Introduction

In any business or organization, all functions are interlinked and connected to each other and are often overlapping. Some key aspects like supply chain management, logistics and inventory form the backbone of the business delivery function. Therefore these functions are extremely important to marketing managers as well as finance controllers.

Inventory management is a very important function that determines the health of the supply chain as well as the impacts the financial health of the balance - sheet. Every organization constantly strives to maintain optimum inventory to be able to meet its requirements and avoid over or under inventory that can impact the financial figures.

Inventory is always dynamic. Inventory management requires constant and careful evaluation of external and internal factors and control through planning and review. Most of the organizations have a separate department or job function called inventory planners who continuously monitor, control and review inventory and interface with production, procurement and finance departments.

Defining Inventory

Inventory is an idle stock of physical goods that contain economic value, and are held in various forms by an organization in its custody awaiting packing, processing, transformation, use or sale in a future point of time.

Any organization which is into production, trading, sale and service of a product will necessarily hold stock of various physical resources to aid in future consumption and sale. While inventory is a necessary evil of any such business, it may be noted that the organizations hold inventories for various reasons, which include speculative purposes, functional purposes, physical necessities etc.

From the above definition the following points stand out with reference to inventory:

- All organizations engaged in production or sale of products hold inventory in one form or other.
- Inventory can be in complete state or incomplete state.
- Inventory is held to facilitate future consumption, sale or further processing/value addition.
- All inventoried resources have economic value and can be considered as assets of the organization

Limitations Of Existing Systems:

-Need of extra manual efforts.

-Do not provide one-stop solution.

-Hard to retrieve relative information at a time.

-Complex with regards to user interaction.

-Certain Reports are difficult to generate.

Need Of System

As daily lots of transactions are being undertaken, records are needed to be maintained for their future reference. Maintaining such huge records in registers would lead to a much tedious and could cause a lot of time consumption.

The management operator would also need to perform certain decision-making based on the analysis made on the sales and purchased. This becomes impossible without having a proper idea of the daily transactions within the organization. Thus maintaining and analyzing these huge records within the registers could cause a lot of inconvenience.

The operator requires that the stock should be deducted when a sale is made and the stock should be added when a purchase is made from the dealer. The automated system is required to perform updation within the stock as and when a new sale or purchase is made.

The automated system would help the organization in providing a better analysis of the daily transaction, rather than making a view over the entire daily transaction thus helping the organization in making better decisions without doing much work. These all transactions cannot be done properly without the help of a complete automated system. Thus there is an ample need of an automated system to carry out these transactions effectively and smoothly and to reduce the daily workload.

The following compelling facts will give the missing dimension perhaps worth giving attention to :

- The overall volume of Fitness center has grown exponentially. The dealers and the customers they have been dealing have grown in numbers. Day by day it is becoming increasingly difficult to manage it with the current method of running business.
- The use of computers, printers enabled software will only simplify their business needs and will help them sustain their current leadership status.

Feasibility Study

Technical Feasibility :-

The minimum hardware requirement to implement the system is Pentium IV processor with 512 MB RAM. The whole system is being developed on Visual Studio 2015 Professional IDE for GUI (Front End) and uses Microsoft SQL Management Studio 2014 for Database. These are the softwares which are very popular and easily available in market.

Economical Feasibility :-

It is cost effective system. No extra efforts are needed to train for using the system. This system is quite beneficial with respect to its cost. The user shall be conversant with computer technology. The system is user friendly and thus easy to use and operate.

Operational Feasibility :-

This system will reduce the workload and loss of vital information. It is time saving and will aid the employee administration to be system generated rather than person dependent. Data retrieval and data presenting will be done by the system so the report generation will be automated.

Technology Review

ASP.Net Framework is chosen for development of Inventory Management System with Visual Studio Professional 2015 (IDE) and Visual C# as core Programming language and SQL Server.

Framework :-

ASP.Net framework is a revolutionary platform that allows to build the following types of applications − Windows applications Web applications Web services. The .Net framework applications are multi-platform applications. The framework has been designed in such a way that it can be used from any of the following languages: C#, C++, Visual Basic, Jscript, COBOL, etc. All these languages can access the framework as well as communicate with each other. The .Net framework consists of an enormous library of codes used by the client languages such as C#. Following two are important components of the .Net framework − Common Language Runtime (CLR) The .Net Framework Class Library Other components are: Common Language Specification, Common Type System, Metadata and Assemblies, Windows Forms, ASP.Net and ASP.Net AJAX, ADO.Ne etc.

Common Language Runtime (CLR) :

Common Language Runtime(CLR) is the basic and Virtual Machine component of the .NET Framework. It is the run-time environment in the .NET Framework that runs the codes and helps in making the development process easier by providing the various services. Basically, it is responsible for managing the execution of .NET programs regardless of any .NET programming language. Internally, CLR implements the VES(Virtual Execution System) which is defined in the Microsoft's implementation of the CLI(Common Language Infrastructure).

The code that runs under the Common Language Runtime is termed as the Managed Code as CLR provides a managed execution environment for the .NET programs by improving the security, including the cross language integration and a rich set of class libraries etc.

Framework Class Library (FCL) :

It is the collection of reusable, object- oriented class libraries and methods etc that can be integrated with CLR. Also called the Assemblies. It is just like the header files in C/C++. Installing framework basically is the installation of CLR and FCL into the system.

Following are the commonly used namespaces that contains useful classes and interfaces and defined in Framework Class Library.

Following are the commonly used namespaces that contains useful classes and interfaces and defined in Framework Class Library:

1. System: It includes all common data-types, string values, arrays and methods for data conversion.
2. System.Data, System.Data.SqlClient: These are used to access a database, perform commands on a database and retrieve database.
3. System.Windows.Forms, System.Windows.Forms.Design: These namespaces are used to create Windows-based applications using Windows user interface components.

Integrated Development Environment :-

For the project, Microsoft Visual Studio is used as a platform for system development. Microsoft Visual Studio is an Integrated Development Environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms etc. Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,C++, C++/CLI, Visual Basic .NET, C#, F#,JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins.

Visual Studio 2015 Professional version has been used for developing INVENTORY. 2015 Pro version of Visual Studio has following added key features:
Syntax Highlighting Visual Studio IntelliSense
Error in Visual Studio 2015 Code Fixes and Refactoring
Visual C# language has been used for the purpose.

Programming Language

C# is one of the programming languages designed for the Common Language Infrastructure. C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg. It is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages on different computer platforms and architectures. The language is chosen for the project because of following reasons:

It is a modern, general-purpose programming language. It is a part of .Net Framework.
It is object oriented and component oriented. It is structured and easy to learn.
It can be compiled on a variety of computer platforms.
It provides Automatic Garbage Collection and Standard Library. It supports Properties and Events.
It also features Delegates and Events Management.

