CSE-180001,

Analysis and Development of Online Shopping System.

A Report Submitted by:

Name	ID
Motiur Rahman	CSE-1403003021
Mohammed Ismail Hossain	CSE-1402002003
Md.Satu Prodhan	CSE-1403003005
Shakhowat Hossain Anik	CSE-1403003027
Md.Rayhan Islam	CSE-1403003006

In Partial Fulfillment of the Requirements for the Award of

Bachelor of Computer Science and Engineering



Department of Computer Science and Engineering

Sonargaon University [SU]

Spring-2018

Supervisor Declaration

This is to certify that Practicum report on "**Analysis and Development of Online Shopping Management System.**" has been carried out by:

Name	ID
Motiur Rahman	CSE-1403003021
Mohammed Ismail Hossain	CSE-1402002003
Md. Satu Prodhan	CSE-1403003005
Shakhowat Hossain Anik	CSE-1403003027
Md.Rayhan Islam	CSE-1403003006

The report has been prepared under my guidance and is a record of the bona fide work carried out successfully. To the best of my knowledge and as per their declaration, no parts of this report has been submitted anywhere for any degree, diploma or certificate.

Now they are permitted to submit the report. I wish their all success in their future endeavors.

Supervisor

.....Rafiqul.....

Md Rafiqul Islam (Rafiq)

Lecturer, Department of Computer Science and Engineering

Sonargaon University

Table of Contents

Supervisor Declaration ..... **Vi**

Chapter 1 ..... **1**

Introduction **1**

1.1 Chapter Overview 2

1.2 Project Overview 2

1.3 Objective 2

1.3.1 Main Objective` 2

1.3.2 Broad Objective 2

1.4 Scope of the project 2

1.5 Limitation of the project 3

1.6 Methodology 3

1.7 Selection of project development process model 3

1.8 Feasibility Study: 3

1.8.1 Technical feasibility 4

1.8.2 Economic Feasibility 4

1.8.3 Operational feasibility 4

Chapter 2 ..... **5**

System **5**

Requirement **5**

2.1 Methodology 6

2.2 Requirements Engineering 6

2.2.1 User Requirements 7

2.2.2 System Requirement 7

2.2.3 Functional Requirement 9

2.2.4 Non-Functional Requirement 9

2.3 USE CASE Diagram for Online Shopping Management System: 10

2.4 Use Case Text 11
Chapter 3 14
System Planning and Schedule 14
3.1 Functions of Proposed System 15
3.2 Function Description 15
3.3 System Project Planning 15
3.3.1 System Project Estimation 16
3.3.2 Function Oriented Metrics 16
3.3.3 Process Based Estimation 16
3.3.4 Task Scheduling 18
3.3.5 Project Schedule Chart 19
Chapter 4 20
Analysis 20
and Design 20
4.1 Structural Diagrams 21
4.2 Activity Diagrams for this system are given bellow: 21
4.2.1 Activity Diagram for login (Admin) 21
4.2.2 Activity Diagram for login (Customer) 22
4.2.3 Activity Diagram for Shopping 23
4.2.4 Activity Diagram for Manage Order (Admin) 24
4.2.5 Activity Diagram for Manage Product (Admin) 25
4.2.6 Activity Diagram for Review and Comment (Customer) 25
4.2.7 Activity Diagram for Return Product (Customer) 26
4.2.8 Activity Diagram for Invoice 26
4.3 Swim lane Diagram for this System 27
4.5 Sequence Diagram for the System 28
4.6 ERD: 29
4.7 Data Flow Diagram (DFD) 30

4.7.1 DFD: Context label 30
4.7.2 DFD: Level 1 30
4.7.3 DFD Level 2 30
4.7.3.1 Process 1 31
4.7.3.2 Process 2 31
4.7.3.3 Process 3 32
4.7.3.4 Process 4 32
4.7.3.5 Process 5 33
4.7.3.6 Process 6 33
4.7.3.7 Process 7 34
4.7.3.8 Process 8 34
4.7.3.9 Process 9 35
4.7.3.10 Process 10 35
4.7.3.11 Process 11 36
Chapter 5 37
System Quality 37
and Testing 37
5.1 System Quality Management 38
5.1.1 Software Quality Management Process 39
5.2 System Testing 41
Chapter-06 Termination 44
6.1 Preface 45
6.2 Practicum and Its Value 45
6.3 Conclusion 46
6.4 Bibliography 47

Chapter 1

Introduction

1.1 Chapter Overview

This project named as **Analysis and Development of Online Shopping Management System** has been assigned to us. This report is generated to describe the processes and works done in different levels of management system development. In this report we have described every part of the development segments with proper illustrations.

1.2 Project Overview

Online Shopping Management System can be an important system for any organization. As it is an automated system, it can reduce the time and complexity of administrator work. With this system administrator can easily handle his business and can fulfill the demand of Customer, which will be very helpful for both administrator and Customer. The Project named as "Analysis and Development of Online Shopping System. We have done this project using "Rapid application Development Model". We have completed this project under assign project teacher. We were assigned to analysis the system, designing and developing different segments. In this report we have describe how we developed this system and how it will work.

1.3 Objective

1.3.1 Main Objective

The main objective of this Online Shopping Management System is to handle orders and the demands of Customers. This software provides the administrator to receive order (automated) from Customer and process (by manual) it as soon as possible. We are proposing to build a fully digital, faster, error free and easy to manage Online Shopping Management System. It will help to increase the online based shopping as well as easier for administrator to manage and customers also.

1.3.2 Broad Objective

Our goal is to provide viewing product, search product, ordering product, manage customer, can give comment, etc. Administrator can see the order request, can manage product as well as order.

1.4 Scope of the project

This system works with internet connection so that user can access this system with a computer capable of browsing the intranet. While browsing they can easily interact with their status. As we are going to develop this system for two types of user's i.e administrator, and customer, the system would be versatile and can be optimized for future changes or can be

modified according to the institute demand. This Software is flexible and adaptable to changes.

1.5 Limitation of the project

Developing a RAD model and following it for the entire system development life cycle is a hard job to do specially with the only experience of software engineering theoretical knowledge. Understanding the complexity of a system for a technical person like a CSE graduate is enough to cause trouble. But at the later stage we made ourselves familiar with the overall scenario.

1.6 Methodology

For this project in data collection phase we collected only primary data.

We know that the primary data are generated within the organization. The organizations practical experience, observation, and face-to-face interview with the company executive and clients helped us generate the primary data.

1.7 Selection of project development process model

There are many situations in which initial software requirements are reasonably well-defined, but the overall scope of the development effort precludes a purely linear process, in addition, there may be a compelling need to provide a limited set of software functionality to users quickly and then refine and expand on that functionality in later software releases, in such cases, the organization chosen a process model that is designed to produce the software in Waterfall process model.

1.8 Feasibility Study:

Feasibility study determines whether that solution is feasible or achievable for the organization. This means that the tasks that we will perform are worth enough or not. There are three major areas of investigation and generating ideas about a new system. On studying the feasibility of the system, three major considerations are dealt with, to find whether the automation of the system is feasible.

- Technical feasibility
- Economic feasibility
- Operational feasibility

1.8.1 Technical feasibility

Technical feasibility addresses concern about hardware capability, reliability and availability and the skills of the development team. This study looks at the hardware and software available to perform the necessary steps for the proposed system.

Hardware Requirement	Software Requirement
1. Computer(Desktop/Laptop/Equivalent)	1.Operating System(Windows 7 or equivalent) with browser(Firefox)
2. Proper electricity Support	2. PHP
3. Adequate system memory and secondary memory.	3. MySQL

1.8.2 Economic Feasibility

Economic feasibility determines to what extent a new system, is cost effective. We consider whether the company will be able to pay cost for redesigning and whether the project will be cost effective or not.

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. Our system is economically feasible because by using the proposed system many works can be done within small time and which is not possible by man power within the same time. It also reduces the man power needed for providing the sales invoice, sales report and stock report. So they have to pay less salary where the current system needs many employee and they are paying much salary. So we can say that, if they use proposed system they will be economically benefited

1.8.3 Operational feasibility

Operational feasibility addresses concern about user acceptance, management support, and the requirements of entities and factors in the organizations external environment. The proposed system is designed from a client's point of view. So, all of the features are included only benefit the employee. The system will remove most of the disadvantage of the existing system and will be accepted by the both parties. The system is developed in a user friendly way so that it is convenient to the user to interact with and retrieve data as they desired. During our implementation of the system we found that they are very much interested and eager to our software.

Chapter 2

System Requirement

2.1 Methodology

This paper describes a methodology for the generation of software requirements for large, real-time unmanned administrative task management systems. It describes what needs to be done, how to manage the application, notice and appointment.

2.2 Requirements Engineering

Requirements engineering is, as its name suggests, the engineering discipline of establishing user requirements and specifying software systems. There are many definitions of Requirements Engineering; however, they all share the idea that ***requirements*** involve finding out what people want from a computer system, and understandings what their needs mean in terms of design.

Requirements engineering is defined in terms of its major activities

I. understanding problems
II. solution determination
III. specification of a solution that is testable, understandable, maintainable and that satisfies project quality guidelines

Steps of requirement engineering:

- Inception: define scope and nature
- Elicitation: help customer to define what is required
- Elaboration: basic requirements are found and modified
- Negotiation: priorities, what is essential, what is required to maintain –defined
- Specification: problems are specified
- Validation: customer concept and the software functional behaviors are same or not – defined.

Requirements engineering is closely related to software engineering, which focuses more on the process of designing the system that users want.

- User requirements
- System requirements
- Functional requirements
- Non-Functional requirements

2.2.2 User Requirements

User:

1) User can able to do registration
2) User can see all product details and price
3) User can search product
4) User can give order
5) User can get invoice
6) User can return product
7) User can give review of a product
8) User can see the popular product
9) User can also see the related product

Admin:

1) Admin can able to manage product category
2) Admin can able to manage product sub-category
3) Admin can able to manage product
4) Admin can able to manage user
5) Admin can able to see Order details.
6) Admin can able to view selling Report.

2.2.2 System Requirement

User:

1) User can able to do registration
   - 1.1) User have to go to system.
   - 1.2) User have to fill-up all field for registration.
   - 1.3) User have to submit Registration Form.
2) User can see all product details and price
   - 2.1) User can able to see product by category
   - 2.2) User can able to see product by sub-category
3) User can able to search product
   - 3.1) User can able to search product by category
   - 3.2) User can able to search product by Sub-category
   - 3.3) User can able to search product by price
   - 3.4) User can able to search product by product name

4) User can give order

4.1) User have to login first

4.2) User have to add product to cart

4.3) System will calculate the total payable amount.

5) System will generate invoice.

6) User can return product by maintaining return policy.

6.1) User have to select that product which he or she wanted to return.

7) User can give review of a product

7.1) User have to watch the product details

7.2) User have to press the review button

7.3) User can rated the product and can give review as comment

8) User can see the popular product

8.1) User have to visit the system

9) User can also see the related product

9.1) User have to see a product details

9.2) User then can also see the related product

Admin:

1) Admin can able to manage Product category

1.1.Admin must login

1.2. Admin can able to see all categories

1.3. Admin can able to add new category

1.4.Admin can able to delete category

2) Admin can able to manage Sub-category of product

2.1. Admin can able to see all sub-categories

2.2. Admin can able to add new sub-category

2.3. Admin can able to delete sub-category

3) Admin can able to manage products

3.1. Admin can able to see all product list

3.2. Admin can able to add new product

3.3. Admin can able to delete a product

3.4. Admin can able to edit product information

3.5. Admin can able to see in stock products list

3.6. Admin can able to see out of stock products list

3.7. Admin can able to see insufficient products list

4) Admin can able to manage user

4.1) Admin can able to see the User details

4.2) Admin can able to change status of user.

5) Admin can able to see Order details.

5.1) Admin can able to see the User's Order detail

5.2) Admin can able to see the current status of Order

6) Admin can able to view selling Report.

6.1) Admin can able to view Daily report

6.2) Admin can able to view Monthly report

6.3) Admin can able to view Yearly report

2.2.3 Functional Requirement

- Add
- Delete
- Update
- Search
- Matching
- Calculation

2.2.4 Non-Functional Requirement

- **Only security:** System will need all the username and password from the Administrator.
- **Password:** System will need all the username and password stored in database.
- **User authorization:** System will need information from Customer and Admin.

