# Chapter 2 – Analysis

# 2.1 Introduction of Analysis

Analysis is an important step for software project as it is critical in the success and failure of the system. Analysis is done to achieve and create a quality software that meets the user’s needs within budget and on time. There are four types of activities involved in analysis such as:

* Eliciting requirements: It is also known as requirements gathering and it communicates with customers and users to find what their requirements are.
* Analyzing requirements: Gathered requirements are determined and checked whether it is unclear, incomplete, ambiguous or contradictory and then resolving these issues.
* Requirements modelling: Requirements are documented using different formats or various forms, such as user stories, use cases, natural language documents or process specification.
* Review and Retrospective: During team meeting or review team members reflect on what happened in the iteration and identifies solution for improvement going forward.

Requirement Analysis is an important step of software development. Analysis activities is done to ensure that the product has all the functions and features required by the client. Requirements are gathered through user needs and document is created after using an analysis technique. Analysis ensures that the final product gives what the user demands rather than trying to mold user expectations to fit the requirements.

There are different types of analysis methodology and the methodology that I will be using for this project is Soft system methodology (SSM). SSM focuses more on people or users than technical during analysis. CATWOE analysis is a tool to prepare a comprehensive root definition model. CATWOE can be used for any project and organization but done when user needs should be considered for better quality and enhance productivity. All the factors involved internally or externally in an organization can be analyzed using this method. It encourages open discussion of problems, perceptions and needs, joint problem solving as well as user participation and commitment. (CATWOE Analysis, 2019) The analysis done for this system is shown below:

A close up of a device

Description automatically generated

Figure 1: CATWOE Analysis

CATWOE stands for

* Customers/Clients: Clients are the receiver of the outputs from the system. For this system they are asked questions about the problem with current system and how they will feel or react about proposed system.
* Actors/Agents: Agents/Actors are the user that undertake the activities involved with the system such as employee for this system who will use it. Following questions are asked for actors such as how they feel about proposed system and what are the impact of new system on them.
* Transformation: Transformation refers to the changes that occurs on data or processes after the development of a system. This step is carried out for this project by listing the inputs and finding the nature of changes it will go through while producing output.
* Worldview: For, this project the surroundings such as what is going on in and outside of the organization that may influence the development of this system is considered. This step involves engaging the process or system under analysis in its wider context to highlight the consequences or relevance of such process to the overall system.
* Owners: The person who owns an organization is also involved in an analysis during this system development. We need to consider what role they will play in analysis and will they help in analysis. Owner need to be considered for the system as they can stop the project and decide whether to go ahead with changes or not.
* Environment: The analysis is to find about the external constraints under which this system works, and which might hamper or restrict the development or changes to the system. Usually political, legal, economic, social, technological factor e.t.c are considered for their impact on system development.

# 2.2 Feasibility Study

Feasibility Study means the study about this project or its idea to find out whether it is legally, technically, socially, economically feasible or not to develop. It helps us study whether this project is worth the investment or not. It’s an initial design stage of project which shows or indicate if a project is possible or not. (Feasibility study, 2019) Different factors or types of feasibility are studied to measure the likelihood and ability of completing a project successfully.

Feasibility Study types:

1. Technical Feasibility: Technical feasibility refers to the feasibility or availability of the hardware and software to complete the proposed project. For this project the technology such as hardware resources, programming software and other all software tools are available to complete it without hindrance. So, this project is technologically feasible to complete it.
2. Economic Feasibility: Economic feasibility is known as financial or cost evaluation studies to find whether the project is economically feasible or not. This project is economically feasible as the cost does not extend from the estimated budget. The cost was estimated for things such as electricity, internet for research, hardware or software components e.t.c.
3. Operational Feasibility: Operational feasibility refers to the study of proposed software whether it will be fully operational after development or not. This project is operationally feasible as the current manual system will be an automated user-friendly system. With the required training the users will be able to operate it easily. Usability will be high as this system has high operational feasibility.
4. Legal Feasibility: Legal feasibility is about the software being developed whether it is legally available to develop or not. This project is legally feasible as there is nothing unethical about it. The proposed system is legal, and it is not against the law or rules of our country.
5. Schedule Feasibility: It refers to the task that will be needed to be completed on different schedule as proposed. This project is feasible as the task for software development are sub-divided into small task and time and days is assigned for each task. So, completing each task in their deadline helps achieving schedule feasibility for this project.

# 2.3 Requirement Analysis

## Functional Requirements

Functional Requirements are the specific functions and behaviors of the system. It specifies the things system perform. Functional Requirements document are prepared for the user and they should be able to understand it without any technical knowledge. Functional Requirements shows outlines of workflows performed by the system, include functions performed by specific screens and other organization requirements it must meet. The functional requirements of this system are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependency** |
| FR1 | Register User | User should be registered to log in the system | To create user profile and access system | N/A |
| FR2 | User login | User login is needed for the security of system | To allow access and secure the system | FR1 |
| FR3 | Update User | User can update profile and change password | To allow user to edit their details | FR2 |
| FR4 | Dashboard | All the features are present so, user can use it easily through dashboard | To allow user to perform task | FR2 |
| FR5 | Remove User | User can remove their account | To allow user to delete their account | FR2 |
| FR6 | Add Employee | Employee details can be added | To allow employee information to be added | FR2 |
| FR7 | Update and Remove Employee | User can update and delete employee details | To edit, delete employee details | FR6 |
| FR8 | Add Product | Product can be added | To allow user to add product sock | FR2 |
| FR9 | Update and Remove Product | User can update and delete product details | To edit, delete product or stock details | FR8 |
| FR10 | Generate Bills | Bills can be generated for customer | To allow customer bills to be generated | FR11 |
| FR11 | Add Customer | Customer or sales information can be added | To allow sales details to be added | FR2 |
| FR12 | Update and Remove Customer | Customer or sales information can be edited and removed | To edit and delete customer or sales details | FR11 |
| FR13 | View Report | Generating sales, stock and employee report | To generate sales, stock report | FR6, FR8, FR11 |
| FR14 | View Data | Inserted data of employee, customer and stock can be viewed | To allow user to view inserted data | FR6, FR8, FR11 |

## Non-Functional Requirements

Non-Functional Requirements refers to the indirect supporting features of the system. It covers all the other requirements that is not included in functional requirements. It describes how the system works and specifies how the system should behave. It is a constraint upon the systems behavior and known as quality attributes of a system. The non-functional requirements for this system are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependency** |
| NF1 | Security | System should be secured, and unauthorized access not allowed. | Security is needed to secure stored data | NF2 |
| NF2 | Performance | Output of system should be fast, and system should not lag or stuck | Performance helps complete task faster and gives faster response time | N/A |
| NF3 | Reliability | Output of system should be precise or correct and data integrity is needed | Performance should be fast with reliable output | NF2 |
| NF4 | Usability | System should be easy to use and understand. Non-technical user should also feel easy | Accessibility should also be included for greater usability | NF3 |
| NF5 | Scalability | The system should be able to store any amount of data given. | To store data inserted by users without any system or memory issue | NR1, NR2, NR4 |
| NF6 | Portability | System should be cross platform that is run on different platform | To increase portability different platform should be supported by system | NR2, NR3, NR4 |
| NF7 | Maintainability | System should be maintained regularly | To prevent errors and remove bugs system should be maintained | NR4 |

## MOSCOW Prioritization

## 

MoSCoW is the popular requirements managing prioritization technique and it is also known as MoSCow analysis. This method is used to identify or prioritize the user stories gathered according to the importance for this system. It was initially designed as a prioritization framework for time boxed projects and DSDM. It stands for four categories known as must have, should have, could have, won’t have.

Must have: It is used for prioritizing a basic and non-negotiable requirement of this project or system. It prioritizes the requirements without which the system might not function as needed.

Should have: It is below the must have for prioritization and important but not vital as must have. It tells about requirements without which system function properly but if added adds significant value to this system.

Could have: It is also known as nice to have function. If left out it will not impact on core function and have smaller impact on outcome but if such requirements are added it creates a significant system.

Won’t have: It tells about the requirements which are irrelevant for this current system or user stories that are not to be a priority for this current time frame. It helps listing the functions that will not be in this system.