Database :-

SQL Server Management Studio 2014 Express version is used to design database for the proposed Employee Management System. SQL language and Relational DataBase Management Systems(RDBMS) are used to set the foundation of system's back end.
RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, Oracle, MySQL, and Microsoft Access.
The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

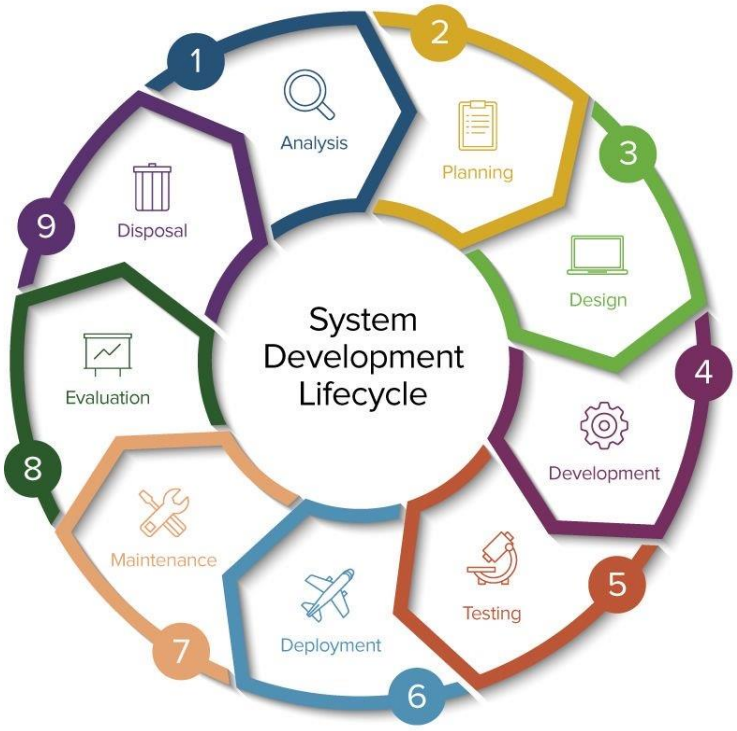
Structured **Q**uery **L**anguage or **SQL** is a standard Database language which is used to create, maintain and retrieve the relational database. It is particularly used to work with structured data where there are relations associated within the data itself.
Programmers embed SQL commands into their application programs to access the data. SQL is a client/server language. Programs use SQL to communicate over a network with database servers that store shared data.
SQL being distributed database language, Distributed Database Management Systems use SQL to distribute data across multiple connected computer systems.
In the proposed system, SQL serves as the link between "front-end" computer systems optimized for user interaction and "back-end" systems specialized for database management.

What is SDLC ?

✓ Software Development Life Cycle :-

- It offers the basis for project planning, scheduling, and estimating.
- It is a mechanism for project tracking and control.
- Enhance development speed.
- Improved client relations.
- Helps you to decrease project risk.

The following figure is a graphical representation of the various stages of a typical SDLC.



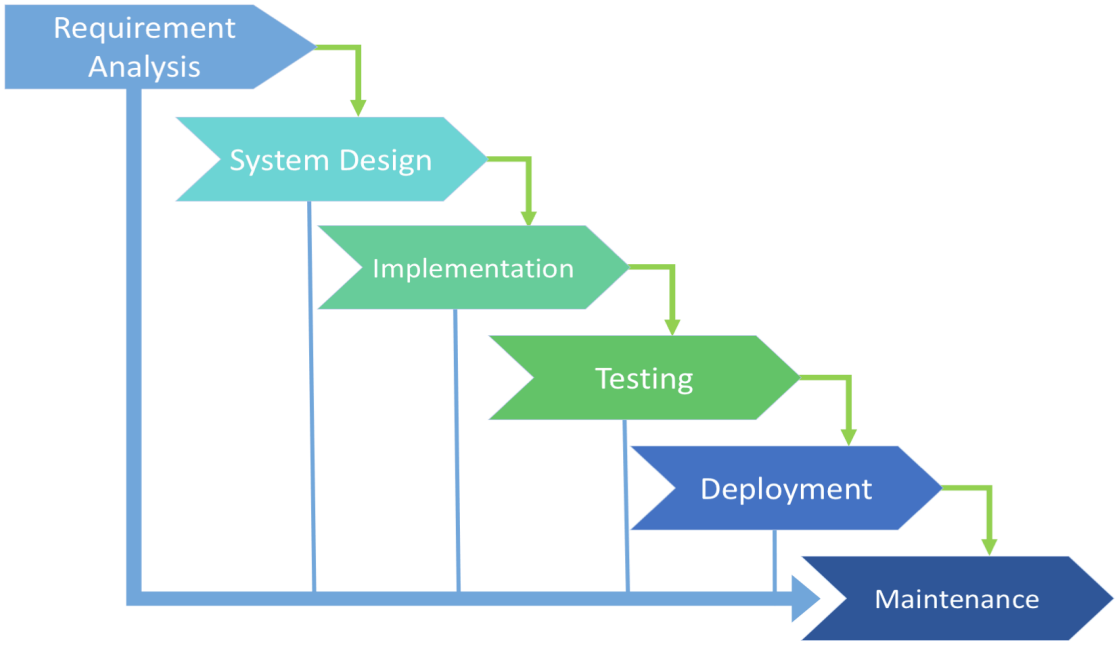
SOFTWARE DEVELOPMENT LIFE CYCLE

SDLC MODELS :-

- ❖ Waterfall Model.
- ❖ Iterative & Incremental Model.
- ❖ Spiral Model.
- ❖ V-Shaped Model.
- ❖ Agile Model.

Waterfall Model

- It works in a linear sequential flow project.
- In which progress is seen as flowing steadily downwards.
- Previous stage should be completed before going to another.



Advantages & Disadvntages :-

Advantage :-

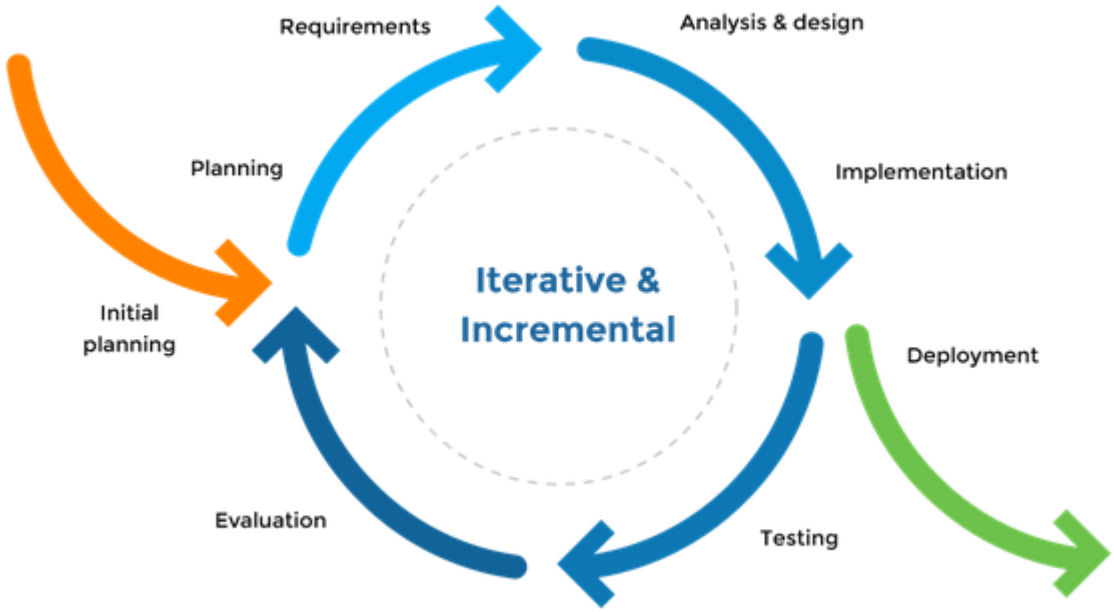
- Easy to explain to the users.
- Stages & activities are well defined.
- Each Phase has specific deliverables.

Disadvantage :-

- Very difficult to go back to any stage after reaching the end.
- Costly and required more time in addition to detailed plan.

Iterative & Incremental Model

- A system through repeated cycles (iterative) and in smaller portions at a time (incremental).
- Allowing software developers to take advantage of what was learned during the development of earlier parts.