2.3 USE CASE Diagram for Online Shopping Management System:

In order to achieve the highest understanding of the project next there will be illustrations containing various cases of system. First, the following diagram is the use case diagram of the system:

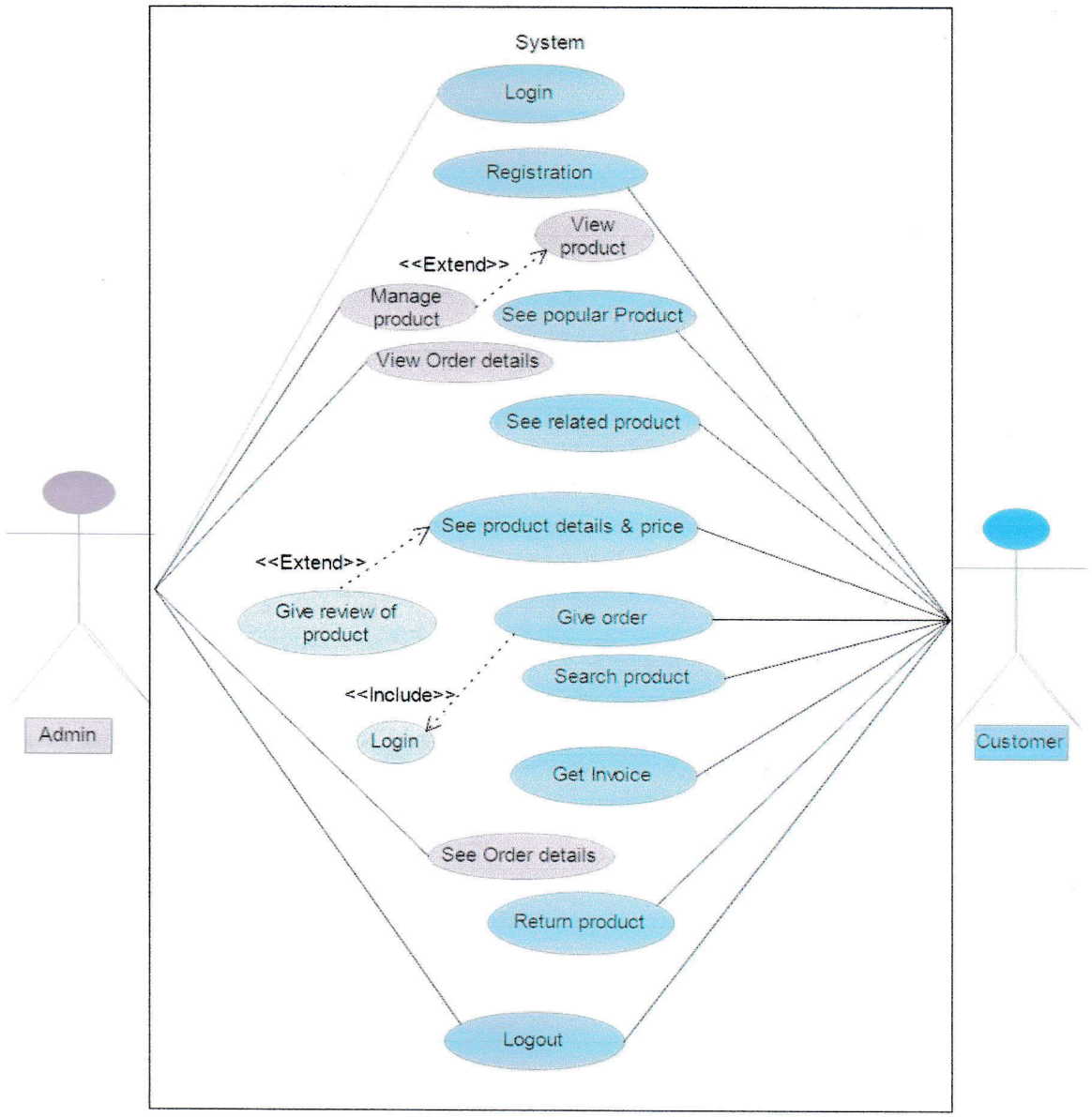


Figure 0.1: USE CASE Diagram of Online Shopping Management System

Use Case diagrams show the various activities the users can perform on the system. The System is something that performs a function. They model the dynamic aspects of the system. It provides a user's perspective of the system.

Actor: An actor is a user of the system playing a particular role.

Use case: Use case is a particular activity a user can do on the system.

Relationship: Relationships are simply illustrated with a line connecting actors to use cases.

2.4 Use Case Text

Use-case title: Login.

Actor: Administrator.

Description: I like to log-in myself into the system.

Use-case title: Registration

Actor: Customer

Description: I like to log-in myself into the system.

Use-case title: Manage Product

Actor: Administrator

Description: I like to add, delete and update the Product.

Use-case title: View Order details.

Actor: Administrator.

Description: I like to view order details.

Use-case title: See Popular Product

Actor: Customer

Description: I like to see popular product through the system.

Use-case title: See related product.

Actor: Customer.

Description: I like to see related product through the system.

Use-case title: See product details and price

Actor: Customer.

Description: I like to see product details and price through the system.

Use-case title: Give review and comment.

Actor: Customer.

Description: I like to give review and comment of product through the system.

Use-case title: Give order

Actor: Customer.

Description: I like to give order request through the system.

Use-case title: Search product

Actor: Customer

Description: I like to search request to find product through the system.

Use-case title: Get invoice.

Actor: Customer

Description: I like to get invoice through the system.

Use-case title: Return product

Actor: Customer

Description: I like to request to return product through the system.

Use-case title: Logout

Actor: Admin, Customer.

Description: I like to logout myself from the system.

Chapter 3

System Planning and Schedule

3.1 Functions of Proposed System

Admin Login	[F1]
Customer Registration	[F2]
Manage Product	[F3]
Manage Order	[F4]
Manage Brand	[F5]
Manage Category	[F6]
Manage Sub-Category	[F7]
View Product	[F8]
Order Product	[F9]
Return Product	[F10]
Add to Cart	[F11]
Give Feedback (review & comment)	[F12]

3.2 Function Description

Function description descriptive the function in details. It concerns on three factors: what is the possible input, possible output for a particular function and which table of the database uses by that function.

3.3 System Project Planning

Software project planning is the second activity of CPF. Software project management commences with a set of activities that collectively called software project planning. Before starting any project, it is compulsory to estimate the work to be done, the resources that will be required, the time will elapse from start to finish and to analyze the project to determine whether it is feasible or not.

The following activities of software project planning that have followed in this project are:

- **System Project Estimation**
- **Function Oriented Metrics**
- **Process Base Estimation**
- **Task Scheduling**
- **Project Schedule Chart**

3.3.1 System Project Estimation

The accuracy of a software project estimate predicated based on a number of things:

i. **Properly estimated the size of the product to build.**
ii. **The ability to translate the size estimation into human effort, calendar time and money.**
iii. **The degree to which the project plan reflects the abilities of the software team or engineer.**
iv. **The stability of the product requirements and the environment that supports the software engineering effort.**

Software size estimation is the most important matter that I have to consider during the software project. If the software size not calculate properly, then this will cause various problems such as scheduling problems, budget problem etc. As the project goes on. Before estimating the software size, I have to confirm that software scope is bounded.

3.3.2 Function Oriented Metrics

Function point based estimation focuses on information domain values rather that software values. Function points are computed by comparing five information domain characteristics. The information domain values are as follows

Number of external inputs – Each user input that provides distinct application-oriented data to the software is counted inputs should be distinguished from inquires.

Number of external outputs – Each user output that provides application-oriented information to the user is counted.

Number of external inquires – An inquiry defined as an on-line input those results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry counted.

Number of files – Each logical master file counted.

Numbers of external interfaces – All machine-readable interfaces that used to transmit information to another system counted.

3.3.3 Process Based Estimation

In process-based estimation, process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated. Process based estimation begins with a delineation of software functions obtained from the project scope. A series of software process activities must be performed for each function.

Activity	**CC**	**Planning**	**Engineering**		**Construction**		**Imp.**	**Total**
Function			Analysis	Design	Code	Test		
F1			0.11	0.18	0.23			
F2			0.17	0.17	0.37			
F3			0.13	0.09	0.42			
F4			0.14	0.21	0.26			
F5			0.09	0.25	0.19			
F6			0.12	0.12	0.28			
F7			0.19	0.18	0.49			
F8			0.11	0.14	0.27			
F9			0.09	0.17	0.59			
F10			0.17	0.13	0.34			
F11			0.18	0.18	0.23			
F12			0.14	0.13	0.39			
Total	0.69	0.96	1.64	1.95	4.12	0.97	0.42	10.75
Effort	6.41%	8.93%	15.25%	18.13%	38.32%	9.02%	3.91%	100%

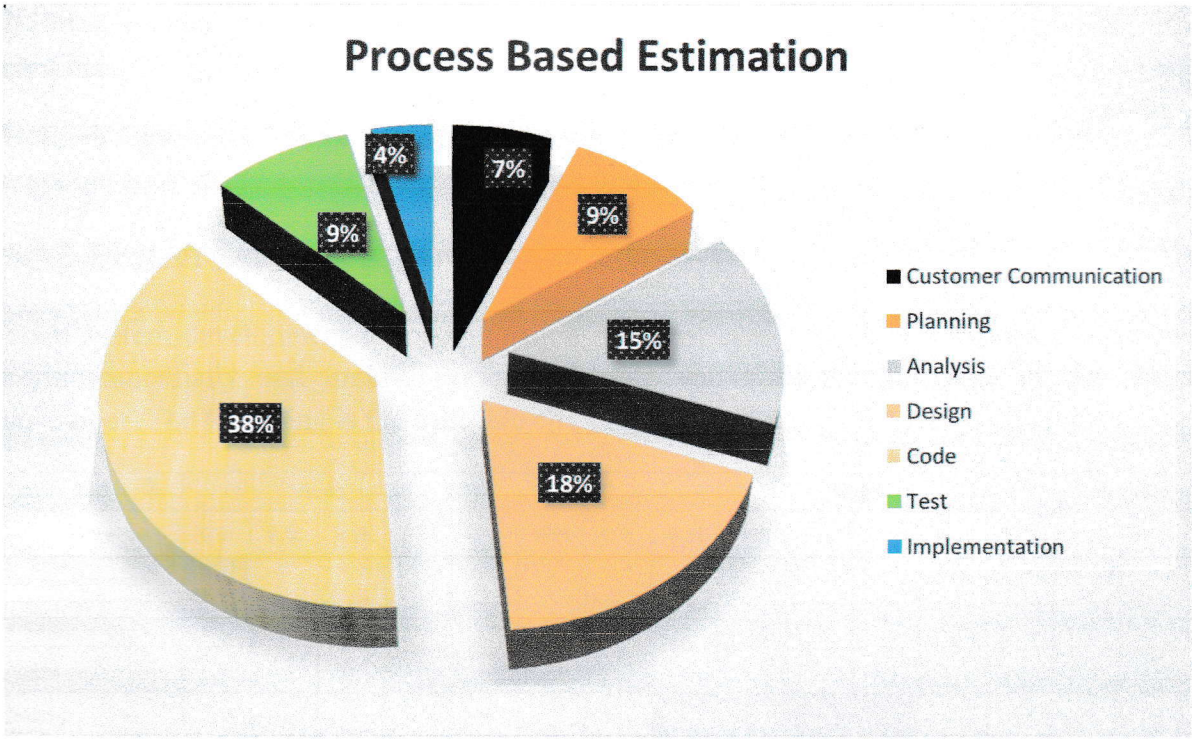


Figure 0.1: Effort Based Estimation

Time Line Calculation

Process Based Estimation = **10.75** man months

Estimated time for the project = Estimated Man Month / No. of People Working.

= (10.75/4) months

= 2.65 ≈ 3 months need for 4 people to complete this system.

3.3.4 Task Scheduling

Project scheduling is an activity of distributing the estimated efforts within the planned project duration. There are some basic rules for project scheduling. They are as follows –

Compartmentalization – The project must compartmentalize into a number of manageable activities and tasks.

Interdependency – The interdependency of each compartmentalized activity or task must be determined. Some tasks must occur in sequence while others can occur in parallel.

Time allocation – Each task to be scheduled must allocated some number of work units.

Effort validation – Every project has a defined number of staff members. It should ensure that no more than the allocated number of people has scheduled at any given time.

Defined responsibilities – Every task that is scheduled should assign to a specific team member.

Defined outcomes – Every task that is scheduled should have a defined outcome. The outcome is normally a work product or a part of a work product.

