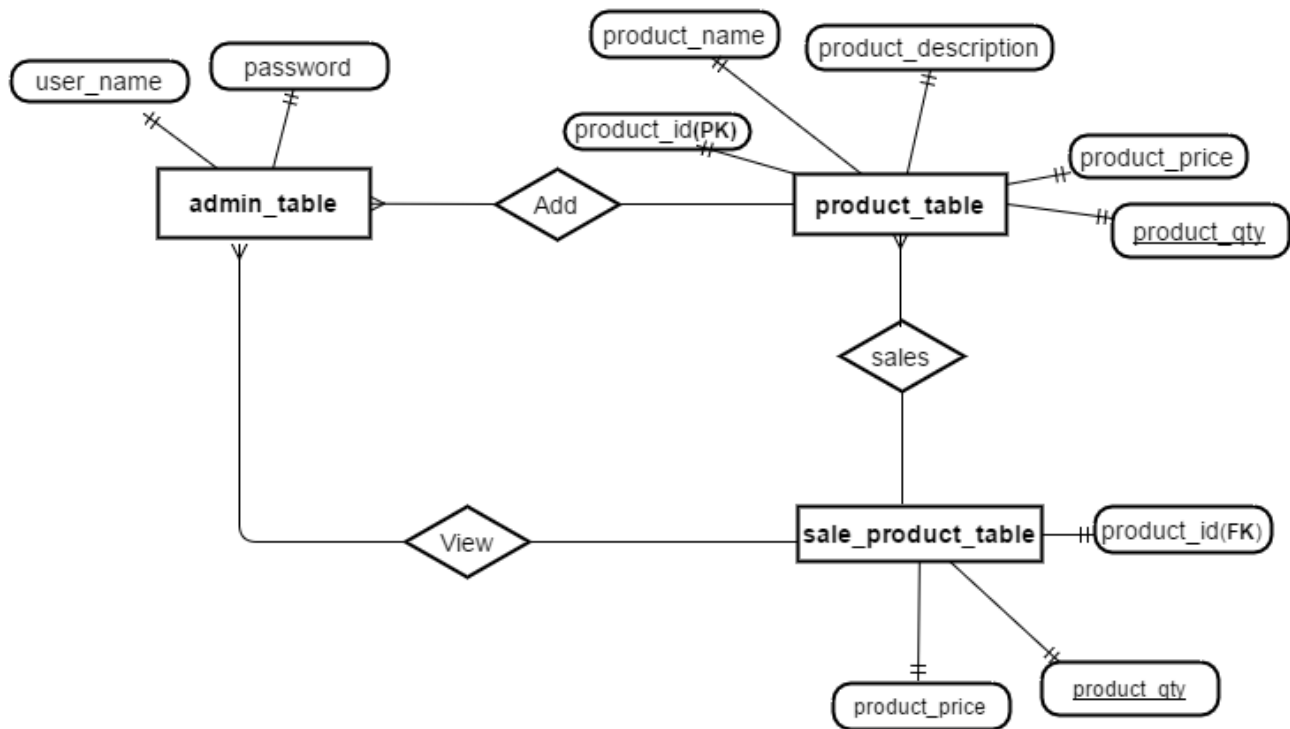


Figure 16: Entity Relationship Diagram

The Entity Relationship Diagram form the physical database is follows:

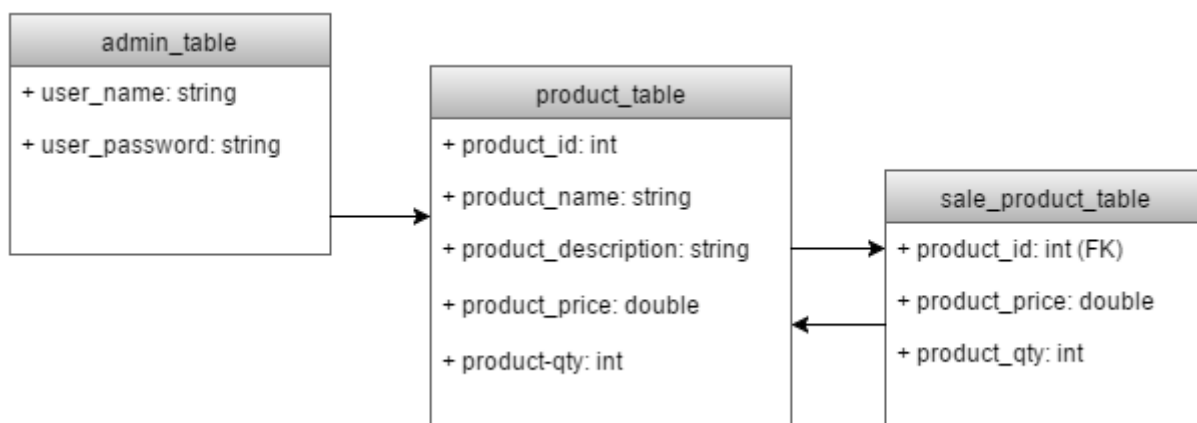


Figure 17: Physical Database

### 4.2.3 Sequence Diagram

Sequence diagram of proposed system is following:

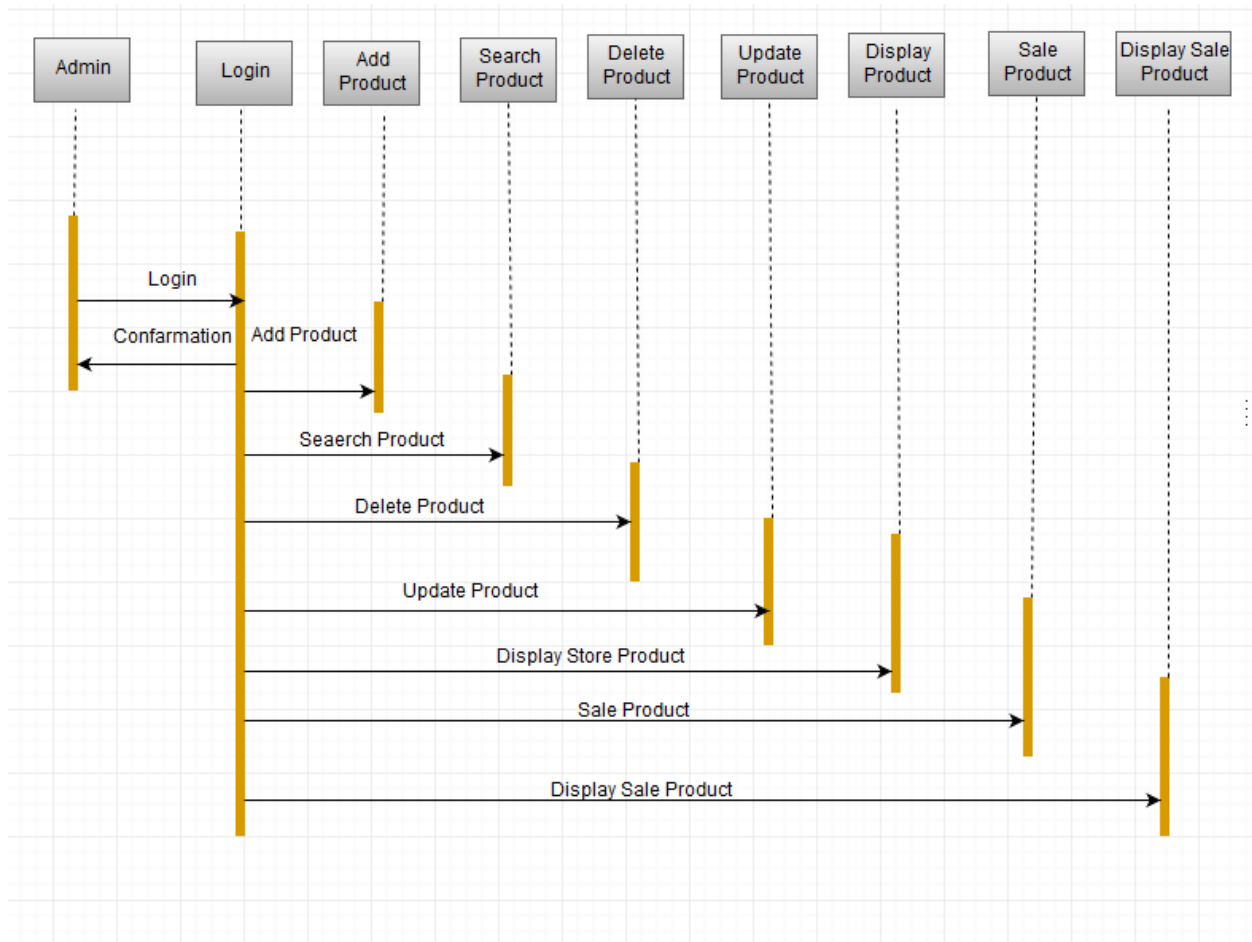


Figure 17: Sequence Diagram

# CHAPTER 5

## SYSTEM TEST AND COMPLETION

---

### 6.1 Introduction

This is aimed at identifying and correcting errors. The major objective of this activity is to ensure that the processing done by the application is correct and meets the objectives of the organization. Test plan aids in effective and systematic testing of the system and it aim at checking the errors of omission and communication that hinder the realization of the objectives.it takes the bottom up testing approach.