Advantages & Disadvantages :-

Advantage :-

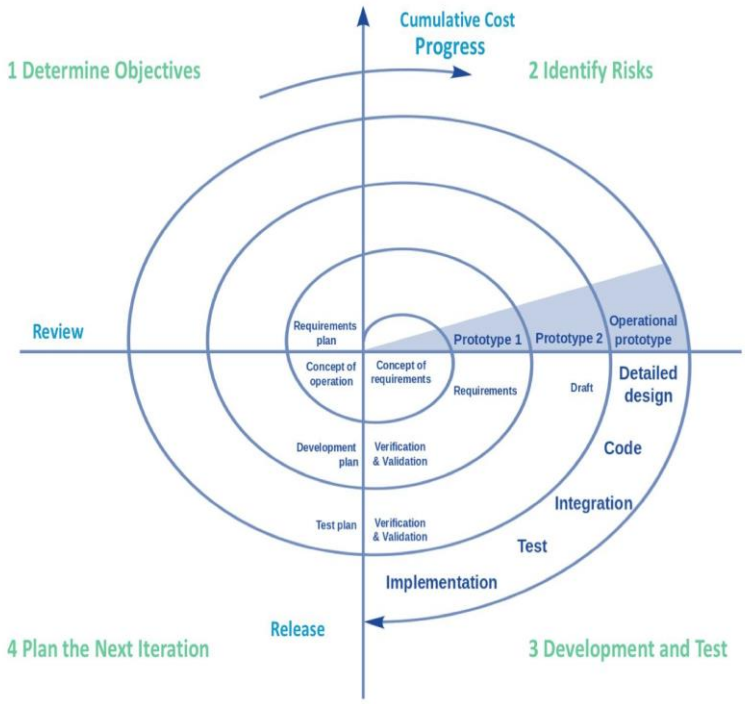
- The shorter the iteration, the easier the testing and debuggingstages are.
- The progress is easy to measure.

Disadvantage :-

- Iterative Model requires more resources than waterfall model.
- Constant management is required.
- Bad choice for small projects.

Spiral Model

- This model of development combines the features of the prototyping model and the waterfall model.
- The spiral model is favored for large, expensive, and complicated projects.



Advantages & Disadvantages :-

Advantage :-

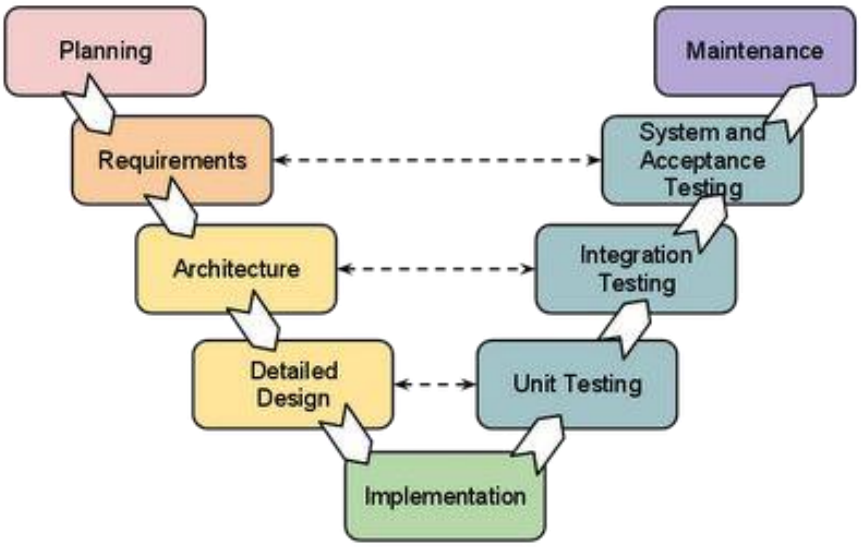
- Issues are identified during the process.
- Early involvement of developers.

Disadvantage :-

- Time consuming.

V-Shaped Model

- It is an extension of the waterfall model, Instead of moving down in a linear way, the process steps are bent upwards after the implementation and coding phase, to form the typical V shape.
- The major difference between the V-shaped model and waterfall model is the early test planning in the V-shaped model.



Advantages & Disadvantages :-

Advantage :-

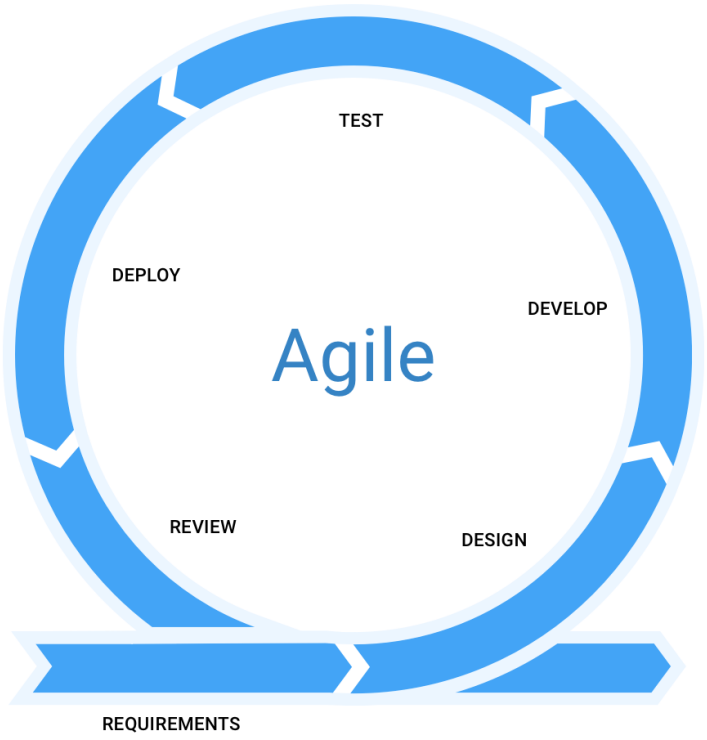
- Testing & verification takes place in early stages.
- Good for small projects where the requirements are static & clear.

Disadvantage :-

- Lack of flexibility .
- Adjusting scope can be expensive.

Agile Model

- In the agile methodology after every development iteration, the customer is able to see the result and understand if he is satisfied with it or not.



Advantages & Disadvantages :-

Advantage :-

- Project is divided into short and transparent iterations.
- Risks are minimized due to flexible change process.

Disadvantage :-

- Need special skills for the team.
- Reduces usability of the components.

Data Dictionary

FILED NAME	TABLE NAME	DATATPE
Id	Tbl_User Tbl_Categories Tbl_Deal_cust Tbl_Product Tbl_Transaction_Details Tbl_Transaction	Int
First_Name	Tbl_User	Varchar (50)
Last_Name	Tbl_User	Varchar (50)
Email_Id	Tbl_User Tbl_Deal_cust	Varchar (50)
UserName	Tbl_User	Varchar (50)
Password	Tbl_User	Varchar (50)
Contact	Tbl_User Tbl_Deal_cust	Varchar (50)
Address	Tbl_User Tbl_Deal_cust	Text
Gender	Tbl_User	Varchar (10)
UserType	Tbl_User	Varchar (15)
Added_Date	Tbl_User Tbl_Categories Tbl_Deal_cust Tbl_Product Tbl_Transaction_Details Tbl_Transaction	DateTime
Added_By	Tbl_User Tbl_Categories Tbl_Deal_cust Tbl_Product Tbl_Transaction_Details Tbl_Transaction	Int
Title	Tbl_Categories	Varchar (50)