3.3.5 Project Schedule Chart

Total system development is a combination of set of tasks. These set of tasks should have done sequentially and timely. Project schedule works as the guideline of the system developer. The following is the schedule chart of this project:

Task ⬇	Month1				Month 2				Month 3				Month 4			
Time ➡	W 01	W 02	W 03	W 04	W 05	W 06	W 07	W 08	W 09	W 10	W 11	W 12	W 13	W 14	W 15	W 16
Customer Communication																
Planning																
Analysis																
Designing																
Coding																
Testing																

Figure 0.2: Project Schedule Chart

Chapter 4
Analysis and Design

4.1 Structural Diagrams

Analysis modeling uses a combination of text and diagrammatic forms to depict requirements for data, function, and behavior in a way that is relatively easy to understand, and more important, straightforward to review for correctness, completeness and consistency. This section presents resources for conventional and object-oriented analysis (OOA) methods as well as resources for UML.

4.2 Activity Diagrams for this system are given bellow:

4.2.1 Activity Diagram for login (Admin)

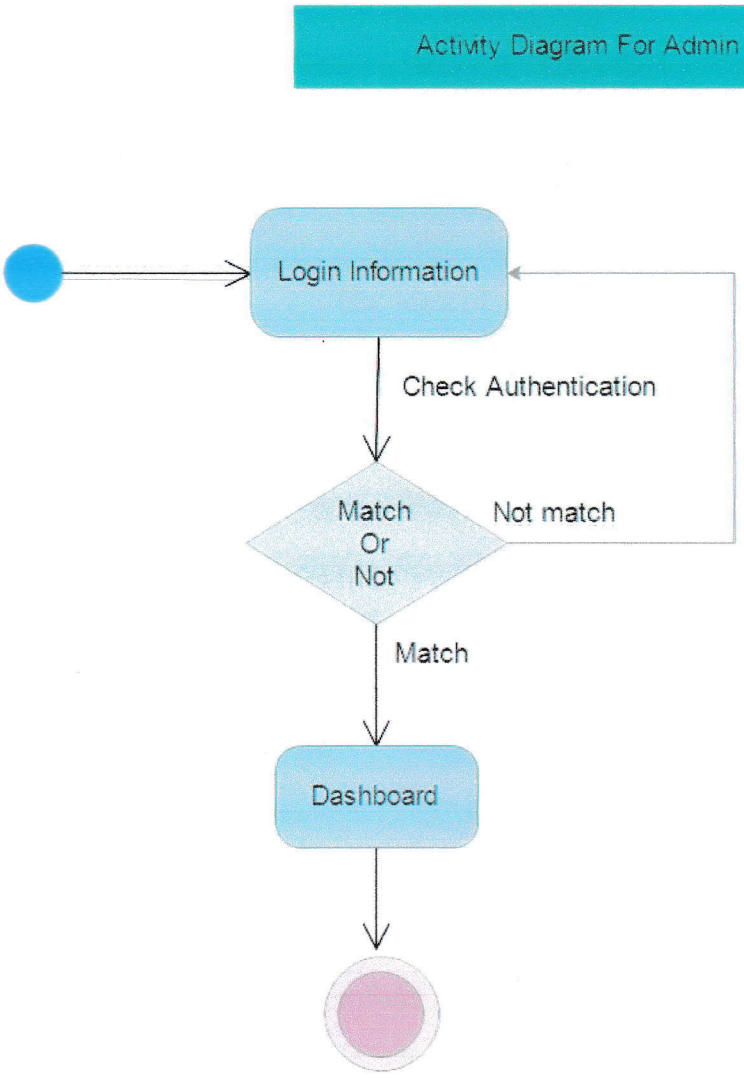
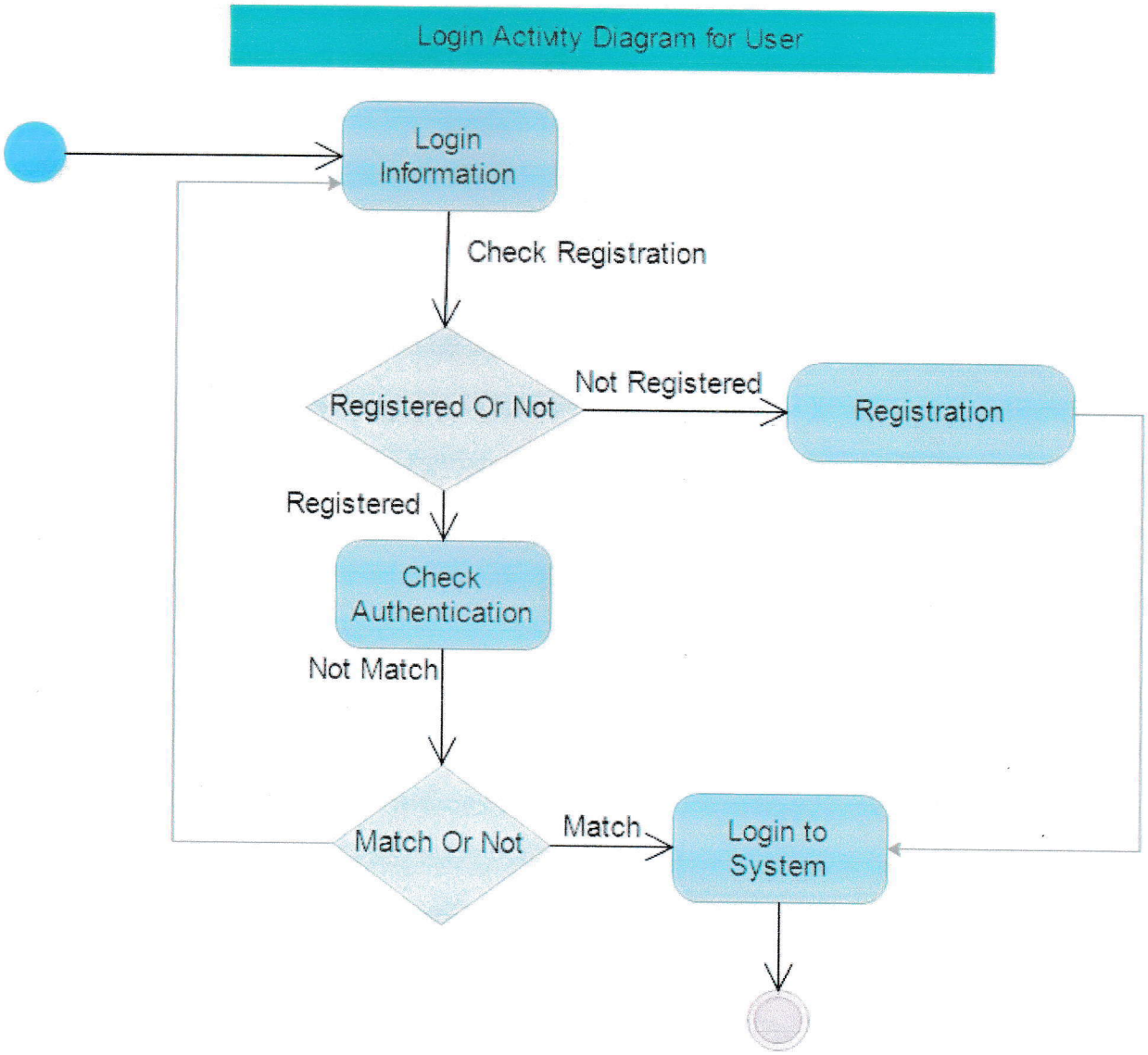


Figure 0.1: Activity Diagram for Login (Admin)