### 6.2 Importance of testing

1. Testing is used to find out program error on the system
2. It is used to find undercover errors in a program through the use of defect testing
3. Testing in also used to uncover new types of errors associated with new inventions and technology
4. Testing aims at assuring quality by enforcing consistency and reliability.
5. It is used for both validation and verification to develop a product that meets user requirement.
6. It is used to identify the best component combination for effective error identification



## 6.3Unit Test Case

A unit is the smallest testable part of an application like functions, classes, procedures, interfaces. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. Unit tests are created and executed by software developer during the development process.

Some major unit tests on each module is shown below:

Module 1: Login

Serial No	Inputs(username ,password)	Click Button	Expected result	Pass/fail	Code module
01.	Null, null	login	Message display Invalid user name or password	pass	Main Activity  Activity_main layout, Saveinfo Class, saveData(parameter1),
02.	Null, 12345	login	Message display Invalid user name or password	pass	
03.	12345,null	login	Message display Invalid user name or password	pass	
04.	admin, admin 12345	login	Successfully login , redirect to Home	pass	
06.	98453, 12345	login	Message display Invalid user name or password	pass	
08.	admin, null	login	Message display Invalid user name or password	pass	
09.	Null , admin	login	Message display Invalid user name or password	pass	
10.	@#\$\$%&*, null	login	Message display Invalid user name or password	pass	
11.	@#\$\$%&*, 12345	login	Message display Invalid user name or password	pass	
12.	@#\$\$%&*, @#\$\$%&*	login	Message display Invalid user name or password	pass	
13.	@#\$\$%&*, @#\$\$%&*	login	Message display Invalid user name or password	Pass	

Table 07: Login

## Module 2: Add new product

Inputs:

- (a)Product Id: integer
- (b) Product Name: String
- (c) Product quantity: integer
- (d) Product quantity: double

Serial No	Inputs	Click Button	Expected Result	Pass/fail	Code module
01.	Null (all)	Submit	Message display error	pass	Activity_home_design layout, Saveinfo Class, saveData(parameter1), BackgroundTask class, Product Class.
02.	(a)Product Id : 1 (b) Product Name: Wheel (c) Product Quantity : 10 (d) Product Price: 1000	Submit	Success message	pass	
03.  (bold mark are invalid input )	(a) <b>Product Id : !23</b> (b) <b>Product Name: \$\$pats</b> (c) <b>Product Quantity : 10.1263</b> (d) Product Price: 182.394	Submit	Unfortunat ely emulator has been stopped	fail	

Table 08: Add new product

### Module 3: **Search Product**

(a)Product Id: integer

(b) Product Name: String

Serial No	Inputs	Click icon	Expected Result	Pass/fail	Code module
01.	Productsearch by id Id=null	search	Message Display- “Product not found”	pass	
02.	Product search by name Id=null	search	Message Display- “Product not found”	pass	
03.	Productsearch by id Id=1	search	Success Message- “Product found”	pass	Activity_search layout, SearchActivityClass , searchProduct(parameter1),
04.	Productsearch by id Name=Wheel	search	Success Message- “Product found”	pass	

Table 09: Search product

#### Module 4: **Delete Product**

(a)Product Id: integer

Serial No	Inputs	Click icon	Expected Result	Pass/fail	Code module
01.	Product delete by id Id=null	delete	Message Display- “Product not found”	pass	Activity_delete layout, SearchActivity Class , searchProduct(parameter1),
02.	Product delete by id Id=1  Id always integer value  Id =1,2,3 ,4  id always unique for individual customer	delete	Success Message- “Product deleted”	pass	
03.  (bold mark are invalid input )	Product delete by id  Id = #\$\$%^&()+_?! <b>or five</b>  Special character not allow for id	delete	Message display – “Product not found”	pass	

Table 10: Delete product

## Module 5: Update Product

(a)Product Id: integer

Serial No	Inputs	Click icon	Expected Result	Pass/fail	Code module
01.	Productupdate by id Id=null	update	Message Display- “Product not found”	pass	Activity_update layout, UpdateActivity Class , searchProduct(parameter1), updateProduct(parameter1)
02.	Productupdate by id Id=1  Id always integer value  Id =1,2,3 ,4  id always unique for individual customer	update	Success Message- “Product update”	pass	