FILED NAME	TABLE NAME	DATATPE
Description	Tbl_Categories Tbl_Product	Text
Type	Tbl_Deal_cust Tbl_Transaction	Varchar (50)
Name	Tbl_Deal_cust Tbl_Product	Varchar (50)
Category	Tbl_Product	Varchar (50)
Rate	Tbl_Product Tbl_Transaction_Details	Decimal (18,2)
Qty	Tbl_Product Tbl_Transaction_Details	Decimal(18,2)
Product_Id	Tbl_Transaction_Details	Int
Total	Tbl_Transaction_Details	Decimal (18,2)
Deal_cust_id	Tbl_Transaction_Details Tbl_Transaction	Int
Grand_Total	Tbl_Transaction	Decimal (18,2)
Transaction_date	Tbl_Transaction	DateTime
IGST	Tbl_Transaction	Decimal (18,2)
IGST_Amount	Tbl_Transaction	Decimal(18,2)
CGST	Tbl_Transaction	Decimal (18,2)
CGST_Amount	Tbl_Transaction	Decimal(18,2)
SGST	Tbl_Transaction	Decimal (18,2)
SGST_Amount	Tbl_Transaction	Decimal(18,2)
Discount	Tbl_Transaction	Decimal (18,2)
Dicount_Amount	Tbl_Transaction	Decimal(18,2)

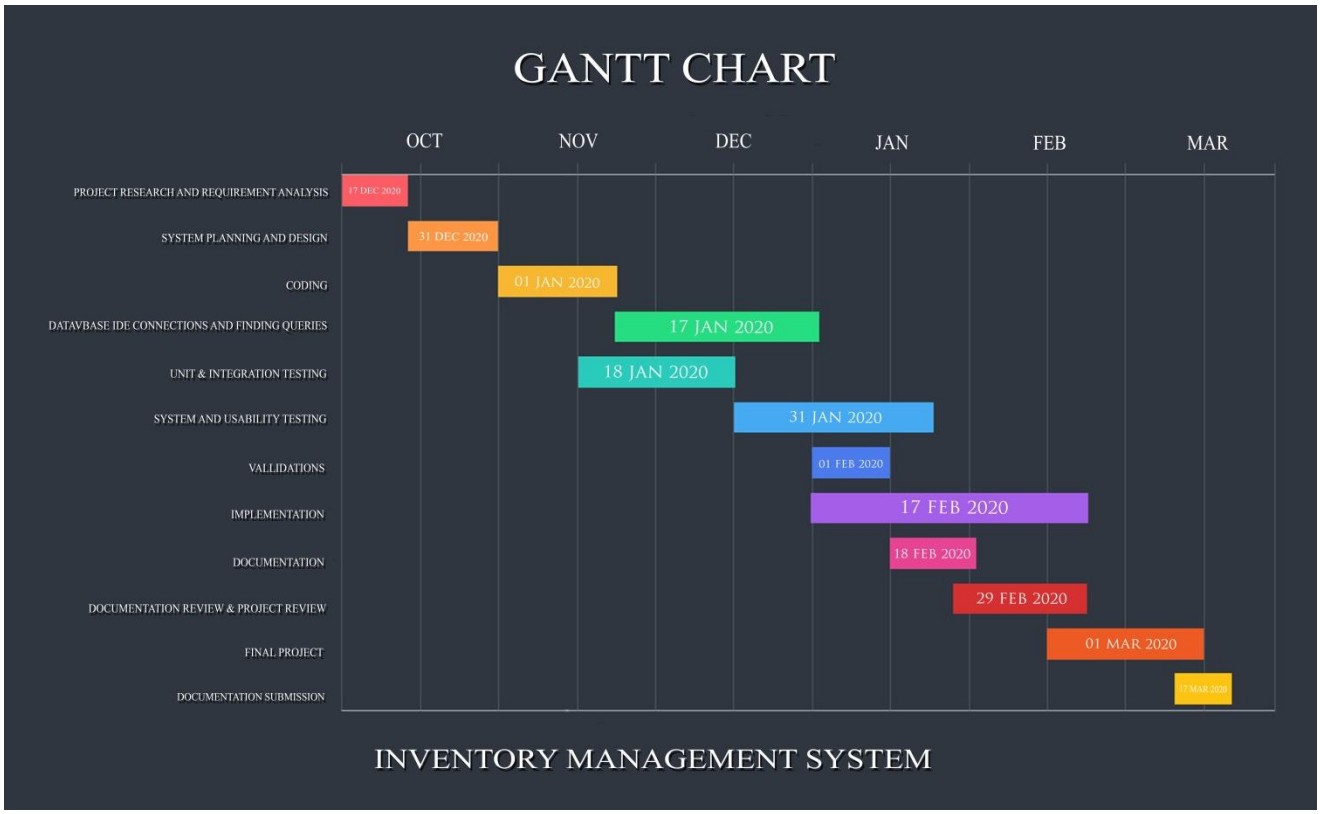
Gantt Chart

Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (months) and a vertical axis representing the tasks that make up the project: **Inventory Management Sytsem**.

It depicts our Project Development Road Map based on task scheduling right from Preliminary Research upto Project Submission. Timelines are indicated that show the time required for the completion of each and every module. It gives a clear idea as to how our project development is phased out on modules to summarise the entire Project Assessment.

Chart



This Gantt chart illustrates the Phases in which the “Inventory Management System” Project was carried out. It gives a systematic overview as to how each task was scheduled and the time required to complete the task. Thus, it summarises the entire Project Assessment.

UML DIAGRAMS

The Unified Modelling Language is a standard visual modelling language intended to be used for modelling business and similar processes, analysis, design, and implementation of software-based systems.

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behaviour of arti-facts of software systems.

UML defines various kinds of diagrams to cover most of the aspects of a system. There are two broad categories of diagrams and they are again divided into subcategories −

Structural Diagrams :-

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

1. Class diagram
2. Object diagram
3. Component diagram
4. Deployment diagram

Behavioural Diagrams :-

Behavioural diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioural diagrams:

1. Use-Case Diagram
2. Sequence Diagram
3. Collaboration Diagram
4. State-Chart Diagram
5. Activity Diagram

ER Diagram

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them.

ER modelling helps to analyze data requirements systematically to produce a well-designed database. So, it is ideal to complete ER modelling before implementing your database.

Entity relationship diagram displays the relationships of entity set stored in a database. This model is based on three basic concepts:

1. **Entities :-** A real-world thing either living or nonliving that is easily recognizable and unrecognizable.
2. **Attributes :-** It is a single-valued property of either an entity-type or a relationship- type. However there exist multi-valued attributes too.
3. **Relationships** :- Relationship is nothing but an association among two or more entities.
4. **Entities:** A real-world thing either living or nonliving that is easily recognizable and unrecognizable.
5. **Attributes :-** It is a single-valued property of either an entity-type or a relationship- type. However there exist multi-valued attributes too.
6. **Relationships** :- Relationship is nothing but an association among two or more entities.