1.8.2 **Activity Diagram for login (Customer)**



4.2.3 Activity Diagram for Shopping

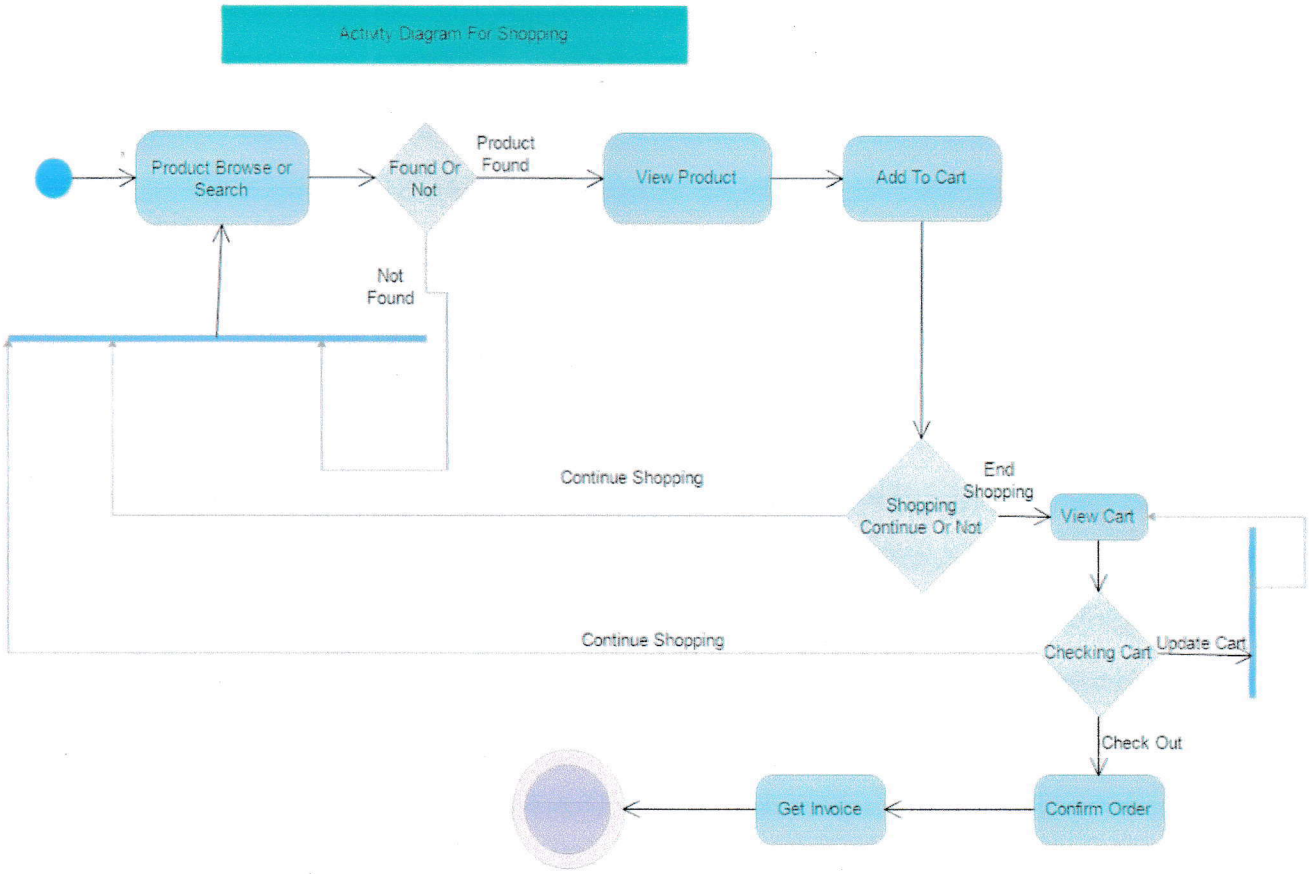


Figure 0.2: Activity Diagram for Shopping

4.2.4 Activity Diagram for Manage Order (Admin)

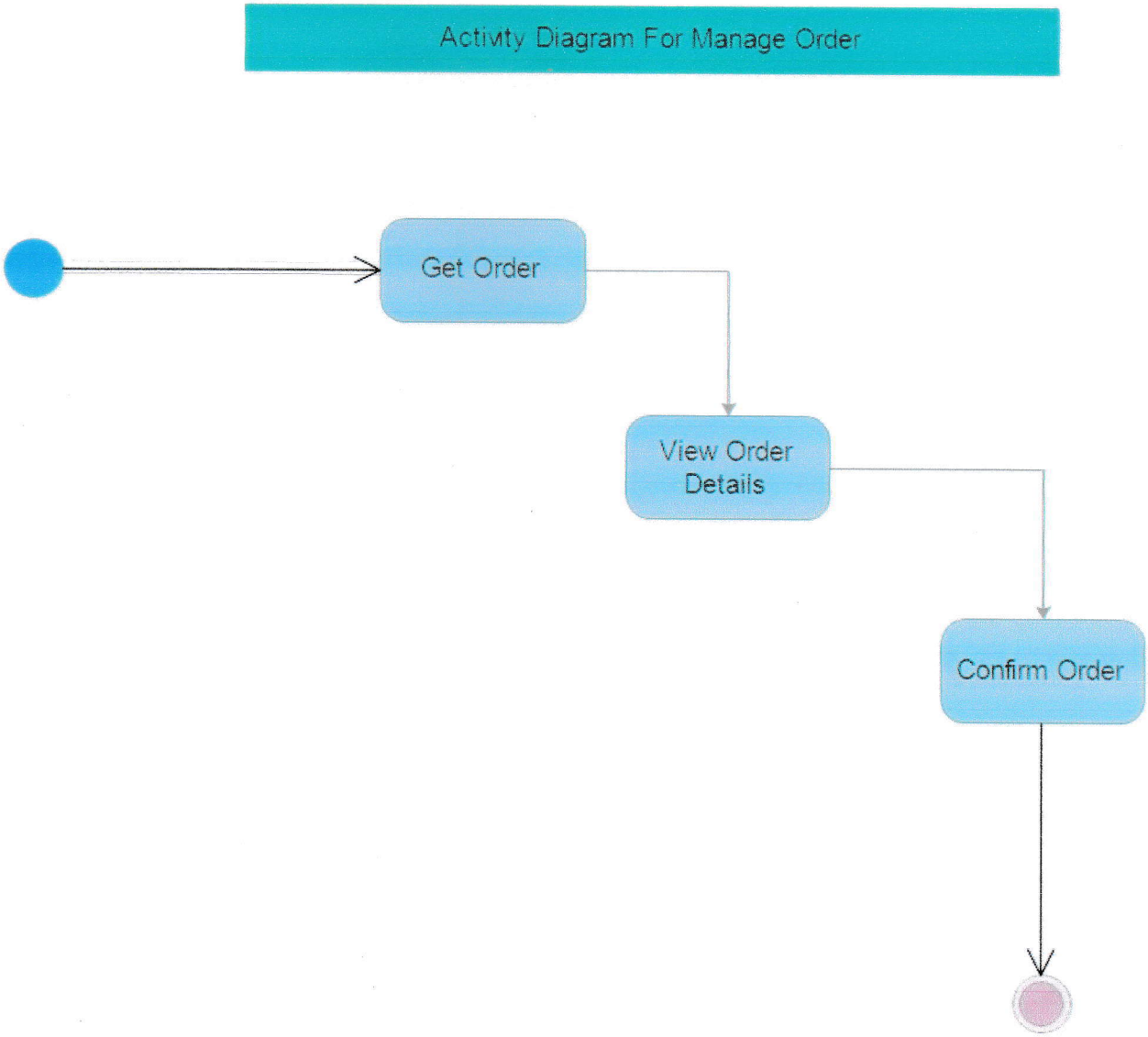


Figure 0.4: Activity Diagram for Manage Order

4.2.5 Activity Diagram for Manage Product (Admin)

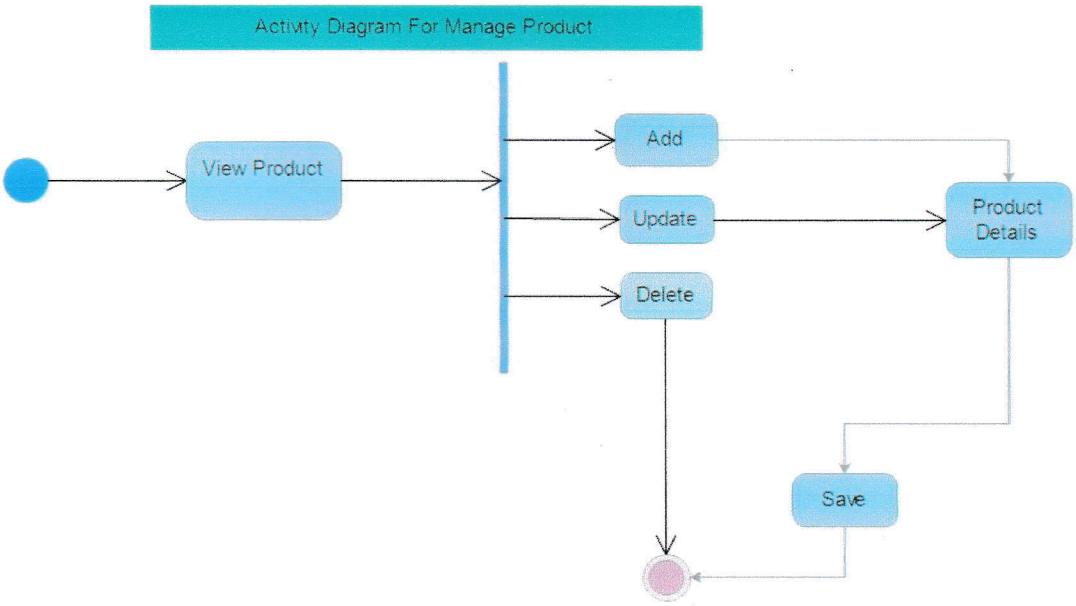


Figure 0.5: Activity Diagram for Manage Order

4.2.6 Activity Diagram for Review and Comment (Customer)

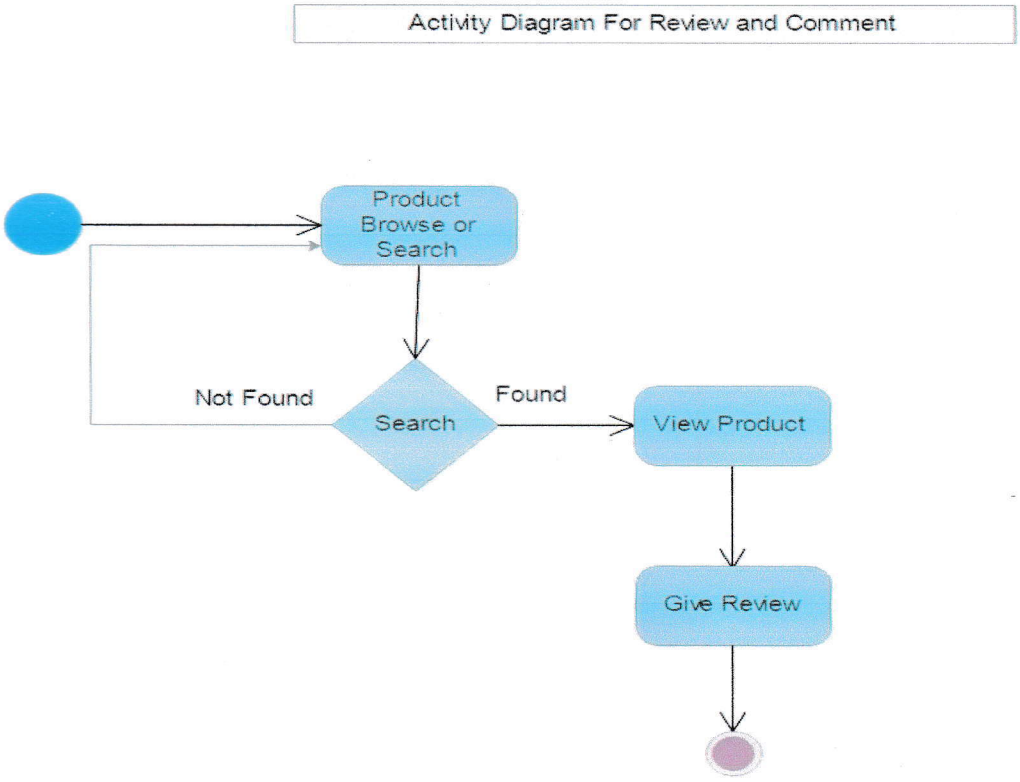


Figure 0.6: Activity Diagram for Review and Comment

4.2.7 Activity Diagram for Return Product (Customer)

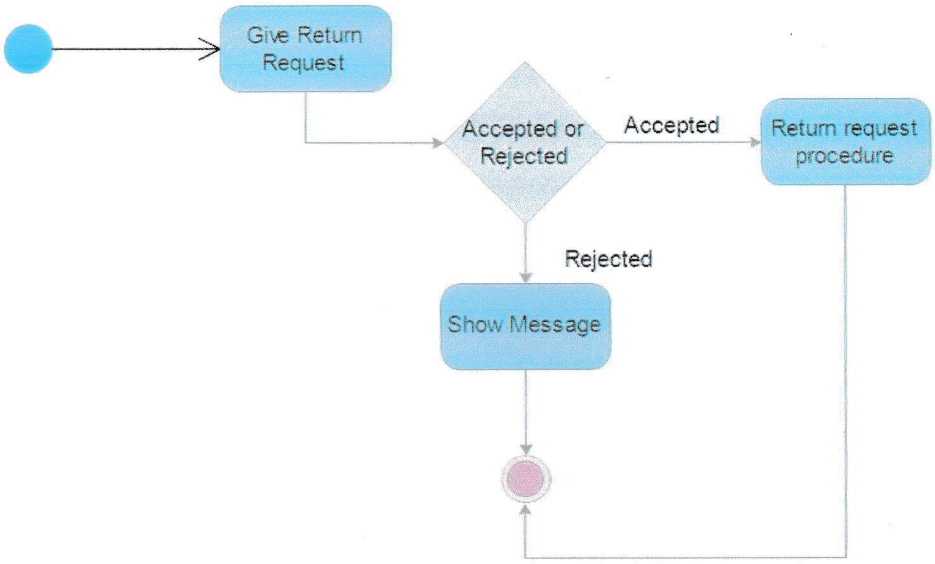


Figure 0.7: Activity Diagram for Review and Comment

4.2.8 Activity Diagram for Invoice

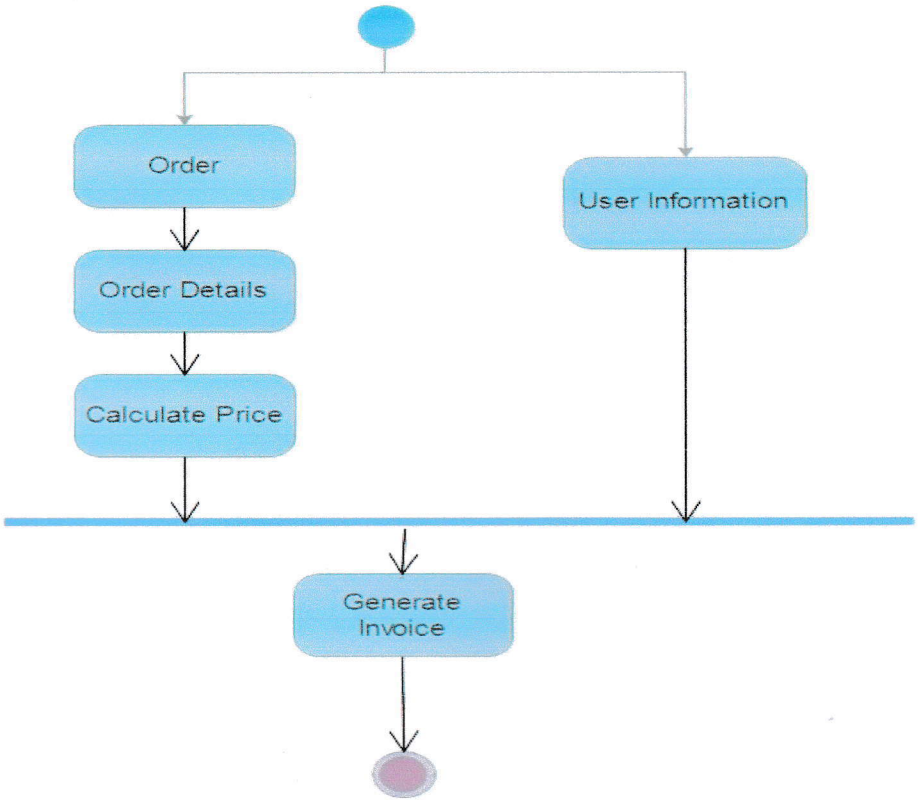