Table 11: Update product

## Module 6: Display Product

Serial No	Inputs	Click icon	Expected Result	Pass/fail	Code module
03.		Display product	Success Message	pass	Activity_display_product layout, display_product_row layout, ProductAdapterClass,DisplayProduct class, add(parameter1), getCount(), getView(parameter1, parameter1, parameter1)

Table 12: Display product

### Module 7: **Sale product**

Inputs:

- (a)Product Id: integer
- (b) Product Name: String
- (c) Product quantity: integer
- (d) Product price: double

Serial No	Inputs	Click Button	Expected Result	Pass/fail	Code module
01.	Null (all)	Submit	Message display error	pass	Activity_home_design layout, Saveinfo Class, saveData(parameter1), BackgroundTask class, Product Class.
02.	(a)Product Id : 1 (b) Product Name: Wheel (c) Product Quantity : 13 (d) Product Price: 1600	Submit	Success message	pass	
03.  (bold mark are invalid input )	(a) <b>Product Id : !23</b> (b) <b>Product Name: \$\$pats</b> (c) <b>Product Quantity : 10.1263</b> (d) Product Price: 182.394	Submit	Unfortunat ely emulator has been stopped	fail	

Table 13: Sale product

## Module 8: Display Sale Product

Serial No	Inputs	Click icon	Expected Result	Pass/fail	Code module
03.		Display sale product	Success Message	pass	Activity_display_product_sale layout, display_product_sale_row_layout, ProductAdapterClass, DisplayProductSale class, add(parameter1), getCount(), getView(parameter1, parameter1, parameter1)

Table 14: Display sale product

## 6.4 Unit Test Report

Total unit test case sample of 8 modules. Some test cases are succeeding in first iteration and some are succeed in second iteration. The succession percent are shown in following table:

Number of Unit	First iteration 100% success	Less than 100%	Total Succession
Total : 8	9	2	80%
Total : 8	11	0	100%

Table 15: Unit test report

## 6.5 System Test Report

Scope of system test: The Test Plan defines the unit, integration, system and Client Acceptance testing approach. The test scope includes the following:

- Testing of all functional, application performance, security and use cases requirements listed in system test plan.

System Test Objective and Goal: Software testing is the process that ensures the quality of software by identifying defects and failures. The major purpose of system testing is to reduce defect and deliver an error free good quality software.

Software defects are solved in each stage of development. The correction percentage of software defects are 90% that shown in a table.

Severity	Description	Correction
Critical	The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data [10]. These types of defects make the system unusable. Example: Installation failed, failed to connect with system.	95%
Moderate	The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate [10]. This type of system may cause to make some features unusable. Example: Cannot add a product, cannot update or delete product.	85%
Minor	The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor [10]. These types of error make the system looking bad. Example: UI design problem	78%
Total correction of defects:		86%

Table 16: System test report

## 6.2 Software Development Completion Report (SDC)

### 6.2.1. Basic Overview

Project Information			
<b>Project Title</b>	<b>Current Market Driven Carrier Build-Up System For Departmental Students: A Case Study of SWE Students</b>		
<b>Start Date</b>	10 <sup>th</sup> June, 2015	<b>End Date</b>	22 <sup>th</sup> December, 2015
<b>Lead Institution</b>	Daffodil International University		
<b>Project Supervisor</b>	Farzana Sadia		
<b>Technology and Tools Used</b>	IDE: Eclipse JUNO, Language: Java		

Table 17: Software Development Completion Report



### 6.2.2 Project Outcomes

It is difficult to describe the outcomes and impacts without going into detail.

The document produced for this project has provided a better understanding and knowledge of Current Market Driven Carrier Build-Up System. The project provides a process to work in a project with meeting between Agile Scrum methodologies.

- The system proposed for Agile Scrum methodology to work.
- It maintains a version controlling of software document on each phase.
- The proposed system is developed to make a proper document in AGILE SCRUM Methodology.

### 6.2.3 Project Achievements

Only one user for proposed system. This project is aimed at developing Android Smart Phone based Inventory Management System for a small auto parts shop. This system can be used to store the details of the inventory, update the inventory based on the sale details, produce receipts for sales, generate sales and inventory reports periodically etc. This system used by the administrators for performing admin level functions such as adding new items to the inventory, update items to the inventory and delete items etc.

What Admin User can do?