Cardinalities :-

Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

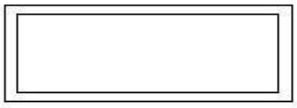
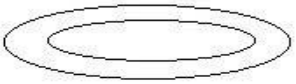
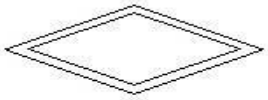
One-to-One − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

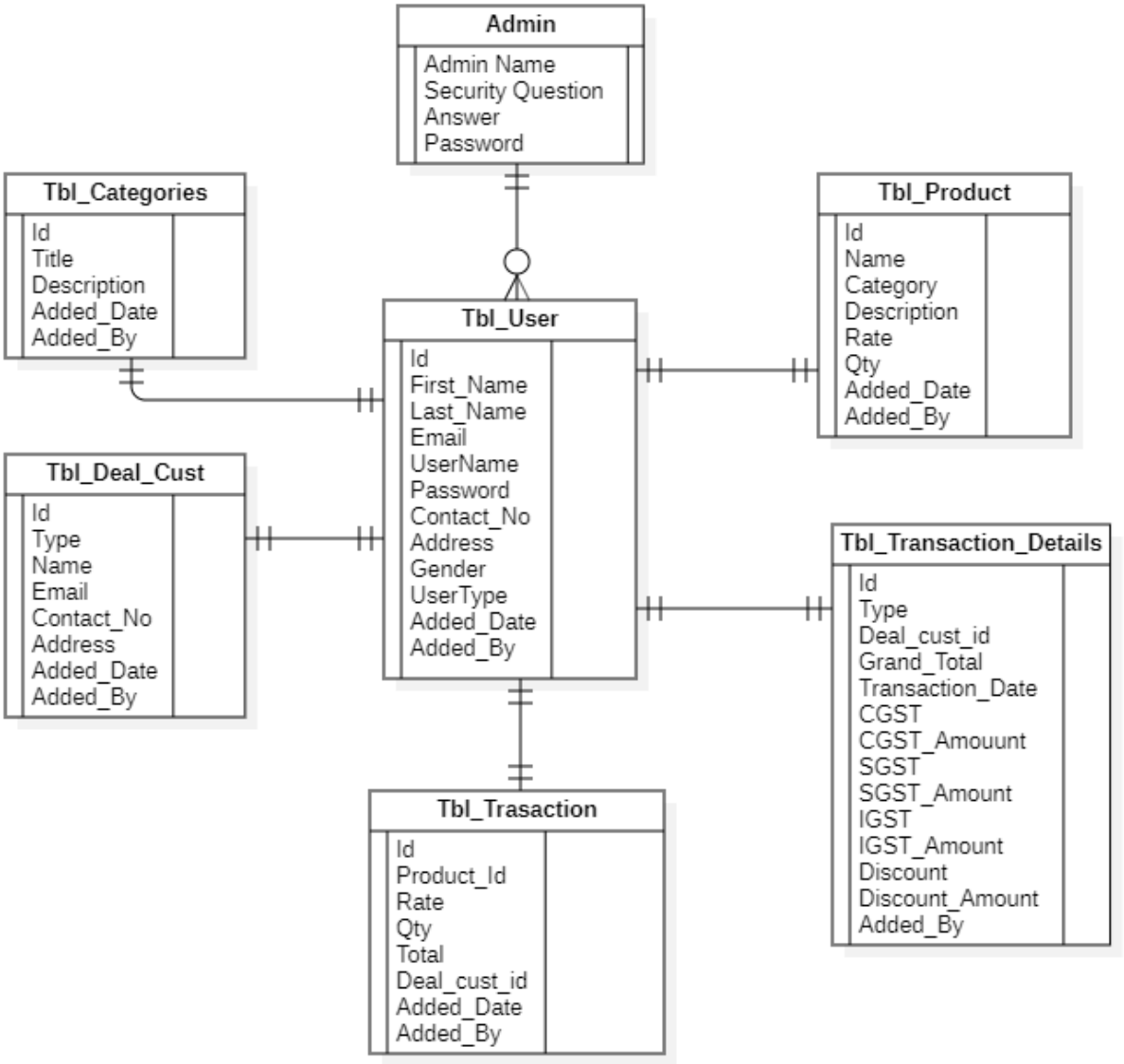
One-to-Many − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

Many-to-One − More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

Many-to-Many − One entity from A can be associated with more than one entity from B and vice versa.

ER DIAGRAM NOTATIONS

Shape	Represents
	Strong Entity
	Weak Entity
	Attribute
	Multivalued Attribute
	Relationship
	Weak Relationship


Admin
Admin Name
Security Question
Answer
Password
Tbl_Categories
Id
Title
Description
Added_Date
Added_By
Tbl_User
Id
First_Name
Last_Name
Email
UserName
Password
Contact_No
Address
Gender
UserType
Added_Date
Added_By
Tbl_Product
Id
Name
Category
Description
Rate
Qty
Added_Date
Added_By
Tbl_Deal_Cust
Id
Type
Name
Email
Contact_No
Address
Added_Date
Added_By
Tbl_Transaction_Details
Id
Type
Deal_cust_id
Grand_Total
Transaction_Date
CGST
CGST_Amouunt
SGST
SGST_Amount
IGST
IGST_Amount
Discount
Discount_Amount
Added_By
Tbl_Trasaction
Id
Product_Id
Rate
Qty
Total
Deal_cust_id
Added_Date
Added_By

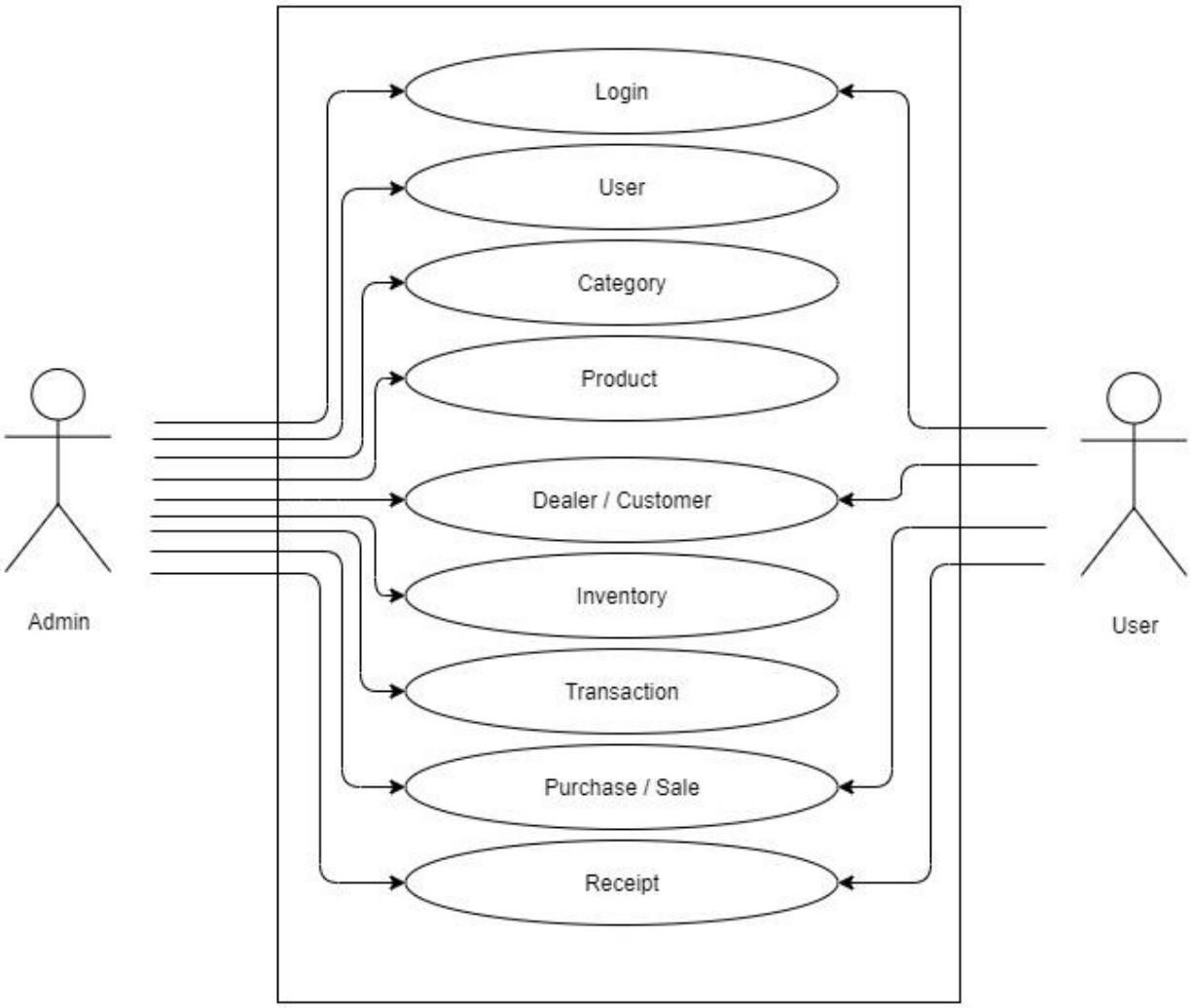
ER DIAGRAM

Use - Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

When the requirements of a system are analyzed, the functionalities are captured in use cases.