Figure 0.8: Activity Diagram for Invoice

4.3 Swim lane Diagram for this System

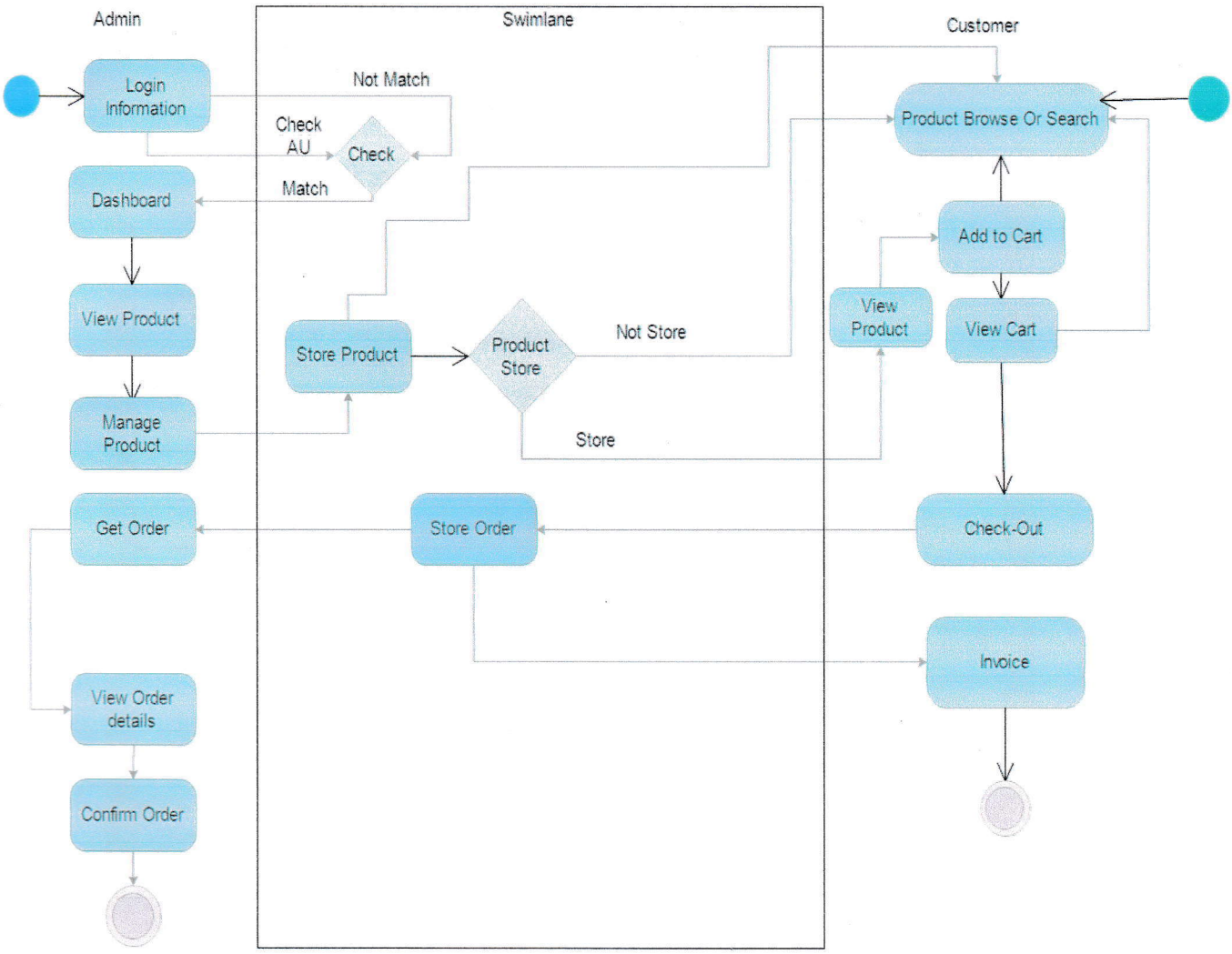


Figure 0.9: Swim Lane Diagram for Application

4.5 Sequence Diagram for the System

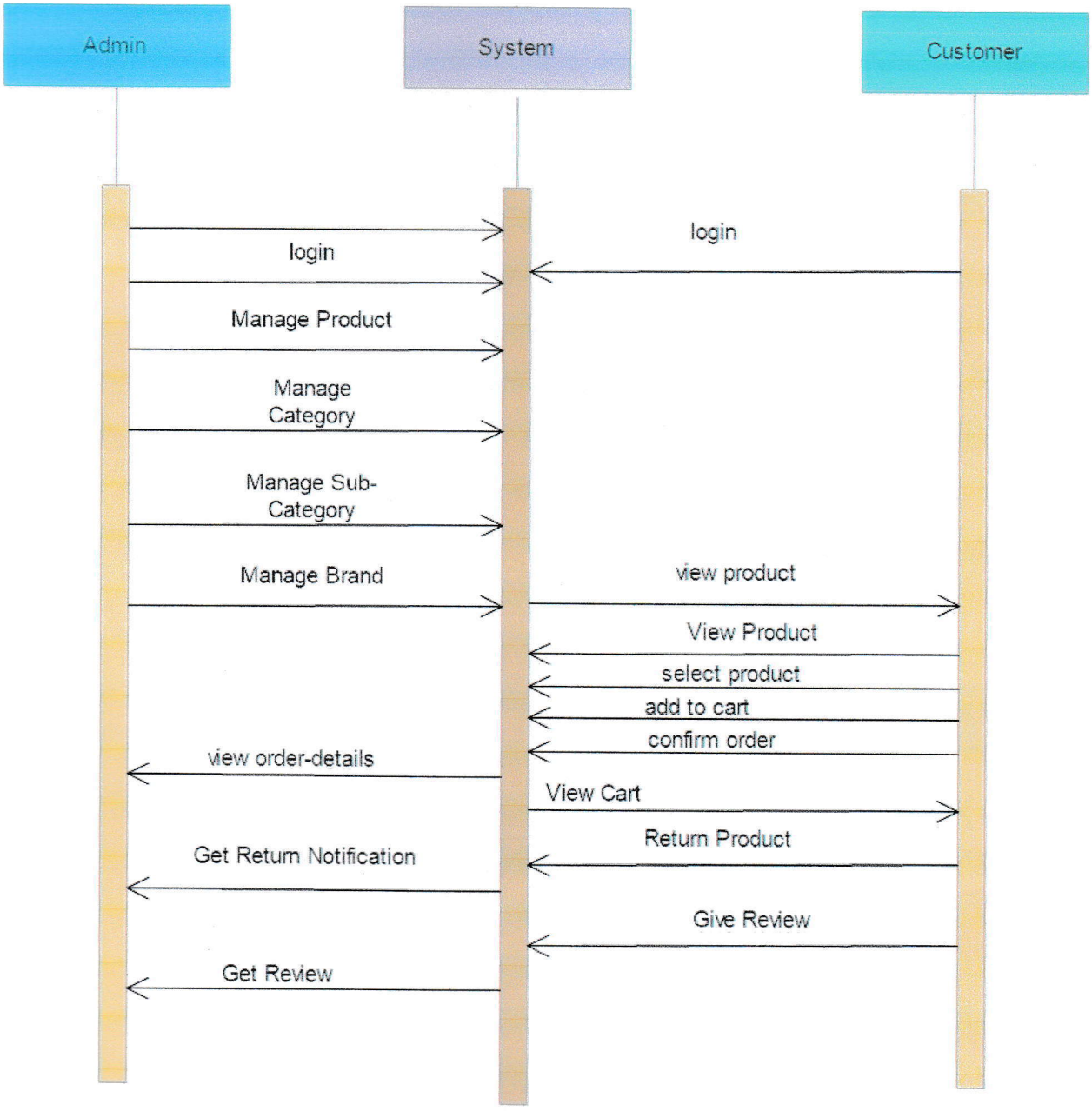


Figure 0.10: Sequence Diagram for System

4.6 ERD:

Here is the ER Diagram of the system-

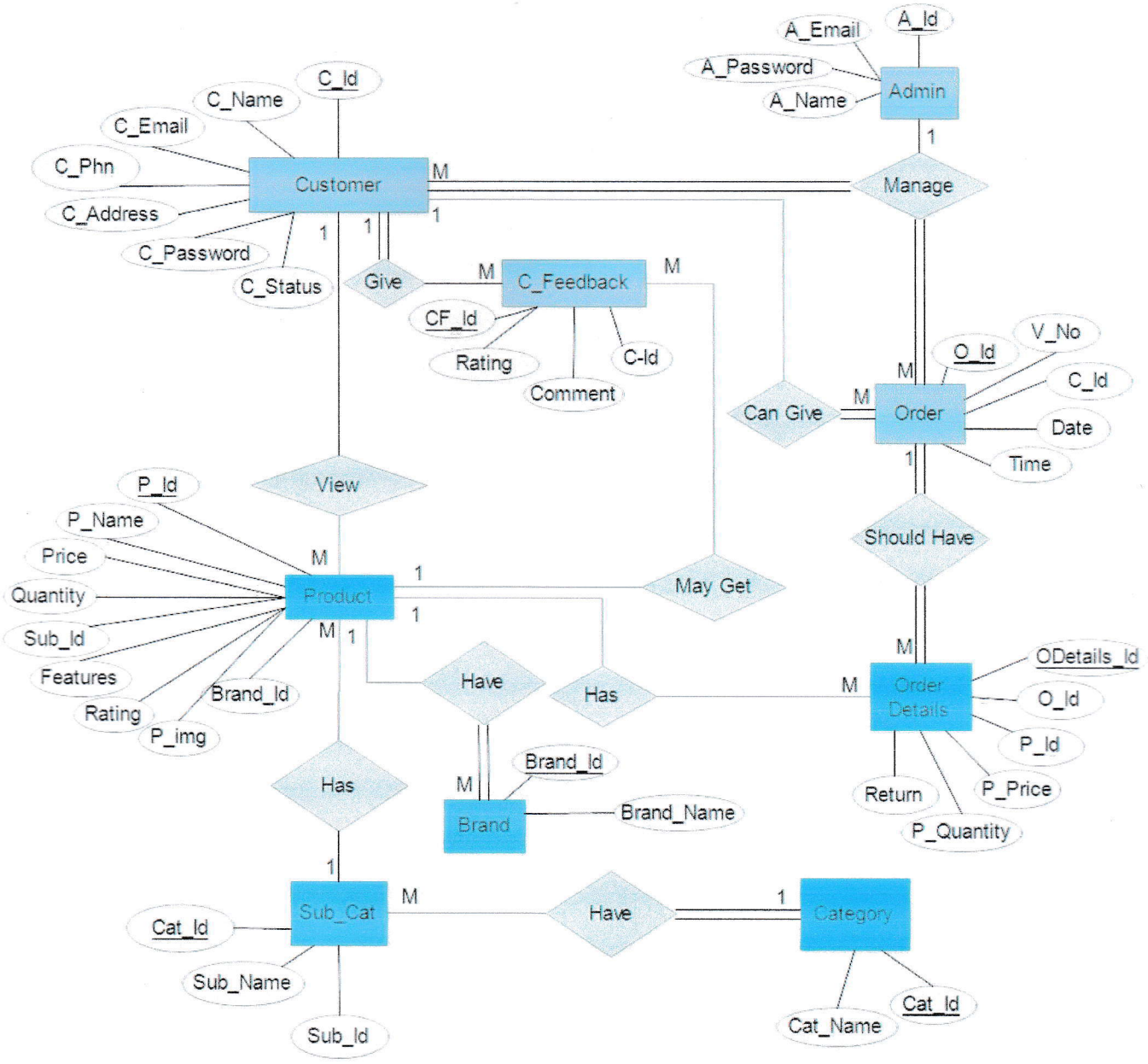


Figure 0.12: ERD Diagram

4.7 Data Flow Diagram (DFD)

4.7.1 DFD: Context label

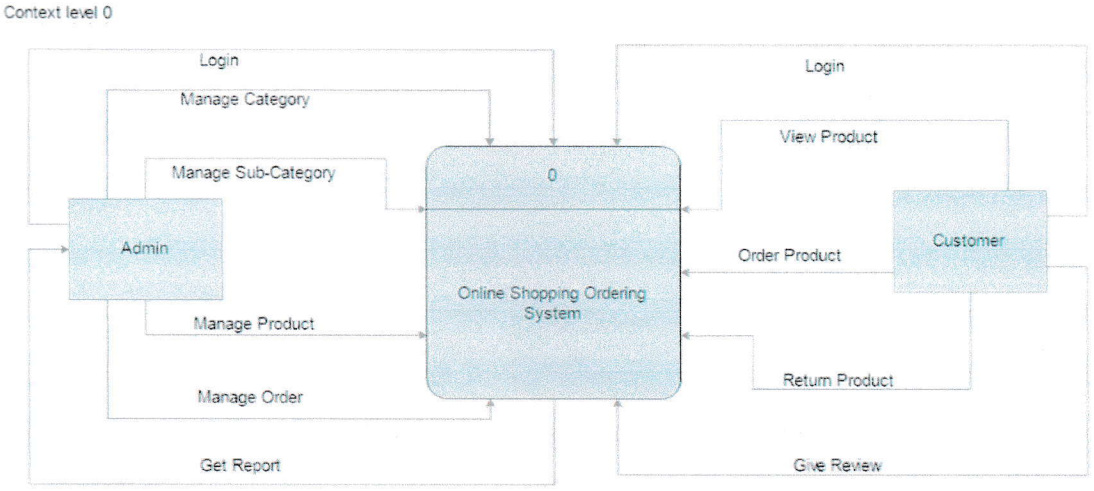


Figure 0.13: DFD Context Label Diagram

4.7.2 DFD: Level 1

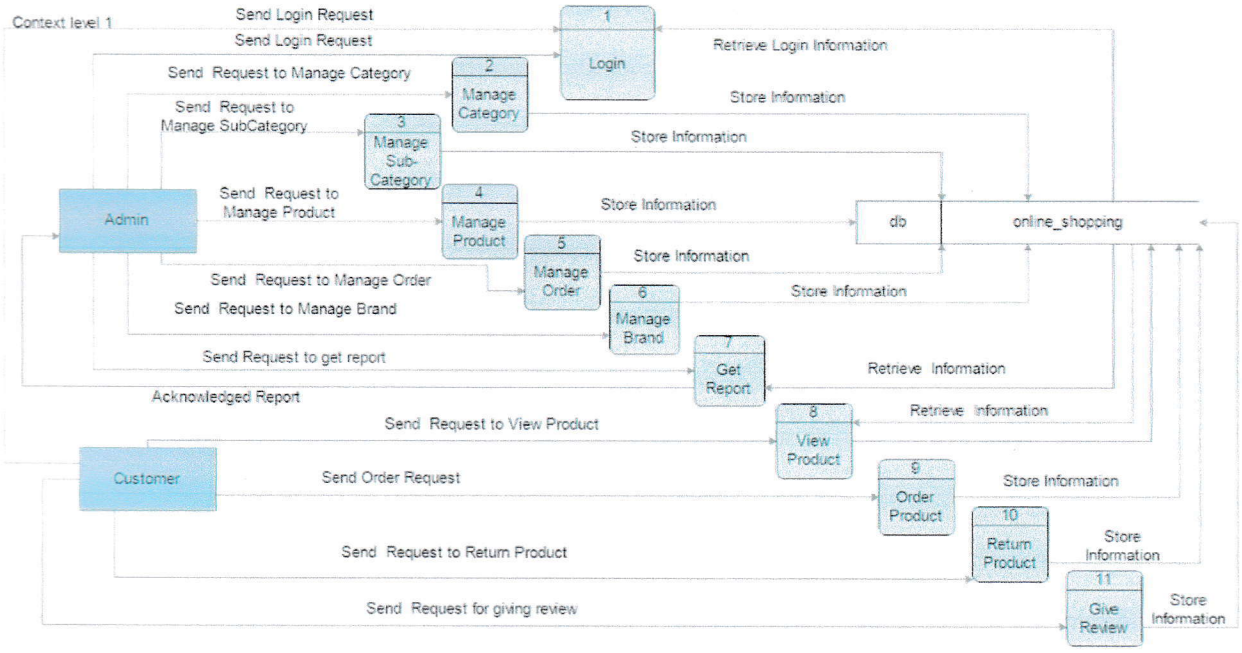


Figure 0.14: DFD Level 1 Diagra

4.7.3 DFD Level 2

4.7.3.1 Process 1

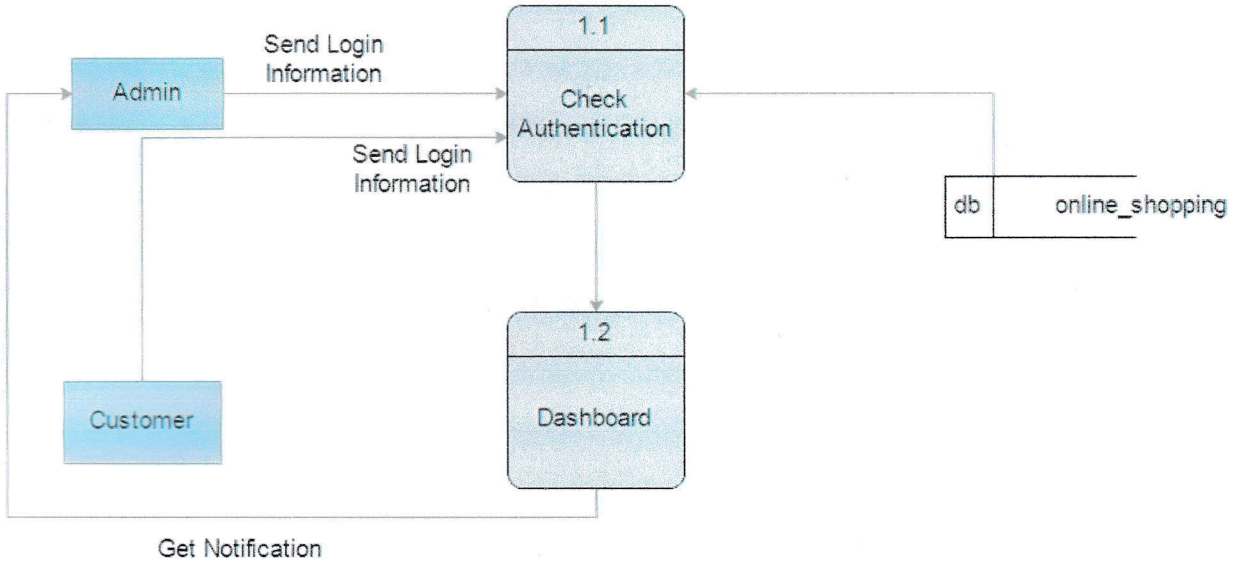


Figure 0.15: DFD Level 2 Process 1

4.7.3.2 Process 2