The MoSCoW prioritization of this system are as follows:

|  |  |  |
| --- | --- | --- |
| **ID** | **Title** | **MoSCow** |
| FR1 | Register User | Must have |
| FR2 | User login | Must have |
| FR3 | Update User | Must have |
| FR4 | Dashboard | Must have |
| FR5 | Remove User | Must have |
| FR6 | Add Employee | Must have |
| FR7 | Update and Remove Employee | Should have |
| FR8 | Add Product | Must have |
| FR9 | Update and Remove Product | Should have |
| FR10 | Generate Bills | Must have |
| FR11 | Add Customer | Must have |
| FR12 | Update and Remove Customer | Should have |
| FR13 | View Report | Could have |
| FR14 | View Data | Should have |
| NF1 | Security | Must have |
| NF2 | Performance | Should have |
| NF3 | Reliability | Should have |
| NF4 | Usability | Could have |
| NF5 | Scalability | Should have |
| NF6 | Portability | Could have |
| NF7 | Maintainability | Could have |

## System Requirement Specification (SRS)

SRS is an important information for user as it tells about the specific hardware or system that is needed to run the developed system. SRS gives information about minimum or maximum requirements for the software to run properly. It is a document that describes the functions and features of a system as well. Every software should have requirement specification so that user knows the required hardware and software components that will be needed to run the software.

System Requirements

|  |  |
| --- | --- |
| Operating System | Windows 7/8/8.1/10 |
| System Type | 32-bit or 64-bit |
| Processor | Intel Core 2 Duo or higher |
| RAM | 2 GB or higher |
| Programming Language Used | C# |
| Database | Microsoft SQL Server |

# 2.4 Use Case

Use case is a part of analysis which identify, clarify and organize system requirements. It is also known as diagram representing high level interactions with a system. It is important because it defines the roles of different users in a system. It is a behavioral diagram that represents functionality using different notations such as Actor, Use-case and association. The use case diagram of the Mobile Shop Management system created using Star UML is shown below.

A close up of text on a white background

Description automatically generated

Figure 2: Use Case Diagram for Mobile MS

|  |  |  |
| --- | --- | --- |
| ID | Title | Description |
| 1 | Register User | It is done by unregistered user to access the system. |
| 2 | User Login | User login is needed for user to use system. |
| 3 | Update User | User can update their profile and change password any time. |
| 4 | Remove User | User can remove their account as well. |
| 5 | Add Employee | User can add employee information. |
| 6 | Edit Employee | User can also edit employee information after adding. |
| 7 | Remove Employee | User can remove the employee if they leave. |
| 8 | Add Product | User can store information about product details. |
| 9 | Edit Product | User can also edit product information. |
| 10 | Remove Product | User will be responsible for removing product. |
| 11 | Add Customer | User save customer information once product is sold and billing info also. |
| 12 | Edit Customer | User can update customer information for error input. |
| 13 | Remove Customer | User can remove customer information once securely kept. |
| 14 | View Employee | User can view the details of employee once inserted. |
| 15 | View Product | User can view product to check stock regularly. |
| 16 | View Customer | User can view customer information as well. |
| 17 | Search Report | User can search data on stock, employee, customer for printing or report. |
| 18 | Logout | User can logout of the system anytime. |

# 2.5 Class Diagram

Class Diagram is also the part of analysis that defines the structure of a system. It is known as structural diagram in UML as it shows system’s classes, their attributes, methods and relationship among class. For, creating class diagrams there are other steps that’s need to be completed.

Natural Language Analysis (NLA) is an analysis process to identify the class, attributes and methods of a system. For, performing NLA we need a scenario about the organization or business as shown below.

**Scenario:**

The system is about the mobile shop which manage the product stock, customer sales and employee information. It's a desktop-based application which stores data onto a database located on the same device. The system will be managed or handled by an employee or owner and they are the user or admin of the system. The system has the login system that ensures the security as the employee that has registered can only use it. The registered user can edit and remove their details as well. The user will record the information about an employee, and they are able to edit and delete the data as well. The main task of the system is keeping track of product or stock of the shop, the product and its detail will be added, edited and removed by the user. Other features include creating bills for customers and storing their information as well. User can also edit and remove the customer information. The inserted data can also be viewed by user and they can search for a report as well.

The repeated nouns, verbs and adjectives are refined and below table shows the candidate classes, attributes and operations.

|  |  |  |
| --- | --- | --- |
| **List of Candidate Classes (noun)** | **List of Candidate Attributes (Adjective)** | **List of Candidate operations (Verb)** |
| System, mobile, shop, product, customer, employee, application, database, owner, user, admin, bills, report | stock, sales, task, store, inserted | Manage product, information, edit, remove, record, delete, track, add, create, view, search |

In, the next step the synonyms and out of context words are removed for choosing possible class, attributes and operations.

|  |  |  |
| --- | --- | --- |
| **List of possible class(noun)** | **List of possible attributes (Adjective)** | **List of possible operations (Verb)** |
| User, employee, customer, product, bill | Stock, sales, store | Create, Add, edit, remove, track, view, search |

A screenshot of text

Description automatically generated

Figure 3: Class Diagram for Mobile shop MS

# Chapter 3: Design

Design specifies the structure or behavior of how the system or software functions and how it will be written before writing the complete implementation. The design for the Mobile shop management system project consists of structural, behavioral, database and architectural diagrams which helps in implementation and understanding the systems architecture.

# Structural Design

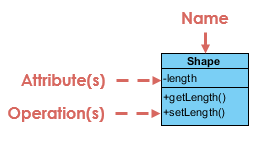
## Class Diagram

Class diagram is a graphical representation of structure of a system which includes classes, attributes and methods or operations and their relationship between different classes or objects. It’s a building block and important diagram for implementation on object-oriented software systems.

The reason for using class diagram are as follows:

* Helps during implementation or makes it easier to identify classes relationship.
* Displays the needed attributes and methods or operations for the system.
* Helps in building system as it can be used as blueprint for final system.
* It describes the static view of the system.

The notations used for the class diagram of mobile shop management system are as follows:

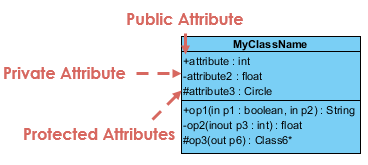


**Class name:** The UML representation of a class is a rectangle containing three compartments stacked vertically and the first compartment include class name.

**Class attribute:** The attribute segment of a class lists each of the class's attributes on a separate line along with attribute types.

**Class operations:** The operations are written at the last compartment and it shows the methods or operation that following class will perform on system.

1. **Class Visibility:**



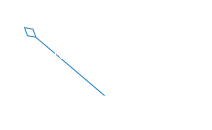
The visibility symbols are +, -, #. **+** represents public types attributes or operations, **-** represents private types attributes or operations and **#** represents protected attributes or operations.

1. **Association:**



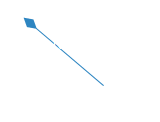
It’s a simple relationship between two classes which is represented by a single solid line or one-sided arrow line.

1. **Aggregation:**



It is a special type of association in which objects are assembled or configured together to create a more complex object.

1. **Composition:**



It represents whole-part relationships and is a form of special type of aggregation where parts are destroyed when whole is destroyed.

Class diagram of Mobile shop management system is shown below:

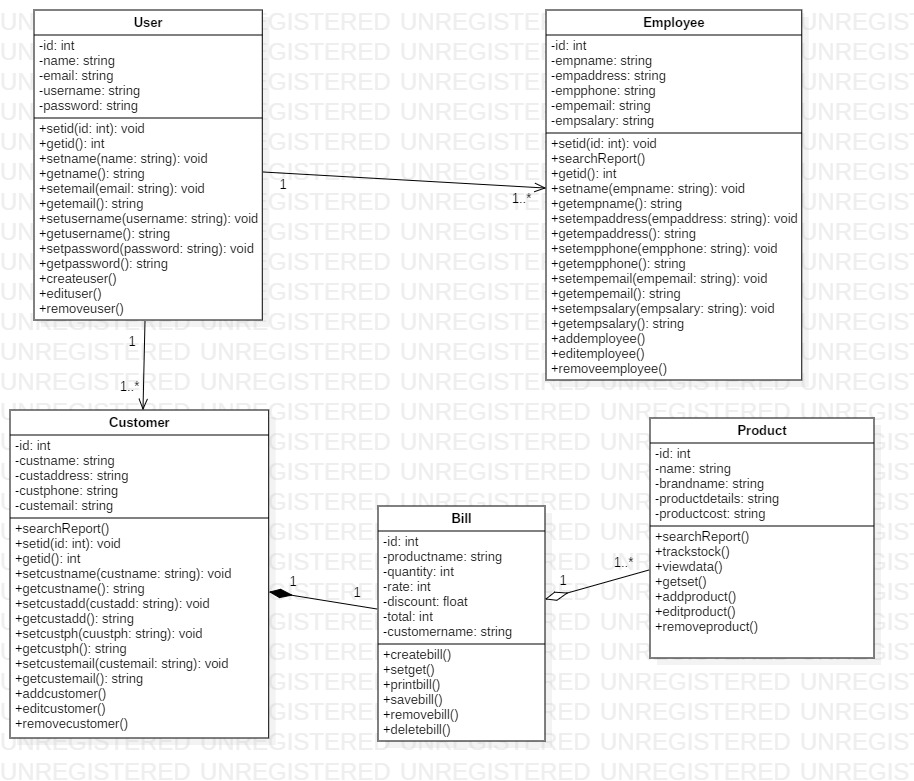


Figure 1: Final Class Diagram

The above diagram shows the final class diagrams of mobile shop management system with proper relation between classes or objects. It shows the association, aggregation and composition relationship along with multiplicity.