USE CASE DIAGRAM

Class Diagram

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

A UML class diagram is made up of :-

1. Set of classes
2. Set of relationships between classes

A class notation consists of three parts:

1. Class Name: The name of the class appears in the first partition.
2. Class Attributes:

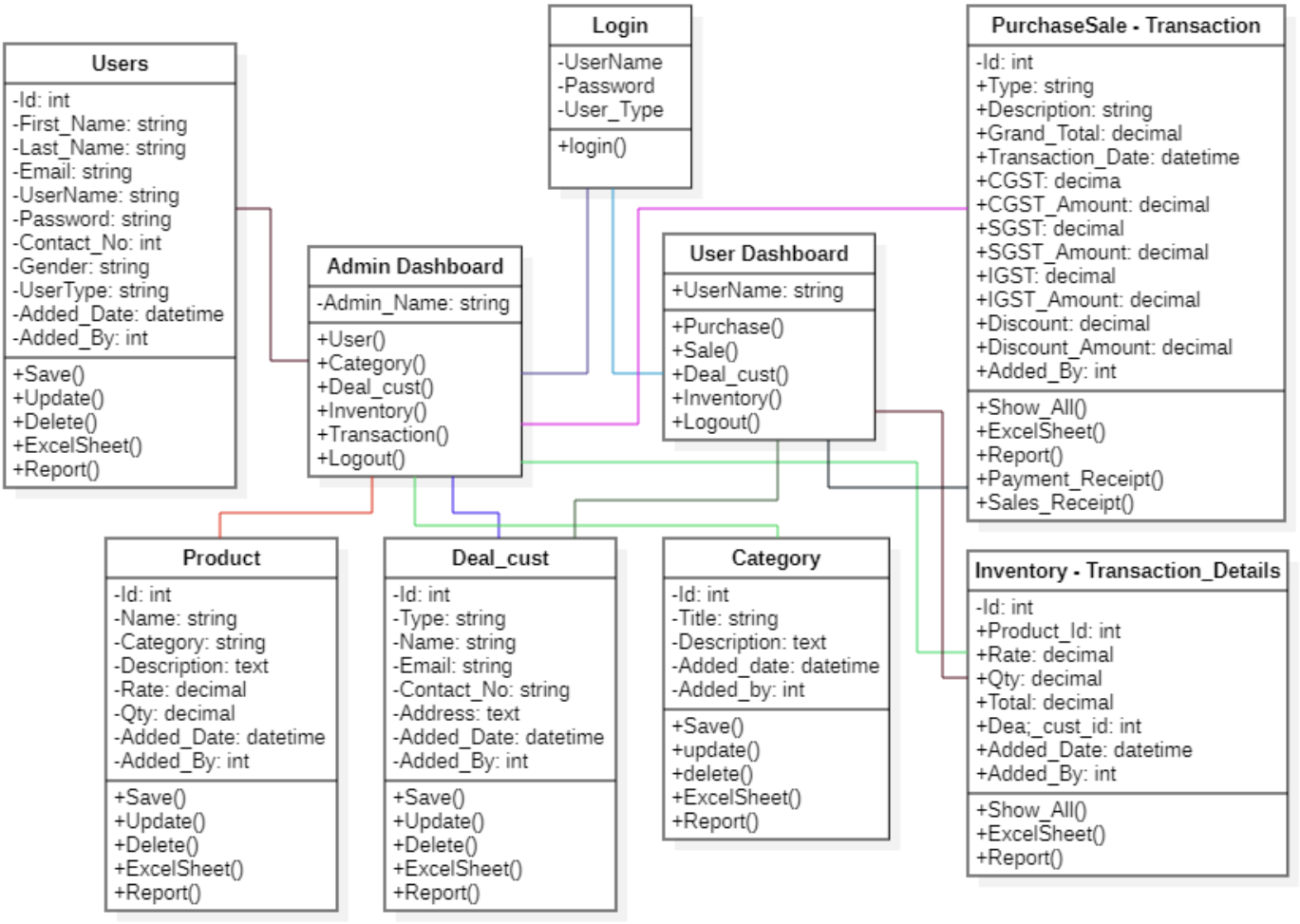
   Attributes are shown in the second partition. The attribute type is shown after the colon.

   Attributes map onto member variables (data members) in code.
3. Class Operations (Methods)

   Operations are shown in the third partition. These are the services a class provides.

   The return type of a method is shown after the colon at the end of the method signature.

   The return type of method parameters are shown after the colon following the parameter name.

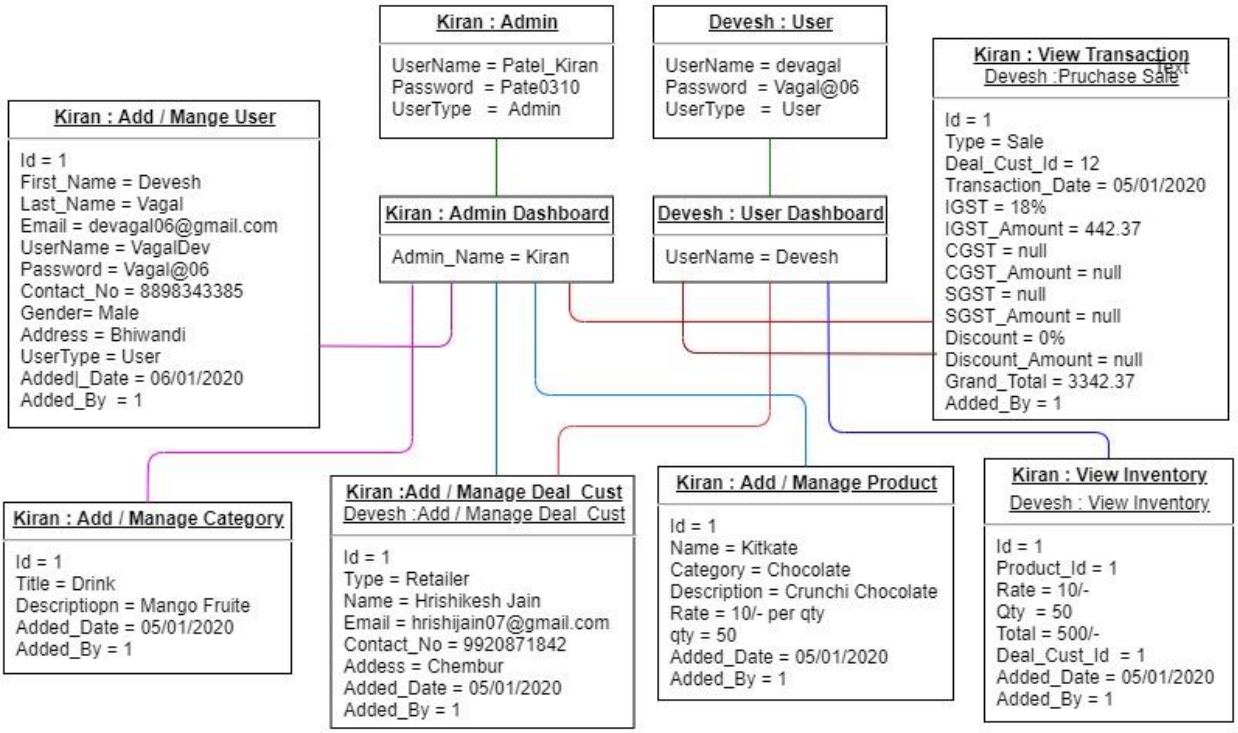


CLASS DIAGRAM

Object Diagram

Object is an instance of a particular moment in runtime, including objects and data values. A static UML object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships at a point in time. It may be considered a special case of a class diagram or a communication diagram.