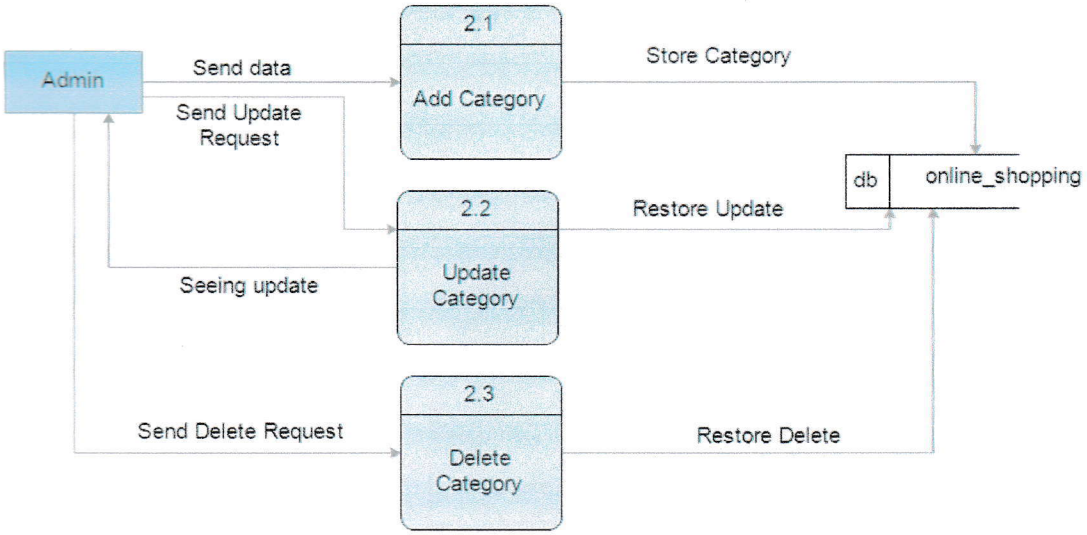


Figure 0.16: DFD level 2 Process 2

4.7.3.3 Process 3

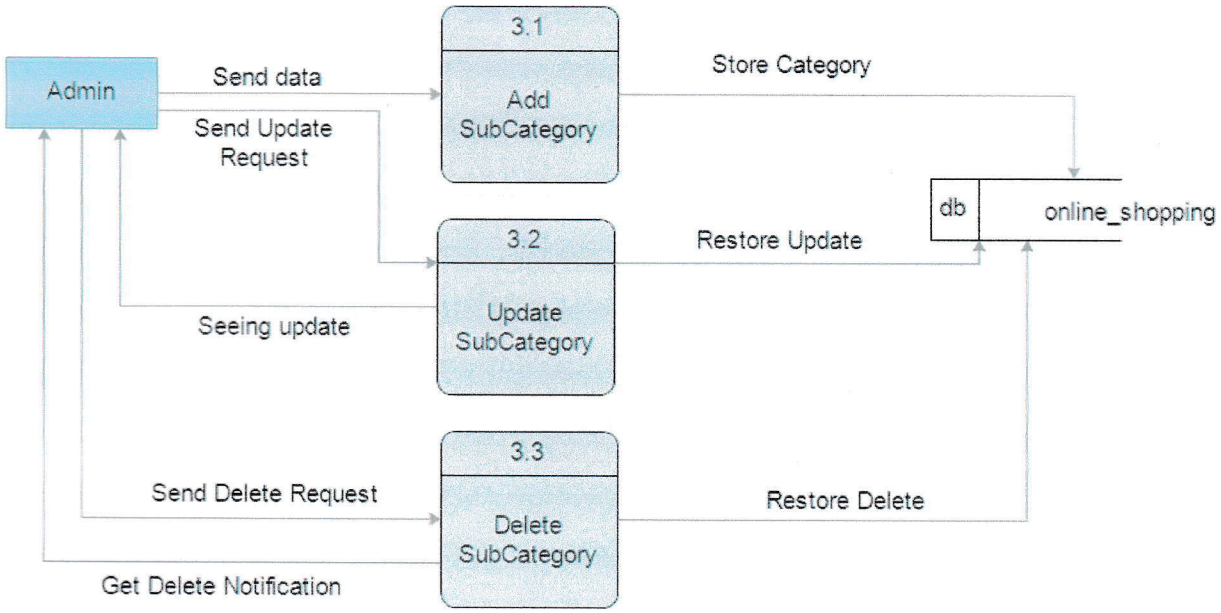


Figure 0.17: DFD level 2 Process 3

4.7.3.4 Process 4

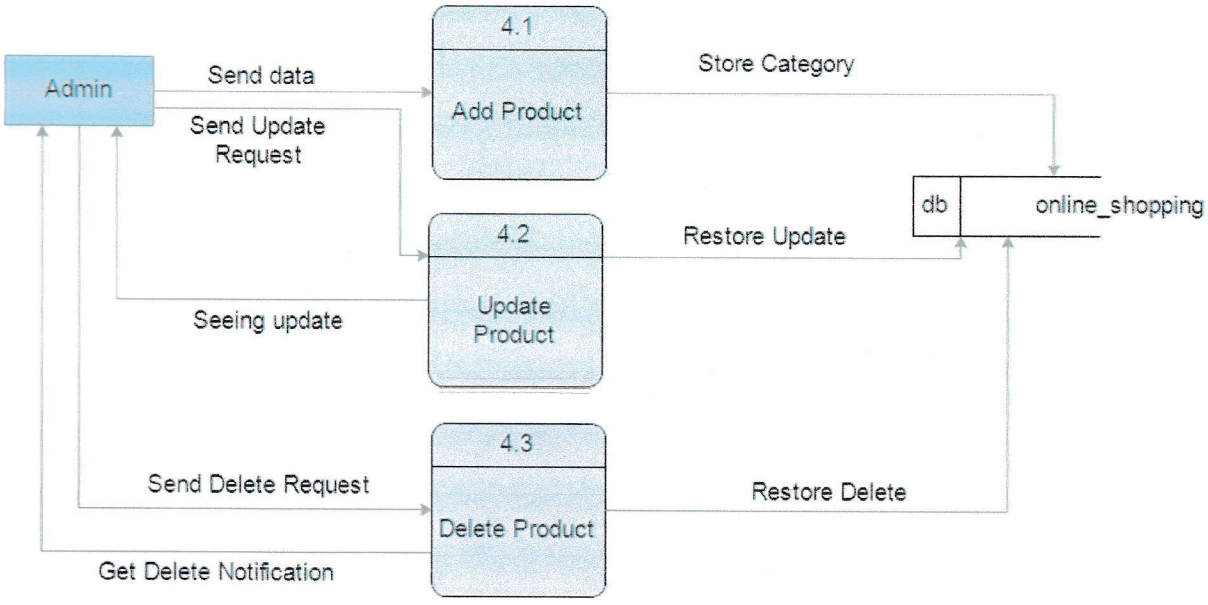


Figure 0.18: DFD level 2 Process 4

4.7.3.4 Process 5

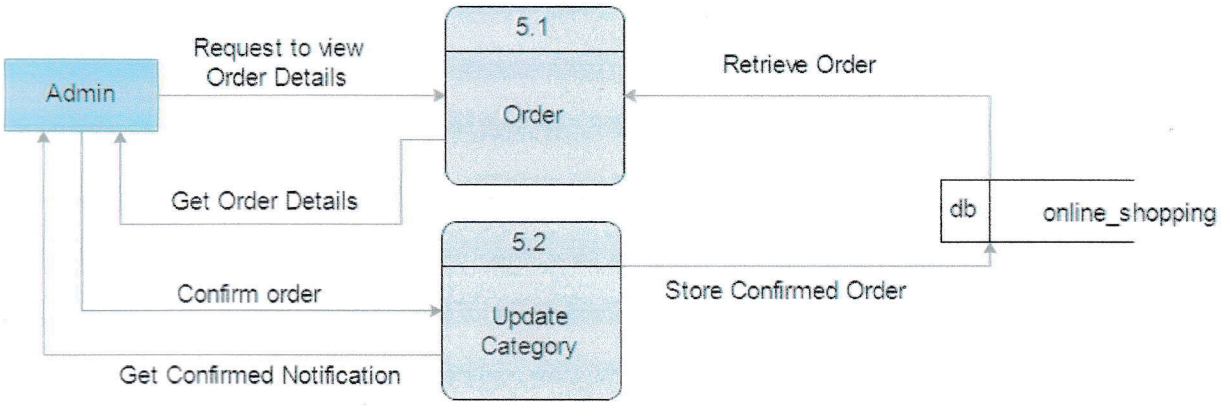


Figure 0.19: DFD level 2 Process 5

4.7.3.6 Process 6

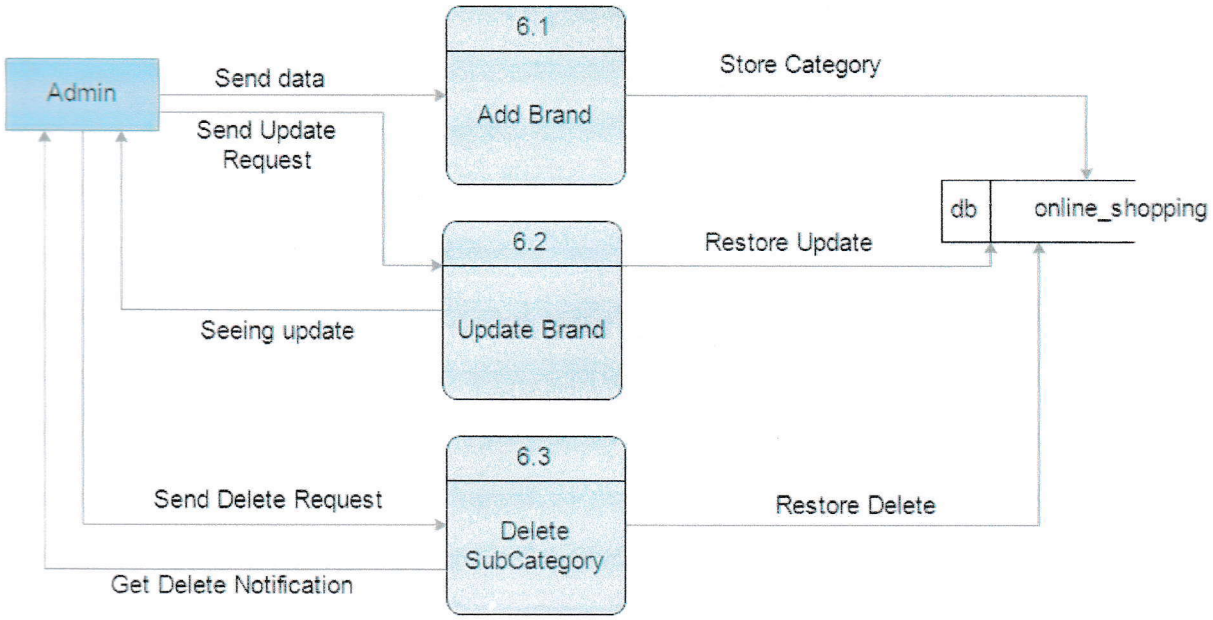


Figure 0.20: DFD level 2 Process 6

4.7.3.7 Process 7

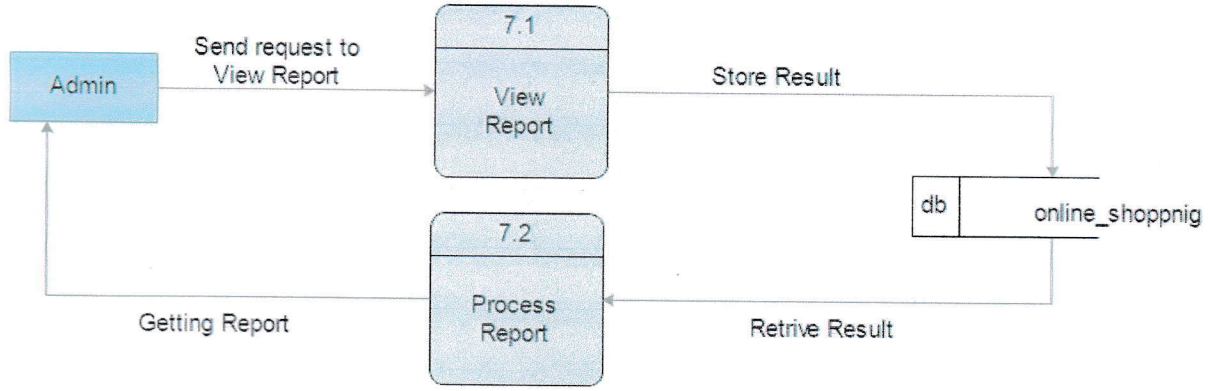


Figure 0.21: DFD level 2 Process 7

4.7.3.8 Process 8

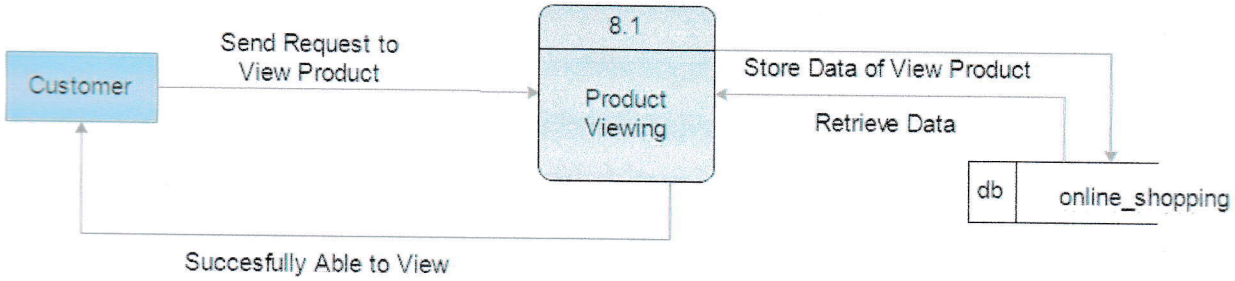


Figure 0.22: DFD level 2 Process 8

4.7.3.9 Process 9

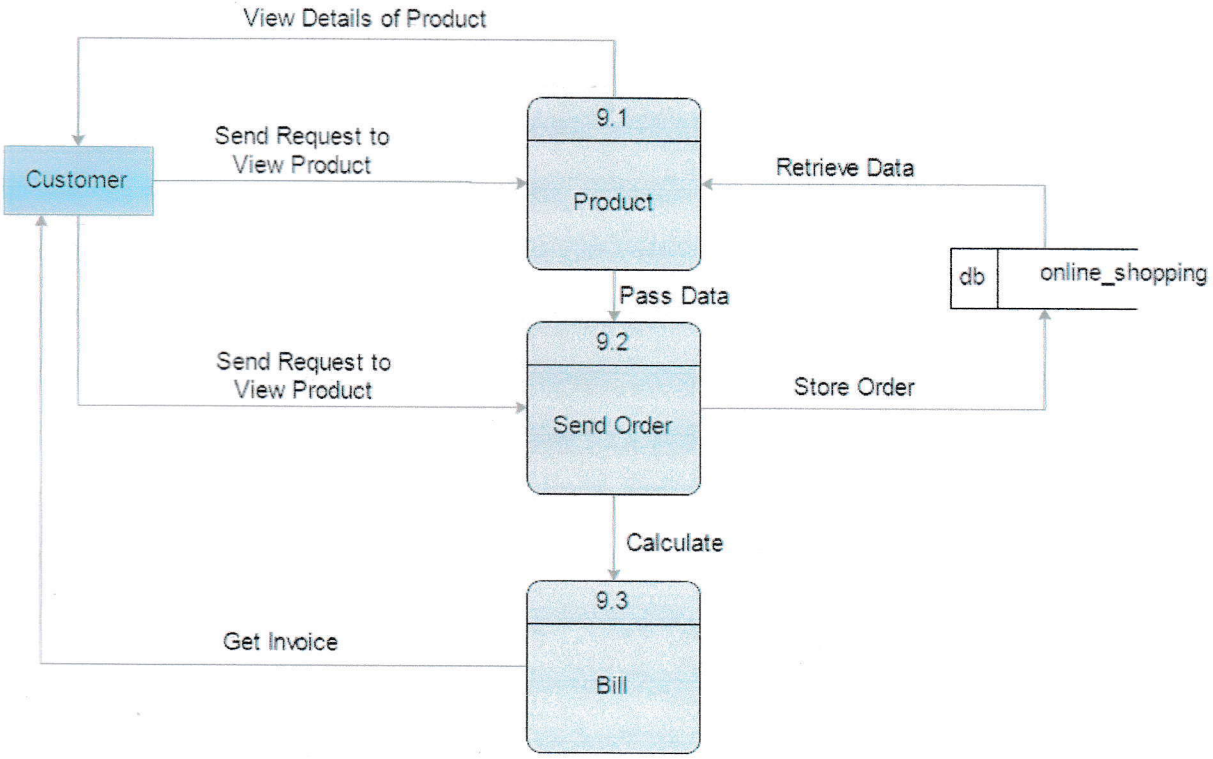


Figure 0.23: DFD level 2 Process 9

4.7.3.10 Process 10

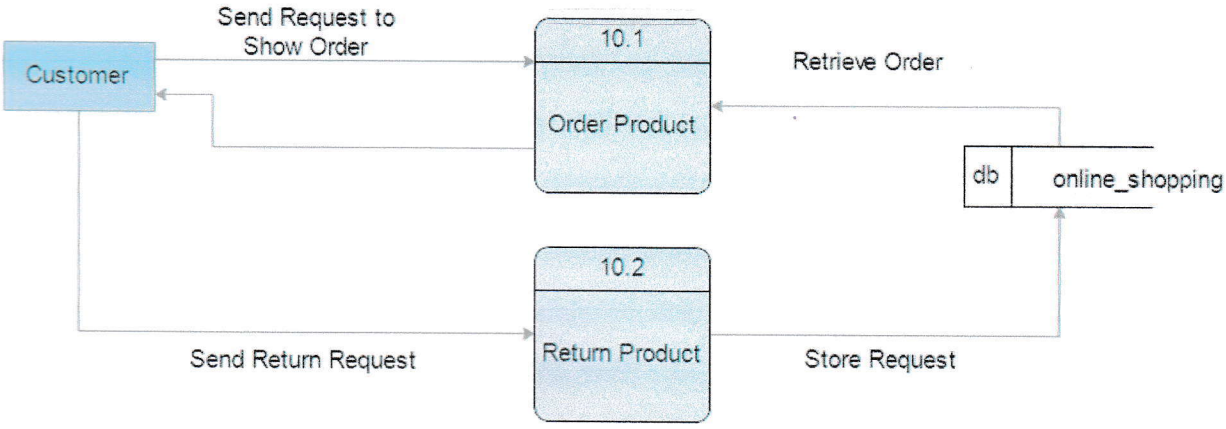