## Data Flow diagram (DFD)

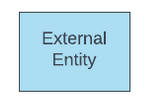
Data flow diagram is a structural representation of a software system. It shows the flow of information about where the data comes from and goes as well as how it is stored. DFD shows how data is handled in a system and defines a difficult explanation through diagrams.

The reason for using data flow diagram are as follows:

* Simple notation and easy to understand data flow of system.
* Shows the scope and boundaries of a system.
* Act as the starting point for redesigning a system.
* Logical information flow of a system.

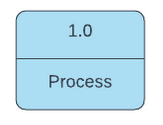
The notations used for the data flow diagram of mobile shop management system are as follows:

1. **External Entity:**



The sources and destination of systems input, and output is external entities. It is object outside the system with which the system communicates.

1. **Process:**

****

A process represents the transformation of incoming data flow into outgoing data flow.

1. **Datastore:**

****

Datastore are repositories of data in the system also known as files sometimes.

Data flow diagram of mobile shop management system is shown below:

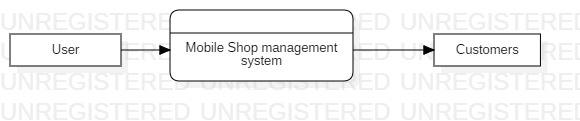


Figure 2: Level 0 Data flow diagram

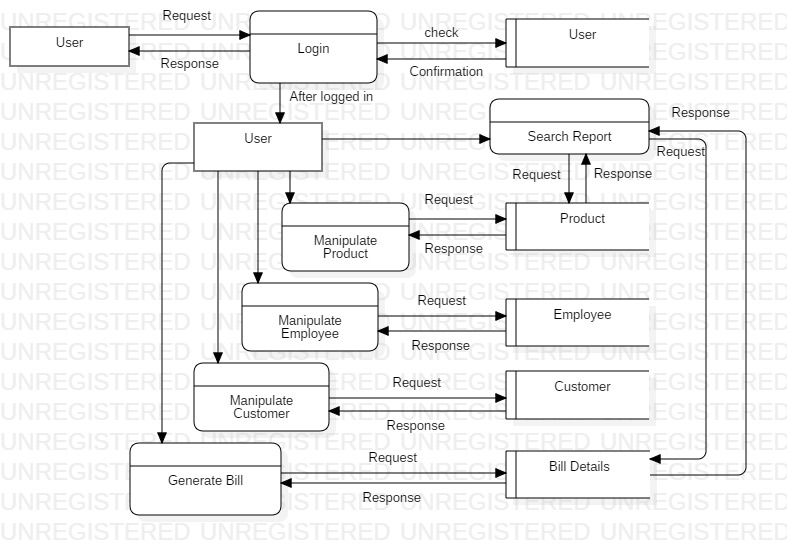


Figure 3: Level 1 Data Flow Diagram

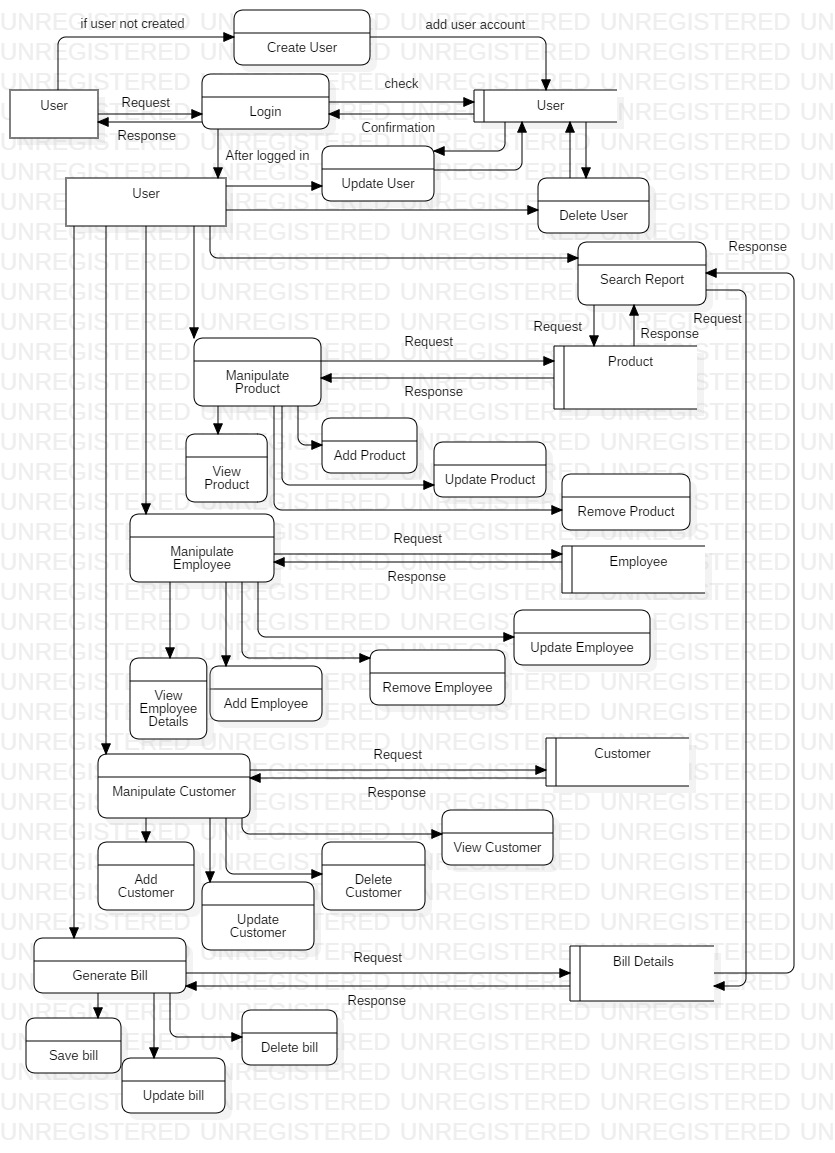


Figure 4: Level 2 Data Flow Diagram

The above data flow diagram shows the full information flow of how data, process and entity interact for the mobile shop management system. The diagrams provide the clear view of how system performs.

# Behavioral Design

## Activity Diagram

Activity diagram is a behavioral diagram that describes the flow of different activities and actions. It shows the series of actions and flow of control in a system like flow chart. Activity diagram can be used to describe every steps of use case and concurrent activities.

The reason for using activity diagram are as follows:

* For high level understanding of the systems functionalities.
* Describe the sequence from one activity to another and draw activities flow of a system.
* Describe the concurrent, branched and parallel flow of the system.
* Mainly used for modelling business requirements.

The notation used for the activity diagram of Mobile shop management system are as follows:

1. Start Symbol:



Represents the start of a process or workflow in an activity diagram.

2. Activity Symbol:



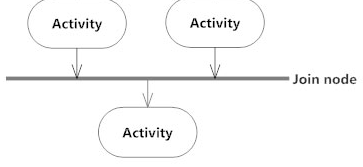
The main component of an activity diagram which indicates activities that make up a modeled process.

3. Connector Symbol:



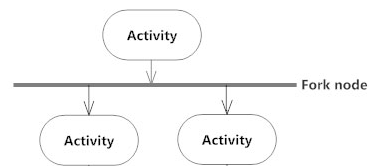
It represents the flow of one activity to another and as well as starts a step of an activity once the step is completed and flow continues with outgoing arrow.

1. Join Symbol:



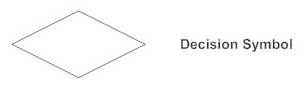
It combines the two concurrent activities and reintroduces them to a flow of one activity.

1. Fork Symbol:



It splits the single flowing activity into two concurrent activities.

1. Decision Symbol:



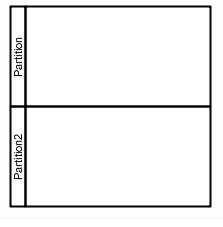
It’s a diamond shape representing the splitting or merging of various flows with the symbol acting as a container or frame.

1. End Symbol:



Represents the completion of a process or stop all control flows or object flows in an activity.

1. Swim lane:



Indicates which objects or actors are responsible for the action and it is separated by a series of line partitioning the diagram.