Object Diagram is used to verify the accuracy and completeness of the class diagram. An object diagram shows this relation between the instantiated classes and the defined class, and the relation between these objects in the system. They are useful to explain smaller portions of your system, when the system class diagram is complex.

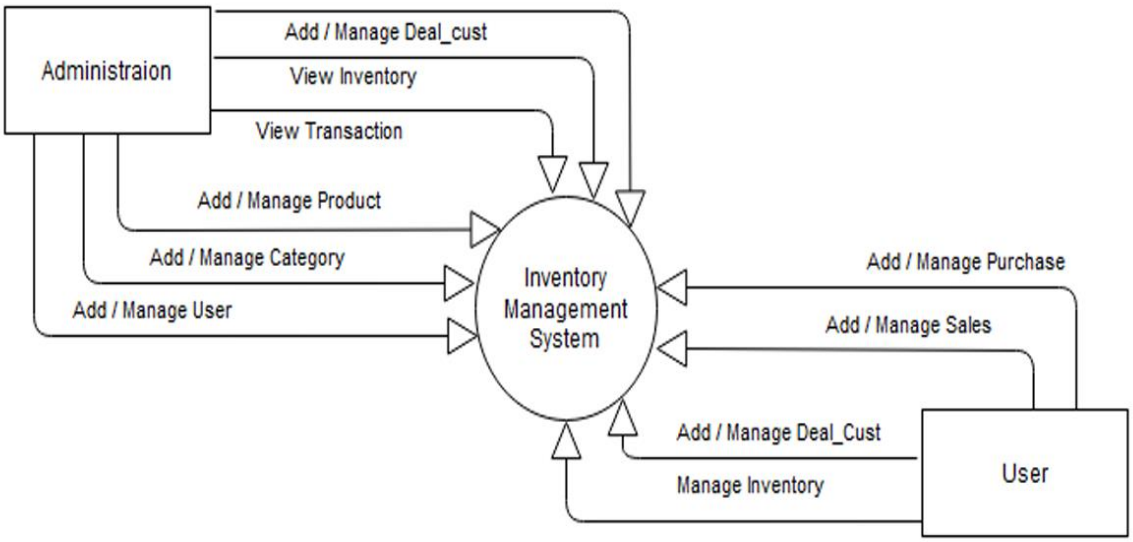


Object Diagram

Context Level Diagram

A context data flow diagram (DFD), also known as a level 0 DFD, gives a broad overview of an information system and the way it interacts with external entities. Being the highest level in a data flow diagram, it contains only one process representing the entire system.

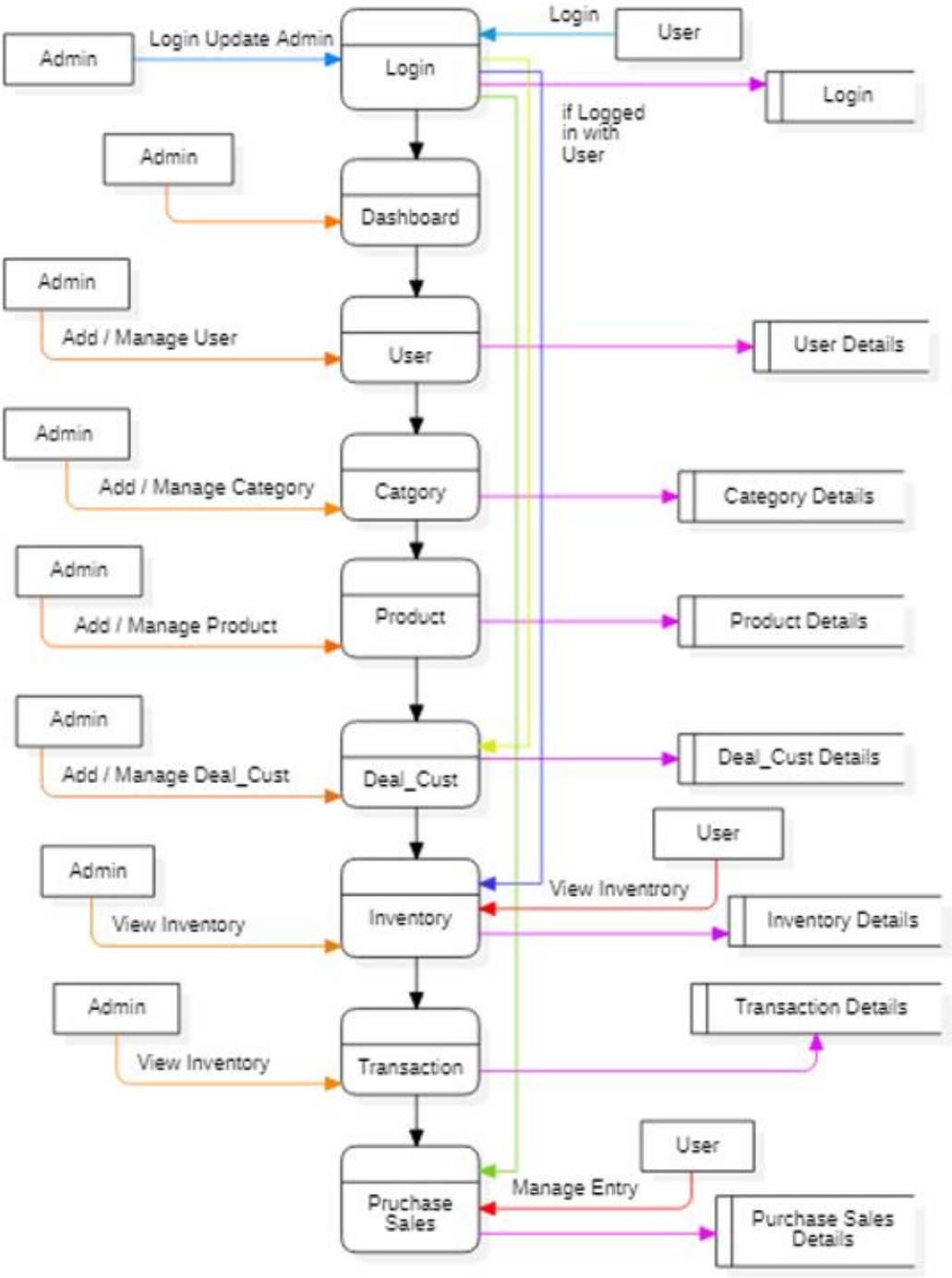
Context Level Diagram describes the overview functionalities required by the external entities; it can be decomposed into a number of sub-level DFDs in hierarchical manner.



CONTEXT LEVEL DIAGRAM

Data Flow Diagram

Data Flow Diagrams (DFD) are used to graphically represent the flow of Data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. A Data Flow Diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles, arrows, etc. to show data inputs, outputs, storagepoints and the routes between each destination.