Figure 0.24: DFD level 2 Process 10

4.7.3.11 Process 11

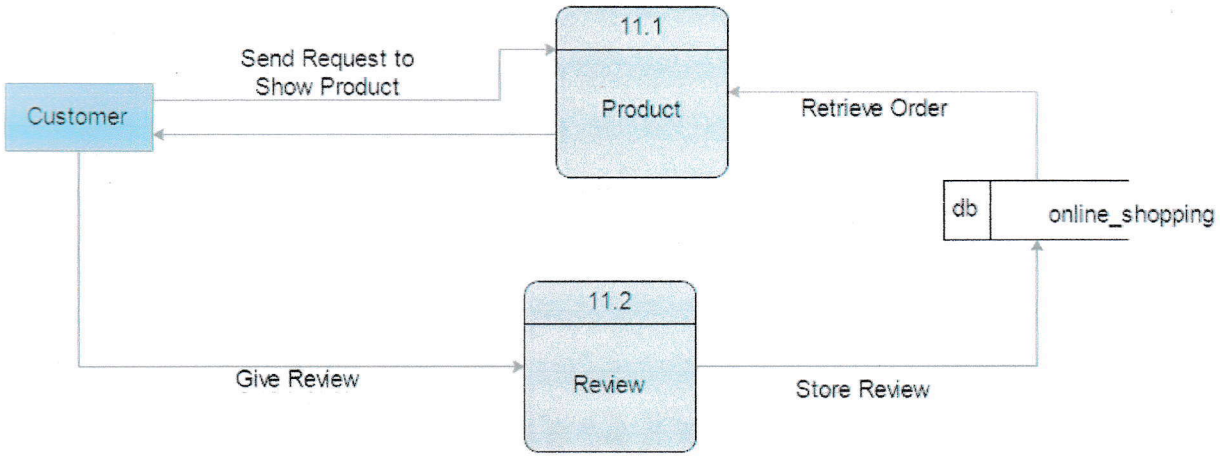


Figure 0.25: DFD level 2 Process 11

Chapter 5
System Quality and Testing

5.1 System Quality Management

A quality management software system that is automated and connects all departments essential for a regulated or ISQ-compliant company. A QMS or a TQM (total quality management) system can connect each phase in a products development lifecycle with every department in a company. This gives everyone an opportunity to provide feedback. Automated, routing, with escalation, ensure the rapid responses to inputs needed from the department. By building quality into products as opposed to forcing QA to bear the burden of the responsibility, everyone wins, engineering, regulatory, QA, manufacturing, sales and marketing