The activity diagram for the mobile shop management system are shown below:

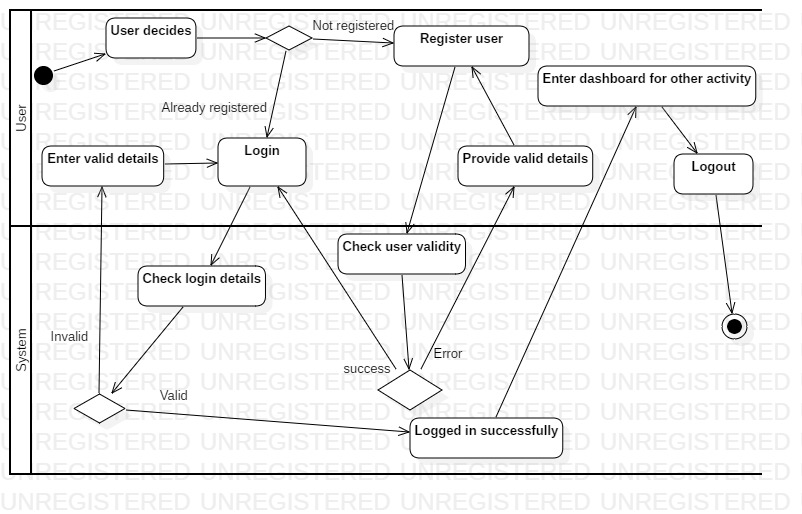


Figure 5: Activity Diagram of User Login

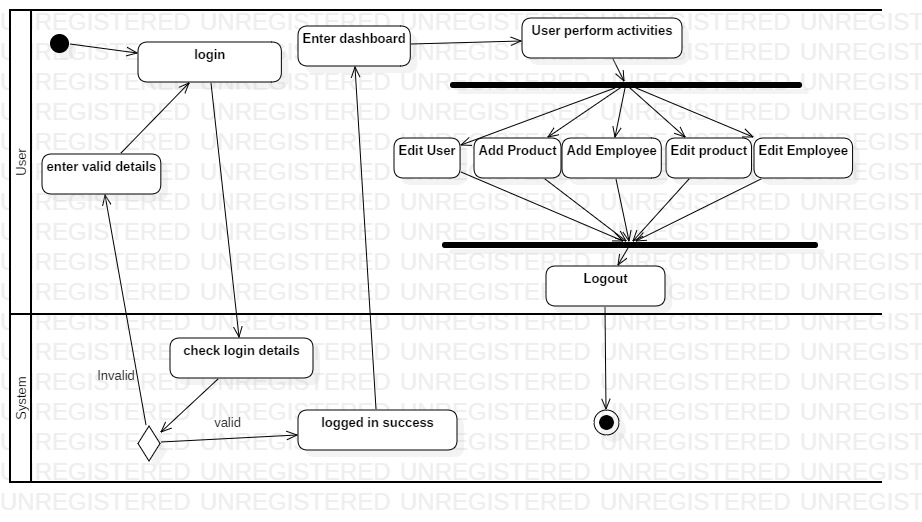


Figure 6: Activity Diagram of User

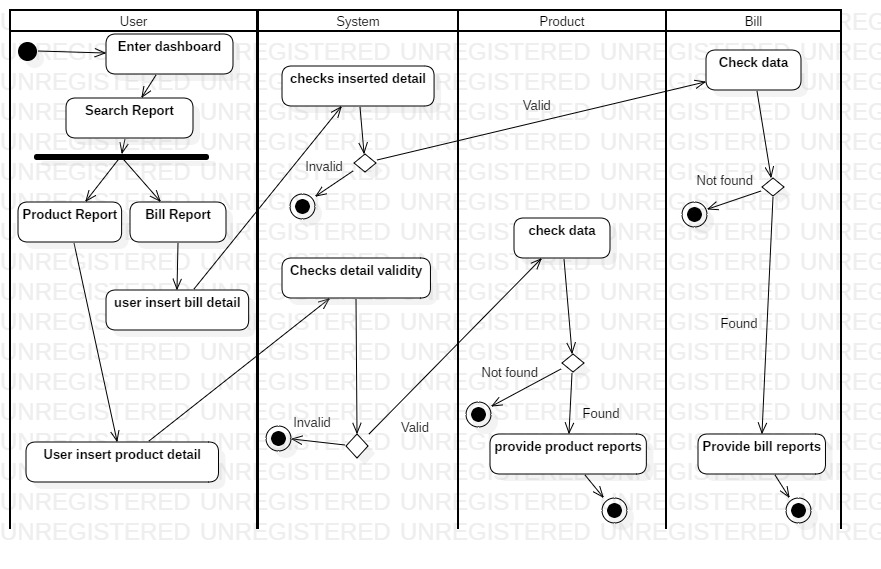


Figure 7: Activity Diagram of Searching Report

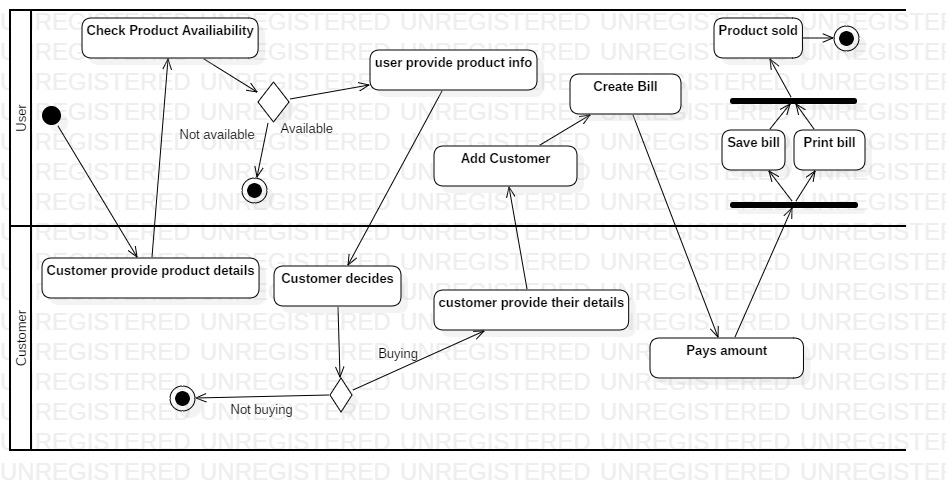


Figure 8: Activity Diagram of Product selling

The above activity diagram shows the different activity performed by user in mobile shop management system such as user login activity, user’s activity and search as well as product selling activity. It describes the use cases from use case diagram.

## Sequence Diagram

Sequence diagram describes the sequence of messages and interactions that happen between objects and actors. It represents a communication sequentially and shows how and in what order the objects in a system function. It portrays the architecture of a system and how components are interconnected.

The reason for using sequence diagram are as follows:

* It shows the details of a UML use case.
* To see how components and objects interact with each other to complete a process.
* Model the logic of a sophisticated function, procedure or operation.
* It shows the order in which methods are invoked and scope or lifetime of objects.

The notation used for sequence diagram of mobile shop management system are as follows:

1. An Actor:



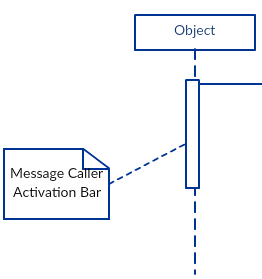
Actors are entities that are both interactive with and external to the system represented by a stick figure.

1. A lifeline:



Dashed vertical line that represents the passage of time and sequential events that occurs to an object during the charted process.

1. Activation bar:



It is the bar or box placed on lifeline and it indicates that an object is active during interaction or shows time needed to object for completing task.

1. Synchronous Message:



Represents or used when sender waits for the receiver to process the message and return before carrying on with another message.

1. Return Message:



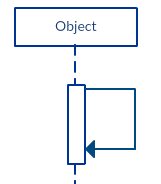
It represents a return message used to indicate that the message receiver is done processing the message and is returning control over to the message caller.

1. Asynchronous message:



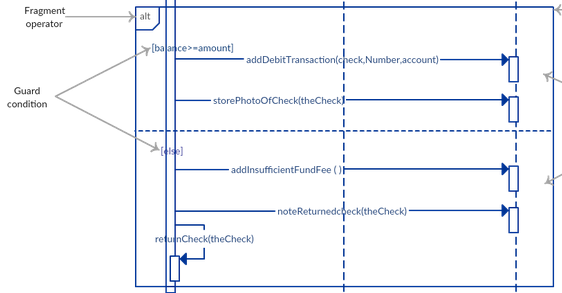
It is used when the message sender does not wait for the receiver to process the message and return before sending other messages to other objects.

1. Reflexive message:



It represents the message sent by an object to itself.

1. Alternatives:



The alternative combination fragment is used when a choice needs to be made between two or more message sequences.