- Login to the system.
- Add New Product.
- View all stored product.
- Search for product by product id and product name..
- Delete product.
- Sort product.
- Sale product.
- View all sale information.

### **Lessons Learned:**

I feel that the project is very much completed the need originally envisaged. All kind of outputs such as reporting, document management are completed.

Some experience, when working. The learning lessons from this project noted down:

- When planned to build software, we need to learn about Agile Scrum Methodology and also learned how to prepare a software document.
- I had the opportunity to interviewing clients in real life. I had opportunity to discuss with the department teacher to get proper solution for well software document.
- As implement the software, I had to learn development technology. I learned Java language, Eclipse IDE and SQLite Database.

## **6.2.4 Project maintenance**

Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes [11].

The project is developed with a document on each module of Agile Scrum Method that refers all required documents. So that, it would be easily modifiable to any developer. Eclipse IDE is develop the software and Java Language is to implement the software. The developer who modify the software, must have knowledge on Eclipse and Java technology.

## **6.5 Conclusion:**

### **6.5.1 Remarks of the work:**

It's wonderful and learning experience for me while working on this project. This project took me through the various phases of project development and gave me real insight into the world of software engineering. The joy of working and the thrill involved while tackling the various problem and challenges gave me a feel of developers industry.

### **6.5.2 Limitation:**

It was due to this project I came to know how professional software are designed. In development process there are some limitations. I will try to overcome my limitation in future.

### **6.5.3 Future work:**

Currently, the software is used for only some specific type of questions. List of my feature work is given below:

- i. Report on individual item sales between date ranges, showing who ordered and how many.
- ii. Once a report is generated it can be printed.
- iii. Automatic email and text alerts for low inventory.
- iv. Database Backups can be manually made or automated for preventing data losses.
- v. Full barcode support is included on related features, with standard barcode scanners.




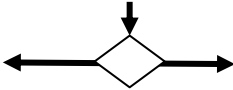

## CHAPTER 6

## REFERENCES

- [1] Laurie Williams and Sarah Heckman (2008, Aug. 14). “Requirements” <http://openseminar.org>. [Online]. Available: <http://openseminar.org/se/modules/8/index/screen.do> [Accessed: Oct .30, 2014]
- [2] Ian Sommerville, “Functional and non-functional requirements” in Software Requirements, Eight Edition, pp. 143-150
- [3] Ian Sommerville, “Requirement elicitation and analysis” in Requirement engineering processes, Eight Edition, pp. 170-174
- [4] Stack Overflow. [Online]. Available: <http://www.stackoverflow.com/>.
- [5] IEEE STD 1058-1998 IEEE Standard for Software TEST PLAN OUTLINE.
- [6] “Software Architecture”, [www.the-software-experts.com](http://www.the-software-experts.com). [Online]. Available: <https://www.york.ac.uk/integrity/ieee.html> [Accessed: Jan. 24, 2015].
- [7] Agile Methodologies. Available: <https://www.versionone.com/agile-101/agile-methodologies/>
- [8] Wikipedia. “Agile Software Development”, [wikipedia.org](http://wikipedia.org). Available: [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)
- [9] PDF. “Agile Scrum”, PDF [Offline].
- [10] “What is the difference between Severity and Priority? ”, <http://http://istqbexamcertification.com/>. [Online]. Available: <http://istqbexamcertification.com/what-is-the-difference-between-severity-and-priority/> [Accessed: Apr. 17, 2017].
- [11] Wikipedia. “Software maintenance”, [wikipedia.org](http://wikipedia.org). [Online]. Available: [http://en.wikipedia.org/wiki/Software\\_peer\\_review](http://en.wikipedia.org/wiki/Software_peer_review) [Last Modified: 9 February 2015, 04:57].



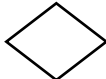

#### Appendix A

##### List of Notations Used To Describe Activity Diagrams

Initial State	
Final State	
Action State	
Decision	
Control flow	


#### Appendix B

##### List of Notations Used to Describe E-R Diagrams

Entity sets	
Use case	
Relationship sets	
Connector	

##### List of Notations Used to Describe Class Diagrams and Use Case Diagram

## Major Components:

Class and Property	
Use case	
Inheritance	
Connector	

**Appendix C:** Major searching keywords in the time of project development are following:

Common document format for all software, Object Oriented base software document, Agile Scrum Methodology, Sprint in Agile Scrum Methodology, JAVA keywords.

**Appendix D:** The tools that used to design logical diagrams are:

Online tool:

1. Draw.io ([www.draw.io](http://www.draw.io))

Software documentation tools:

1. Microsoft Office Word 2013