The quality of software is assessed by a number of variables. These variables can be divided into external and internal quality criteria. External quality is what a user experiences when running the software in its operational mode. Internal quality refers to aspects that are code-dependent, and that are not visible to the end-user. External quality is critical to the user, while internal quality is meaningful to the developer only.

Some quality criteria are objective, and can be measured accordingly. Some quality criteria are subjective, and are therefore captured with more arbitrary measurement.

There are mainly two types of quality

- **Internal Quality**
- **External Quality**

Internal quality:

- Test coverage
- Testability
- Portability
- Thread-safeness
- Conciseness
- Maintainability
- Documentation
- Legibility
- Scalability

External quality

- Features
- Speed
- Space
- Network usage
- Stability

- Robustness
- Ease-of-use
- Determinism
- Back-compatibility
- Security
- Power consumption

5.1.1 Software Quality Management Process

- The aim of Software Quality Management (SQM) is to manage the quality of software and development and of its development process
- A quality product is one which meets its requirements and satisfies the user
- A quality culture is an organization environment where quality is viewed as everyone's responsibility

Some of the specific SQM processes defined in standard:

- **Quality assurance process**

Quality Assurance makes sure the project will be completed based on the previously agreed specifications, standards and functionality required without defects and possible problems. Its monitors and tries to improve the development process from the beginning of the project to ensure this.it is oriented to ''prevention''

- **Verification & validation process**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it full fill its intended purpose. It is normally the responsibility of software testers as part of the software development lifecycle.

In the verification, a client will either view the software, or see it implemented in a test situation. At this point it is imperative that the client who is needed of the software is able to ascertain that this software is hitting all the parameters initially requested or desired. only when this assurance is made should the next part of the verification and validation process be started. While this is not the last chance to "tweak" the software into doing the tasks required it is part of the last steps before a project is completed, and in being too quick to approve the software as this could cause problems later, and could also result in more money required for the software's later changes. The next step of verifications &validation of software is simple. Client Company will approve the software and validate it as being what is required. This stage usually means a systematic checking off various requirements. While this might sound tedious, it is necessary part of the procedure to insure that again, the result is exactly to the specifications of all concerned. The entire verification and validation process is part of a normal sequence of quality control for software.

- **Review& Audit Process A software audit review, or software audit, is a type of software review in which one or more auditors who are not members of the software development organization conduct "An independent examination of a software product, software process, or set of software processes to assess compliance with specifications, standards, contractual agreements, or other criteria"**

Five types of reviews or audits presented in the standard:

I. Management reviews

The purpose of a management review is to monitor progress, determine the status of plans and schedules, confirm requirements and their system allocation, or evaluate the effectiveness of management approaches used to achieve fitness for purpose. This support decision about changes and corrective actions that are required during a software project.

II. Technical reviews

The purpose of technical review is to evaluate a software product to determine its suitability for its intended use. The objective is to identify discrepancies from approved specifications and standards. The result should provide management with evidence confirming (or not) that the product meets the specifications and adheres to standards and that changes are controlled". A technical review requires that mandatory inputs be in place in order to proceed:

- Statements of objectives
- A specific software product
- The specific project management plan
- The issues list associated with this product

III. Inspections

The purpose of an inspection is to detect and identify software product anomalies. Two important differentiators of inspections as opposed to reviews are as follows:

- An individual holding a management position over any member of the inspection team shall not participate in the inspection
- An inspection is too led by an impartial facilitator who is trained in inspection techniques.

The inspection exit must correspond to one of the following three criteria:

- Accept with no or at minor reworking
- Accept with rework verification
- Re inspect

Inspection meetings typically last a few hours, whereas technical reviews and audits are usually broader in scope and take longer.

IV. Walk-through

The purpose of a walk-through is to evaluate a software product. A walk-through may conduct for educating an audience regarding a software product. The major objectives are to:

- ❖ Find anomalies
- ❖ Improve the software product
- ❖ Consider alternative implementations
- ❖ Evaluate conformance to standards and specifications

The purpose of a software audit is to provide an independent evaluation of the conformance of software products and processes to applicable regulations, standards, guidelines, plans, and procedures. The audit is a formally organized activity, with participants having specific roles, such as lead auditor, another auditor, a recorder, or an initiator, and includes a representative of the audited organization. The audit will identify instances of nonconformance and produce a report requiring the team to take corrective action.

5.2 System Testing

- **Black box testing:** You don't need to know the internal design in detail or have knowledge about the code for this test. It's mainly based on functionality and specification, requirements.

- **White box testing:** This testing is based on detailed knowledge of the internal design and code. Tests are performed for specific code statements and coding styles.

- **Unit testing:** The most micro scale of testing to test specific functions or code modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code, may require developing test driver modules or test harnesses.

- **Incremental integration testing:** Continuous testing of an application as new functionality is added. Requires that various aspects of applications functionality be independent enough to work separately before all parts of the programmer are completed or that test drivers be developed as needed. Done by programmers or by testers.

- **Integration testing:** Testing of combined parts of an application to determine if they function together correctly. It can be any type of application which has several independent sub applications, modules.

- **Functional testing: Black box type testing to test the functional requirement of an application. Typically done by software testers but software programmers should also check if their code works before releasing it.**
- **System testing: Black box type testing that is based on overall requirements specifications. Covers all combined parts of a system.**

- **End to End testing:** It's similar to system testing. Involves testing of a complete application environment similar to real world use. May require interacting with a database, using network communications, or interacting with other hardware, applications, or systems.

- **Sanity testing or smoke testing:** An initial testing to determine if a new version is performing well enough to start for a major software testing. For example, if the new software is crashing frequently or corrupting database then it is not a good idea to start testing before all these problems are solved first.

- **Regression testing:** Re-testing after software is updated to fix some problem. The challenge might to be determining what need to be tested, and all the interactions of the functions, especially near the end of the software cycle. Automated testing can be useful for this type of testing.

- **Accepting testing:** This the final testing done based on the agreements with the customer.

- **Load / stress / performance testing:** Testing an application under heavy loads. Such as simulating a heavy traffic condition in a voice or data network, or a web site to determine at what point the system start causing problems or fails.

- **Usability testing:** Testing to determine how user friendly the application is. It depends on the user or customer. User interviews, surveys, video recording of user sessions, and other techniques can be used. Programmers and testers are usually not appropriate as usability testers.

- **Install / Uninstall testing:** Testing of full, partial, or upgrade install / uninstall processes.

- **Recovery / failover testing:** Testing to determine how well a system recovers from crashes, failures, or other major problems.

- **Security testing:** Testing to determine how well the system protects itself against unauthorized internal or external access and intentional damage. May require sophisticated testing techniques.

- **Compatibility testing:** Testing how well software performs in different environments. Particular hardware, software, operating system, network environment etc. Like testing a web site in different browsers and browsers and browsers versions.
- **Exploratory testing:** Often taken to mean a creative, informal software test that is not based on formal test plans or test cases; testers may be learning the software as they test it.
- **Ad-hoc testing:** Similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it.

- **Context driven testing:** Testing driven by an understanding of the environment, culture, and intended use of software. For example, the testing approach for life critical medical equipment software would be completely different than for a low cost computer game.
- **Comparison testing:** Comparing software weakness and strengths to competing products.

- **Alpha testing**: Testing of an application when development when development is nearing completion. Minor design changes may still be made as a result of such testing. Typically done by end users or others, not by programmers or testers.

- **Beta testing:** Testing when development and testing are essentially completed and final bugs and problems need to be found before final release. Typically done by end users or others, not by programmers or testers.

Chapter-06
Termination

6.1 Preface

Today is the age of modern science and information and online communication, which is critical to development of more effective operational and management process. To provide better and uninterrupted services to the employee of sailor info tech a group of Software specialist working together to keep the service all time. I was fortunate and blessed to get this lucky break to work some

of these efficient hard working friendly engineers. My earnest thanks, gratitude and salutations to these wonderful people from the deep down inside my heart.

6.2 Practicum and Its Value

In your career development – as with most life issues – there is direct relationship between effort and reward. To me, practicum can be as a transition from engineering college study life to a real world workplace through hands on experience of engineering practices.

There are several major advantages for students completing a guided Practicum:

1) Demonstration of practical application of studies
2) Development of deeper skills in the selected research area
3) Increases employability

Practicum does not offer hands-on experience only, but also the trait of "coping up" in the society. Meeting with different types of people and encountering situations gives practical orientation to life. There are many more upright issues of practicum, which only the person experiencing it can sense and believe.

It is the gateway to the professional life, bridge between theoretical and practical knowledge. Now these days, engineering job recruiters no longer consider high grand's; rather they value

the particle working experience, for which practicum proves to be vantage for the fresh entry level engineers in the job market.

Student of College of Engineering and Technology (CEAT) at SU go for this practicum program carrying 6 credit hours' weight, which goes for a semester long and usually after the completion of the course work. A report submitted after the completion of the practicum followed by a presentation and a comprehensive examination on the overall four years' education

6.3 Conclusion:

The biggest experience working at this project is indeed being a part of designing and implementing software. Our most experience was round the designing issue. We have learnt a lot of new things which was so much unknown to us. We have also learnt some technical issues which help to do better in future life. The following indicator will indicate some of our technical issue which we have learnt and implemented from this project.

- The designing strategy of a web based project.
- The analysing strategy of a web based project.
- New environment of programming languages.
- Developing a new project using the existing project.
- How to coding SQL and designing by PHP, PHP Framework code igniter, HTML and CSS.

We faced with some limitation of this area in Bangladesh. Maximum Bangladeshi people do not understand the benefit of online system. It is a big problem for our next generation. Therefore, government should take proper steps to spread the beneficial information about online system to the country people.

6.4 Bibliography

Books and Handouts

Kendall, E. Kendall. System Analysis and Design. 4th ed. New Delhi: Prentice Hall. 1999.

Miller, Dave. Data Communication and Networks. 1st ed.2006.

O'Brien, James and M. Markas, George. Management Information System. 8th International ed.2008.

Pressman, Roger S. Software Engineering: A Practitioner's Approach. 5th ed. Boston: McGraw Hill, 2004.

Silberschattz, Abraham, Korth, Henry F., and Sudrashan S. Database System Concepts. 4th ed. Boston: McGraw Hill, 2002.

S. Pressman, Roger, Software Engineering A Practitioner's Approach, 6th edition, 2008

Websites

- http://stackoverflow.com/
- **http://www.slideshare.net/naparnanayak/rmmmrisk-managementmitigation-and-monitoring**
- *http://www.w3school.com/*
- *http://php.net/manual/en/intro-whatis.php*
- "Project risk management: Lessons learned from software engineering environment", <http://home.gwu.edu/~kwak/Project_Risk_Management.pdf>
- "Introduction To Function Point Analysis" http://www.softwaremetrics.com/fpafund.htm
- "Estimation Techniques: Function Point Analysis" http://geekswithblogs.net/Prabhats/archive/2007/03/01/107632.aspx