The sequence diagrams of the mobile shop management system are as follows:

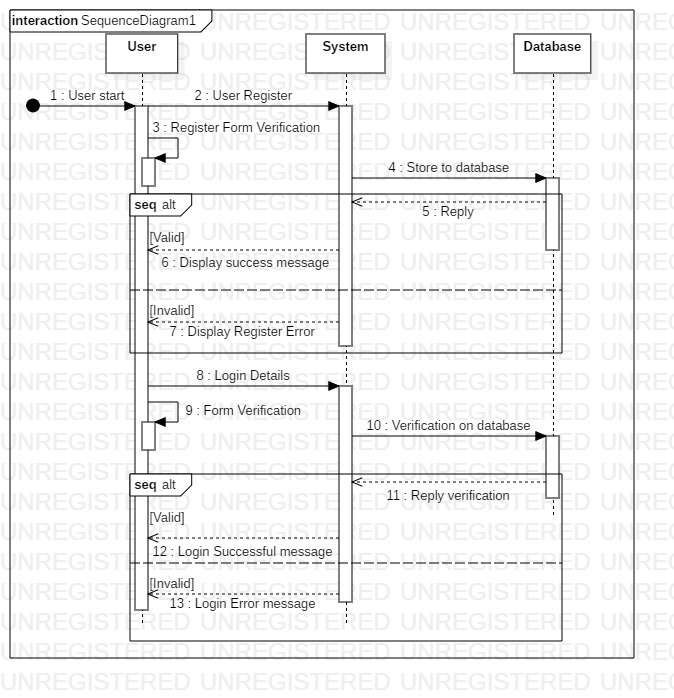


Figure 9: User login sequence diagram

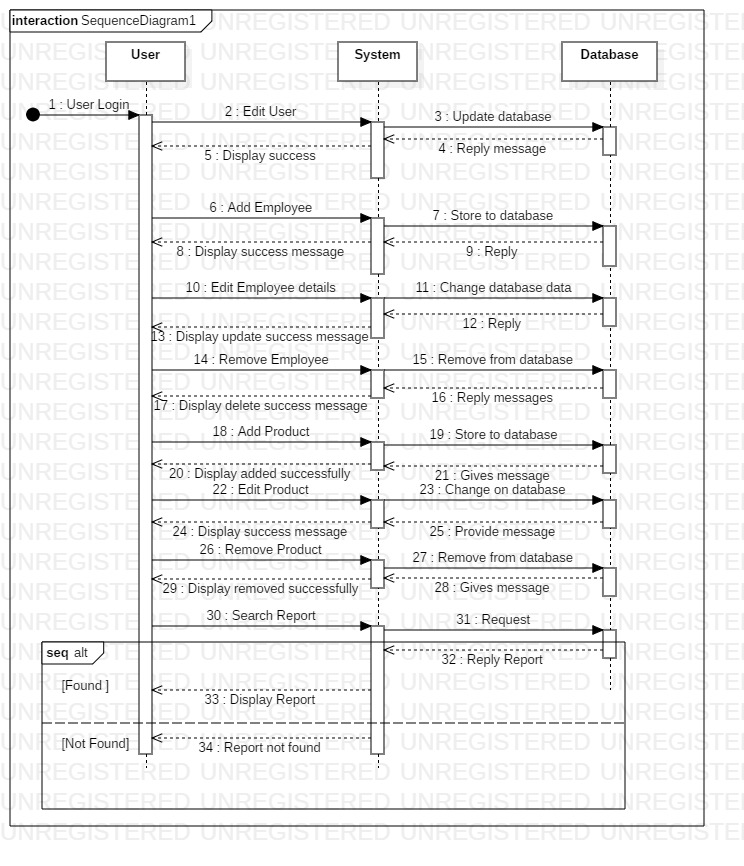


Figure 10: Sequence diagram of User

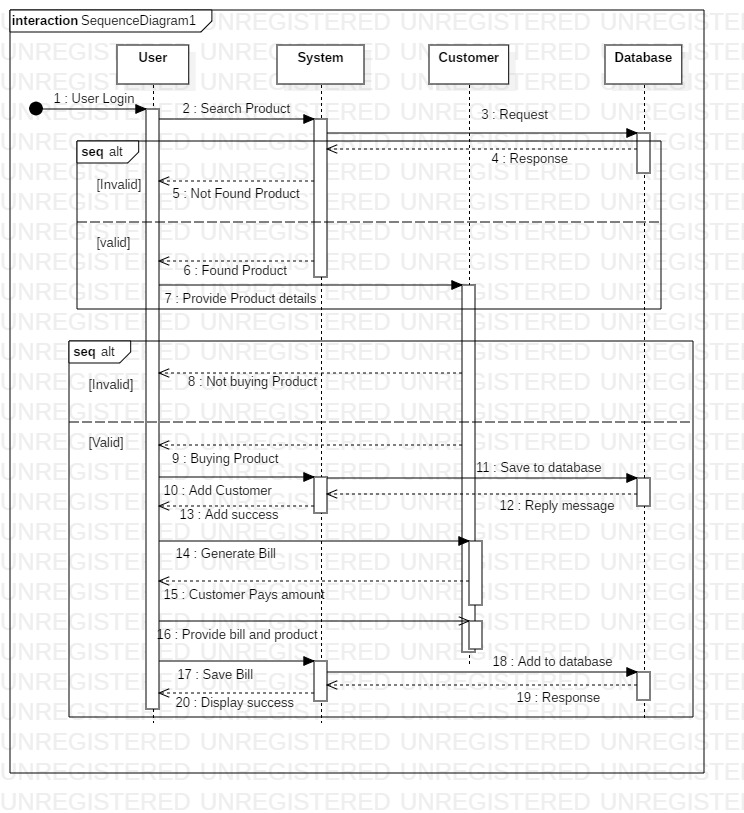


Figure 11: Sequence diagram for Product Selling

Sequence diagram created for mobile shop MS above shows the sequence of messages and interaction that happens between users and other objects of Mobile shop MS. The diagram also shows sequential order of interactions and communication.

# Database

## Data Dictionary

Data dictionary is the table that contains the metadata of the created database. In relational database it shows the metadata such as tables, column, datatype, views and constraints keys and relationship. The data dictionary of Mobile shop management system is shown below.

The reason for using data dictionary are as follows:

* To manage the details in large scale systems.
* To communicate a common meaning for all elements in the system.
* To document features of the system.
* To locate errors an omission in the system.

Data dictionary for the mobile shop management system are as follows:

User Table

A close up of a white background

Description automatically generated

Figure 12: User table data dictionary

Employee Table

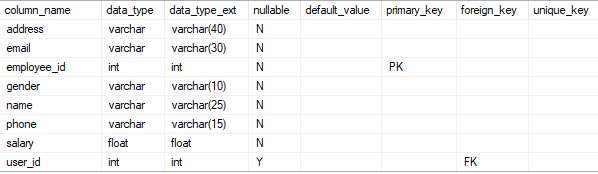


Figure : Employee data dictionary

Customer Table

A screenshot of a cell phone

Description automatically generated

Figure : Customer data dictionary

Product Table

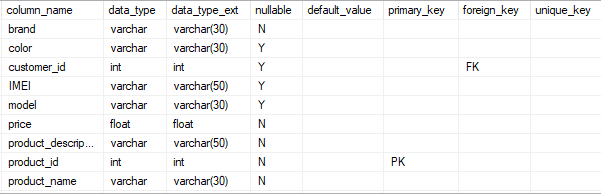


Figure : Product data dictionary

Bill Table

A screenshot of a cell phone

Description automatically generated

## ER Diagram

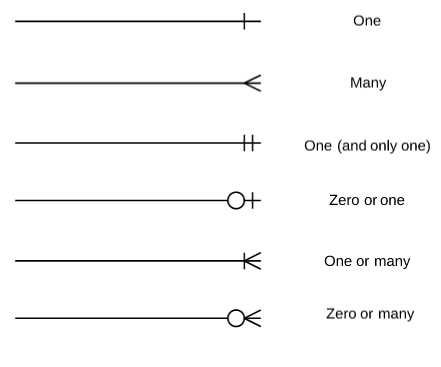
Entity relationship diagram is a high-level conceptual data modelling diagram that shows the relationships of entity set stored in a database. It is based on the notation of real-world entities and the relationship between them. It helps producing a well-designed database and it is created before implementing database.

The reason for using ER diagram are as follows:

* Describe the entities, attributes and relationships.
* ER diagrams are translatable into relational tables which helps in building databases quickly.
* It can be used by database designer as blueprint for implementing data in specific software applications.
* It provides a preview of how each table should connect and defines constraint keys and relationship.

The crow’s feet notation of ER diagram are as follows:

1. In crow’s feet ER diagram, the notation is cardinality and ordinality as shown below for entities relationship.



ER Diagram for the mobile shop management system is shown below:

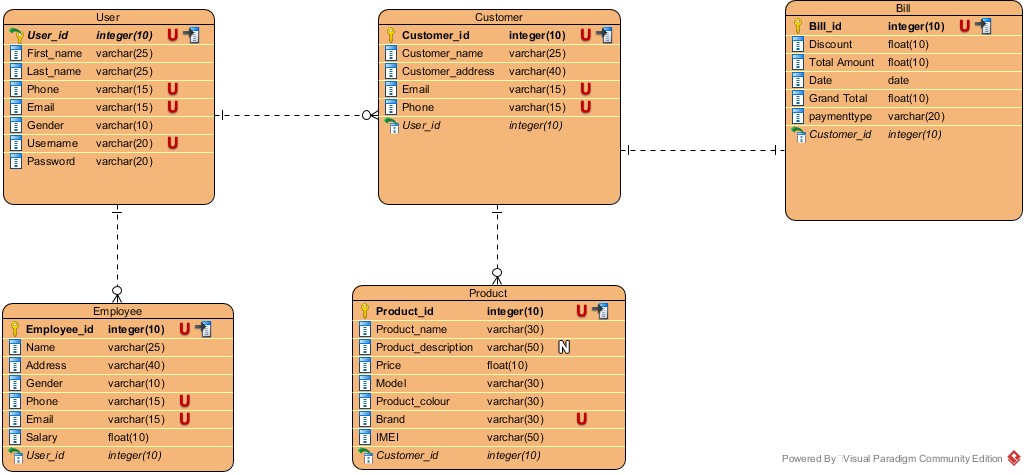
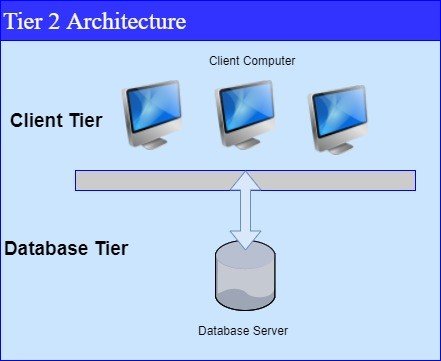


Figure : ER diagram of Mobile shop MS

# Architecture



System architecture is the structural design of systems and it is a conceptual model. System architecture serves as the blueprint for both the system and the project developing it, defining the work assignments that must be carried out by design and implementation teams. Software or system architecture consist of different tiers or layer. Since, the system I am developing is desktop-based application with server I will be using 2-tier architecture. It is divided into two parts client and database. Client system handles both Presentation and Application layers and Server system handles Database layer. The communication takes place between the Client and the Server. Client system sends the request to the Server system and the Server system processes the request and sends back the data to the Client System. It is also known as client server application.

# UI

## Prototype

Prototype is an initial rough draft of the final product. It is essential in product development as it gives rough idea to the creator how the product will be, what it will do, how it operates and can make alteration while the product is still in concept mode.

The reason for using prototype are as follows:

* It gives a proper idea of how the system layout or interface should be.
* It is easier to develop and refine the model at any time.
* It can be changed at any time from design to function since it’s a concept mode.
* It helps finding faults before real product is produced.

The digital prototype of the Mobile shop management system are as follows:

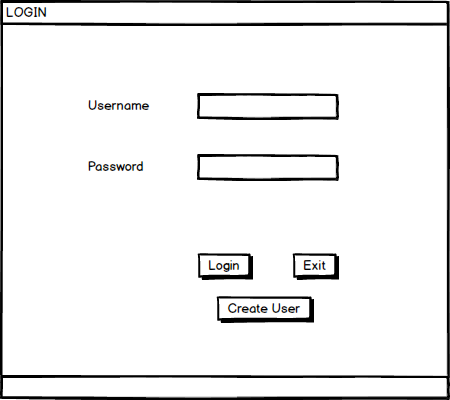


Figure : Login page prototype

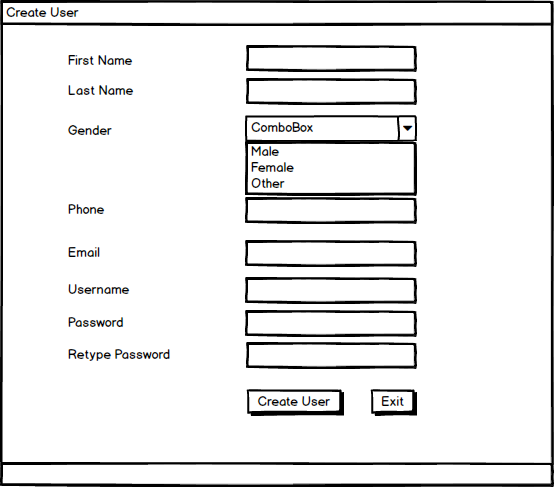


Figure : Create page prototype

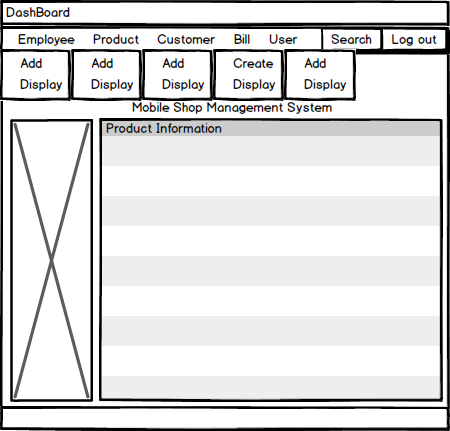


Figure : Dashboard prototype

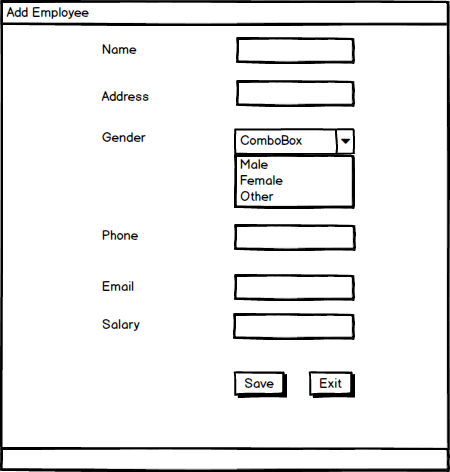


Figure : Employee add page

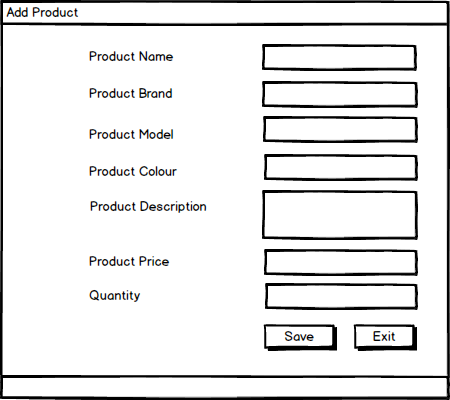


Figure : product add page

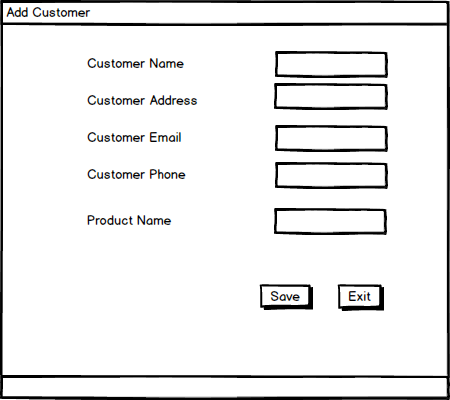


Figure : customer add page

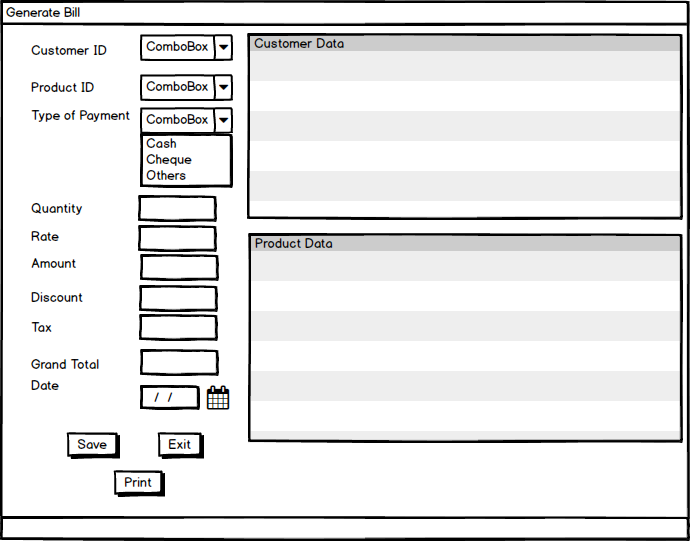


Figure : generate bill page

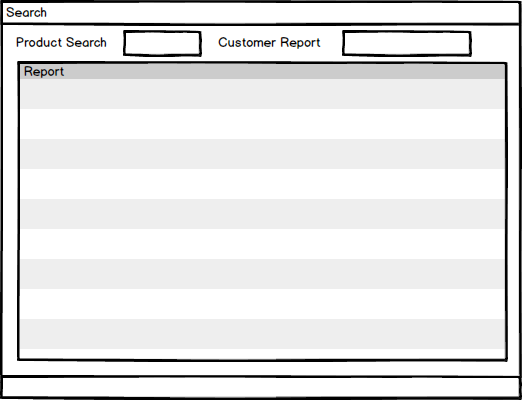


Figure : Search report page

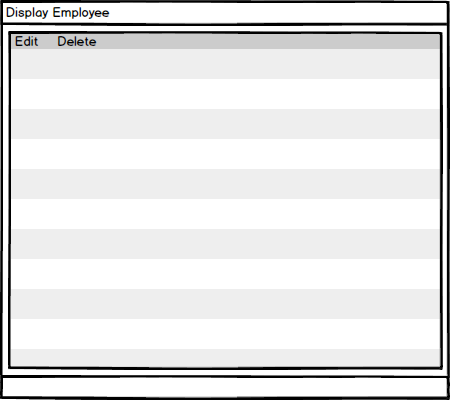


Figure : Display and edit page

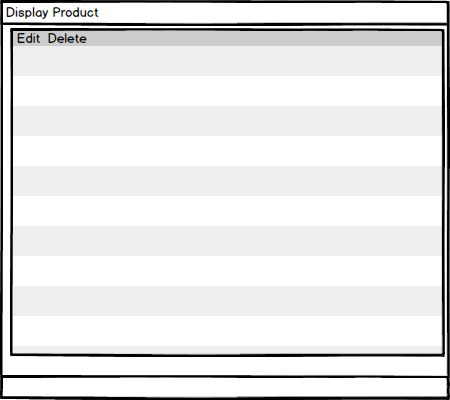


Figure : display and edit page

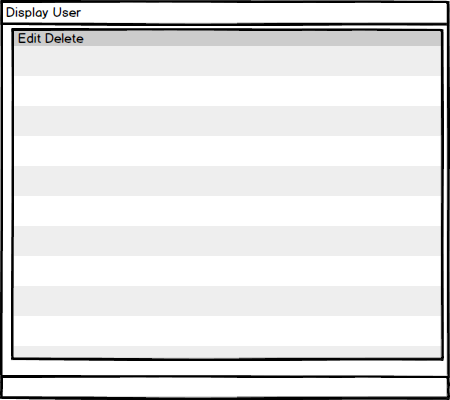


Figure : display and edit page

# Chapter 5: Testing

# Introduction

Testing is an important steps of software development process to verify and validate the correctness, quality and performance of the software. It is done for making sure that the code functions properly and expected results are producing by the software. It is an activity to ensure that the software is defect or error free. It includes execution of a system or software components to evaluate one or more properties of interest. Testing can be done automatically through tools or manually.

# Importance

Testing of software is important because failure in software can cause human and monetary loss as well and there are lots of examples of such things caused by a bugs or error in software. The importance of using testing in software development process are as follows:

* Any error or defects made during the development phase can be identified through testing.
* Testing increases products success as it helps in increasing quality and customer reliability on that software.
* Testing ensure that a high-quality product is delivered as lots of expenses is spend on product development.
* A well-tested product causes less maintenance cost and so the results delivered are more accurate, consistent and reliable.
* Testing ensures that an application does not results any failure in early days of products life cycle.

# Types of Testing

There are different basics as well as essential software testing techniques that should be followed before deploying software to client. Some of the testing are as follows:

1. Unit Testing
2. Black Box Testing
3. White Box Testing
4. Grey Box Testing
5. Integration Testing
6. Beta Testing
7. Alpha Testing
8. Performance Testing

From the above different types of testing, I have chosen two of them for this project. They are:

**Unit Testing**

Unit Testing focuses on the small unit of a project. It is done by a developer using sample inputs and observing its corresponding outputs. Individual units or inter-related units are tested in this testing.

**Test Name: Check User Register**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 1 | Insert User details and check it works or not | User Details | User should be provided message box for success | Message box showed without any error | Pass |

A screenshot of a computer screen

Description automatically generated

Figure : Register User Test

A screenshot of a cell phone

Description automatically generated

Figure : Register Test Passed

**Test Name: Check Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 2 | Insert login details and check it works or not | Login Details | User should be directed to dashboard | Dashboard opened without any error | Pass |

A screen shot of a computer

Description automatically generated

Figure : Login Unit Test Passed

**Test Name: Check Employee Insert**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 3 | Insert Employee details and check it works or not | Details of an employee | Message box should be shown about employee updated successfully | Message box shown without any error | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Insert Employee Test

Screen of a cell phone

Description automatically generated

Figure : Insert Employee Test Passed

**Test Name: Check Product Insert**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 4 | Insert details of product and check it works or not | Product Details | Product updated successfully message box should be shown | Message box shown without any error | Pass |

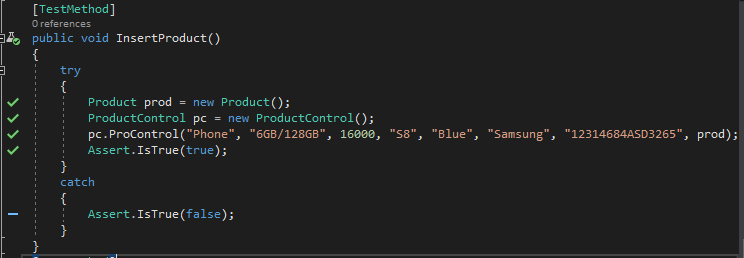


Figure :Insert Product Test

A screenshot of a cell phone screen with text

Description automatically generated

Figure : Insert Product Test Passed

**Test Name: Check Customer Insert**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 5 | Insert Customer details and check it works or not | Customer Details | Customer details updated successfully dialog box | Message box shown without any error | Pass |

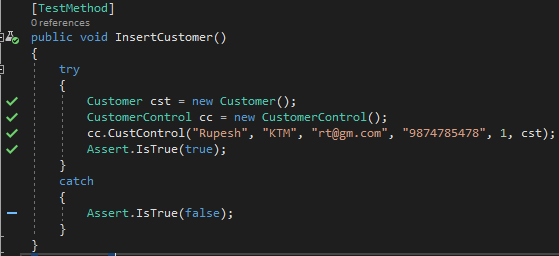


Figure : Insert Customer Test

A screenshot of a cell phone

Description automatically generated

Figure : Insert Customer Test Passed

**Test Name: Check Bill Insert**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 6 | Insert Bill details and check it works or not | Bill Details | Bill details added success message box | Message box shown without any error | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Bill Insert Test

A screen shot of a smart phone

Description automatically generated

Figure : Bill Insert Test Passed

**Test Name: Check Username**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 7 | Insert username details and check for duplicate in database | username | Message box should not be shown | No message box shown, and no error occurred | Pass |

A screenshot of a video game

Description automatically generated

Figure : Check Duplicate Username Test

A screenshot of a cell phone

Description automatically generated

Figure : Check Username Test Passed

**Test Name: Check Email**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 8 | Insert Email and check for its validity | Valid Email | Message box should not be shown | No message box shown, and no error occurred | Pass |

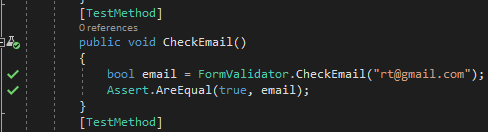


Figure : Check Valid Email Test

A screenshot of a cell phone

Description automatically generated

Figure : Check Email Test Passed

**Test Name: Check Password**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 9 | Insert Password and check for its length | Greater than Six length password | Message box should not be shown | No message box shown, and no error occurred | Pass |

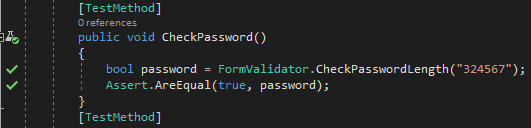


Figure : Check Password Length Test

A screen shot of a smart phone

Description automatically generated

Figure : Check Password Test Passed

**Test Name: Check Two Password**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 10 | Insert two Password and compare for same detail | Greater than Six length password | Message box should not be shown | No message box shown, and no error occurred | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Check Two Password Test

A screenshot of a cell phone

Description automatically generated

Figure : Compare Two Password Tests Passed

**Test Name: Check Phone Length**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 11 | Insert phone and check length | Greater than five length phone number | Message box should not be shown | No message box shown, and no error occurred | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Check Phone Length Test

A screen shot of a smart phone

Description automatically generated

Figure : Check Phone Length Test Passed

**Black Box Testing**

It is a behavioral testing in which the internal structure or item being tested is not known to the tester. Black box testing just focus on inputs and outputs of the software system without focusing about implementation code or internal knowledge of software program.

**Test Name: User Registration Test**

**Pre-Condition: User Should be registered in the system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 1 | To check if the detail is inserted or not | First Name: Rupesh  Last Name: Thapa  Phone: 9876543210  Gender: Male  Email: [rt@gmail.com](mailto:rt@gmail.com)  Username: rupesh  Password: 123456 | Success message box should be shown to user | Success Message box shown, and data saved in database | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : User Data Added

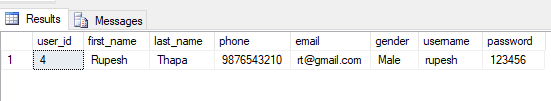


Figure : User Added in database

**Test Name: Login Test**

**Pre-Condition: User Should be directed to dashboard**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 2 | To check details inserted is correct and login working or not | Username: rupesh  Password: 123456 | User should be directed to the dashboard | User is provided access with dashboard | Pass |

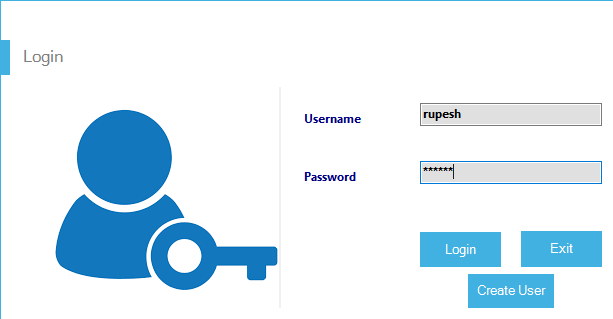


Figure : Login Detail filled

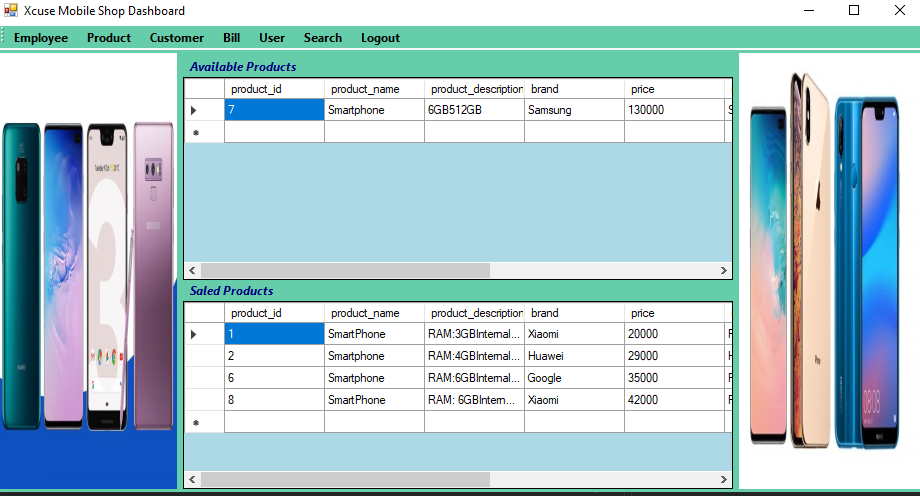


Figure : Login success and Dashboard Shown

**Test Name: Employee Insert Test**

**Pre-Condition: Employee information should be added in system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 3 | To check if employee detail is inserted or not | Name: John  Address: Pokhara  Gender: Male  Phone: 9874561230  Email: [john@gmail.com](mailto:john@gmail.com)  Salary: 18000 | Success message box should be shown to user | Success Message box shown, and data saved in database | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Employee Details Added

A screenshot of a cell phone

Description automatically generated

Figure : Employee Details in Database

**Test Name: Product Insert Test**

**Pre-Condition: Product information should be added in system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 4 | To check if product detail is inserted or not | Product Name: Smartphone  Description: Ram 6GB/ Internal 32GB  Model: Galaxy A70  Brand: Samsung  Color: Navy Blue  Price: 28990  IMEI:978654789547851 | Success message box should be shown to user | Success Message box shown, and data saved in database | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Product Data Added

A screenshot of a cell phone

Description automatically generated

Figure : Product added In Database

**Test Name: Customer Insert Test**

**Pre-Condition: Customer information should be added in system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 5 | To check if customer detail is inserted or not | Name: Kabish  Address: Gongabu  Email: [kb@gmail.com](mailto:kb@gmail.com)  Phone: 9871234785 | Success message box should be shown to user | Success Message box shown, and data saved in database | Pass |

A screenshot of a social media post

Description automatically generated

Figure : Customer Details Added

A screenshot of a cell phone

Description automatically generated

Figure : Customer Data in Database

**Test Name: Product Sale Test**

**Pre-Condition: Product information should be updated in system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 6 | To check if product is assigned to customer successfully by updating product data | Customer Name: Kabish  Product Model: Galaxy A70 | Product details should be updated, and customer should be assigned | Product assigned to customer successfully and customer added in product table | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Product Sale to Customer

A screenshot of a cell phone

Description automatically generated

Figure : Product details updated Customer added

**Test Name: Bill Generate Test**

**Pre-Condition: Bill information should be added in system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 7 | To check if bill detail is inserted | Customer Name: Kabish  Product: Galaxy A70  Quantity:1  Payment Type: Cash  Discount: 1000  Paid:30000 | Success message box should be shown to user | Success Message box shown, and data saved in database | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Bill Generated

A screenshot of a cell phone

Description automatically generated

Figure : Bill Details Added in Database

**Test Name: Search Customer Test**

**Pre-Condition: Customer information should be showed by system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 8 | To Check if data will be shown or not | Customer Name: Kabish | Customer data should be shown, in data grid view | Customer data shown, with success | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Customer Data Searched

**Test Name: Product Update Test**

**Pre-Condition: Product information should be updated in system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 9 | To check if product detail is updated | Update data  Of model: Galaxy A70 set price to 25000 | Success message box should be shown to user and data should be updated | Success Message box shown, and data updated in database | Pass |

A screenshot of a cell phone

Description automatically generated

Figure : Product Data Updated

A screenshot of a cell phone

Description automatically generated

Figure : Product updated in Database

**Test Name: Customer Update Test**

**Pre-Condition: Customer information should be updated in system**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Scenario** | **Test Input Data** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** |
| 10 | To check if customer detail is updated | Update Name: Kabish set  Address: KTM  Phone: 014325654 | Success message box should be shown to user and data should be updated | Success Message box shown, and data updated in database | Pass |

A screenshot of a social media post

Description automatically generated

Figure : Customer Details Updated

A screenshot of a cell phone

Description automatically generated

Figure : Customer Details Updated in Database

## 

# 

# Chapter 6: Other Project Issues

# 6.1 Risk Management

Risk are inevitable during project whether its big or small. Risk are problems that arise during software development. Similarly, Risk Management refers to the process of identifying the risks, analyzing it and planning to act upon it. The possible risks were identified in proposal and managed, but the below list shows the risk that arrived during project.

The risk impact is calculated through likelihood and consequences and solution is shown in table:

**Impact = Likelihood \* Consequence**

Risk Likelihood values are shown as follows

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Risk Consequence values are shown below

|  |  |
| --- | --- |
| **Consequence** | **Value** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

Risk Consequences values are shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Risks** | **Likelihood** | **Consequences** | **Impact** | **Solution** |
| 1 | Data Loss | 2 | 4 | 8 | Data should be stored in multiple place such as google drive, one drive e.t.c |
| 2 | Equipment Failure | 1 | 4 | 4 | Data should be backed up on cloud and project can continue from different device. |
| 3 | Insufficient Research material | 1 | 3 | 3 | Resources should be prepared before starting project. |
| 4 | Resource (Internet) Failure | 1 | 3 | 3 | Maintenance should be done regularly. |

## 6.2 Configuration Management

Configuration Management refers to the process of managing all the files, project and assets in a structured way so, that it can be accessed easily any time. It includes all functional and physical specifications. There are five steps to configuration management such as planning, identification, control, status accounting and audit. Any change made on project are assessed and monitored. The files related to project are kept under one single folder in a structured way. Project files such as Proposal, Analysis, Design, Implementation, Testing, Final Document are kept in CP Project.