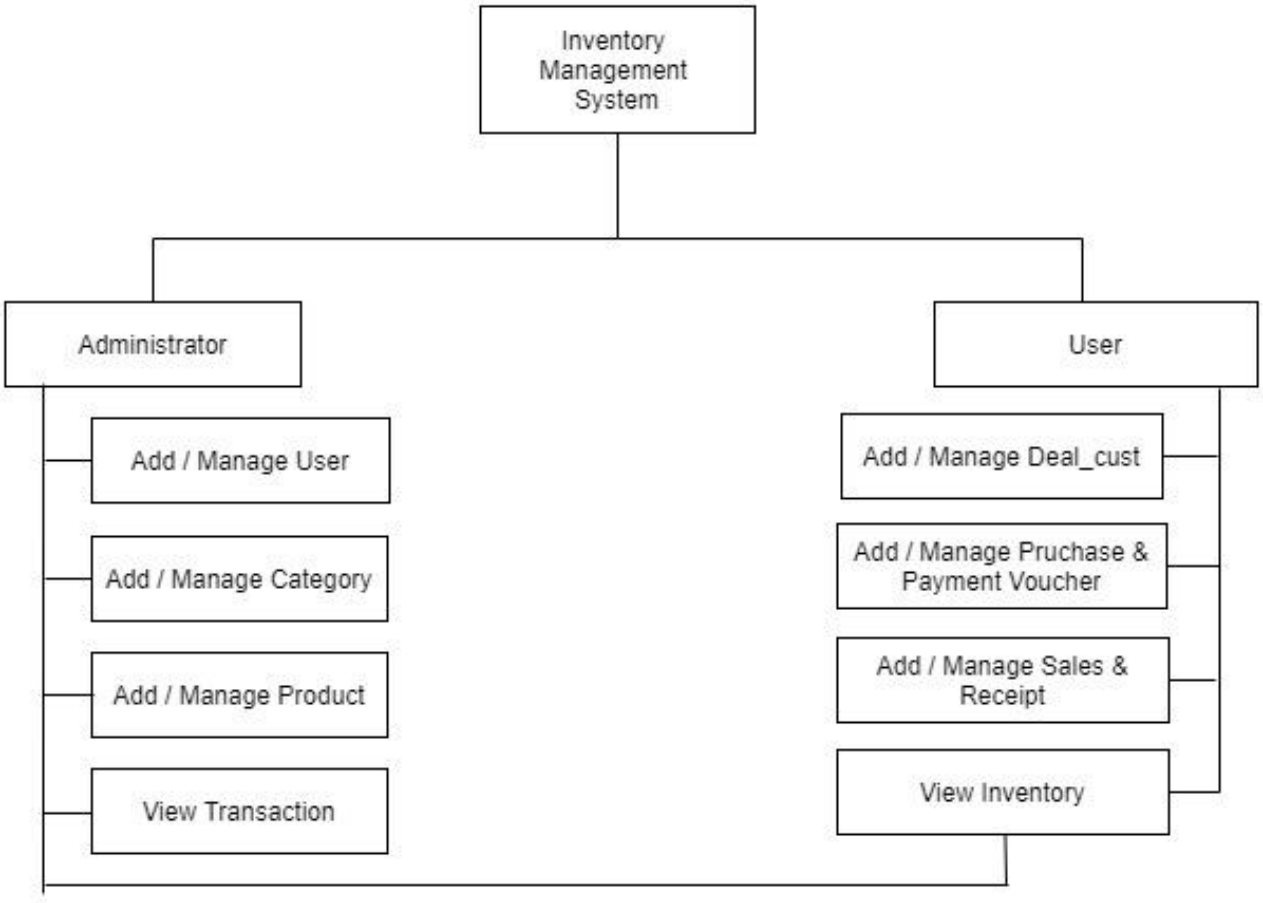


Data Flow Diagram

Functional Decomposition Diagram

Functional Decomposition Diagram (FFD corresponds to the various functional relationships as how the original complex business function was developed. It mainly focusses on how the overall functionality is developed and its interaction between various components.

The purpose of the functional decomposition diagram is to show on a single page the capabilities of an organization that are relevant to the consideration of an architecture. By examining the capabilities of an organization from a functional perspective, it is possible to quickly develop models of what the organization does without being dragged into an extended debate on how the organization does it. Once a basic functional decomposition diagram has been developed, it becomes possible to layer heat-maps on top of this diagram to show scope and decisions.



Function Diagram

Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.

These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations :-

Actors :– An actor in a UML diagram represents a type of role where it interacts with the system and its objects

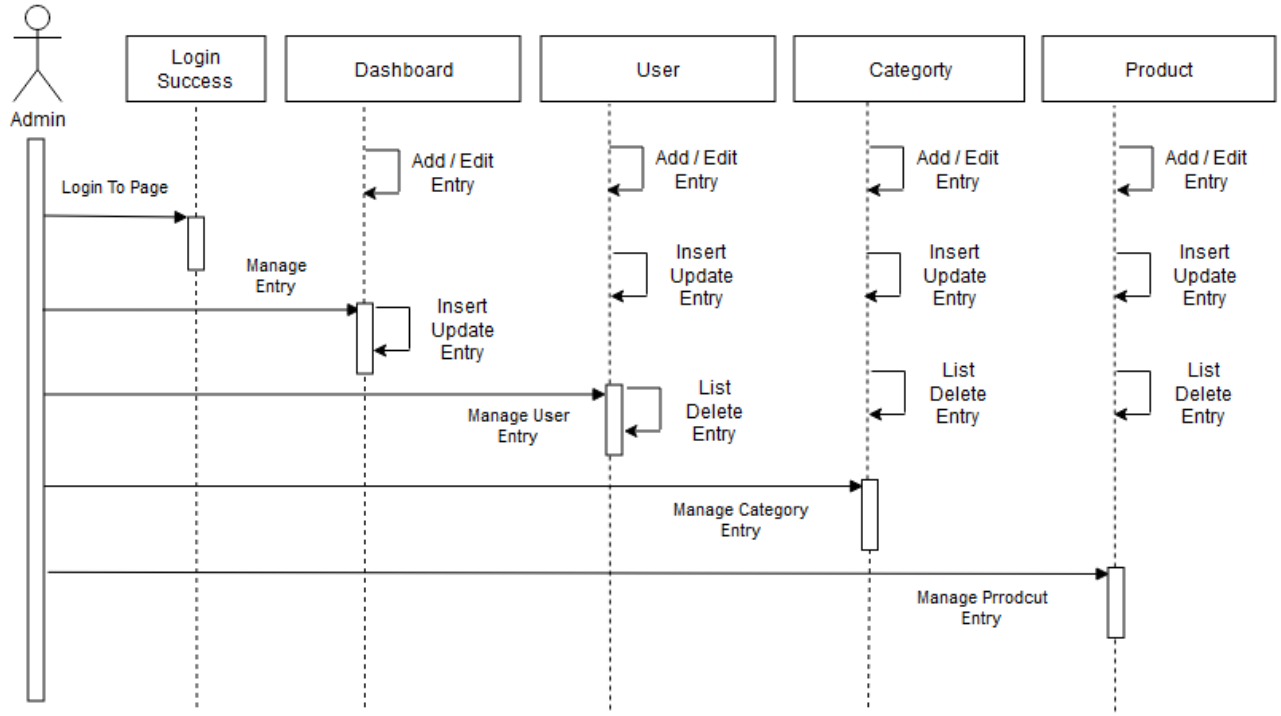
Lifelines :– A lifeline is a named element which depicts an individual participant in a sequence diagram.

Messages :– Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline.

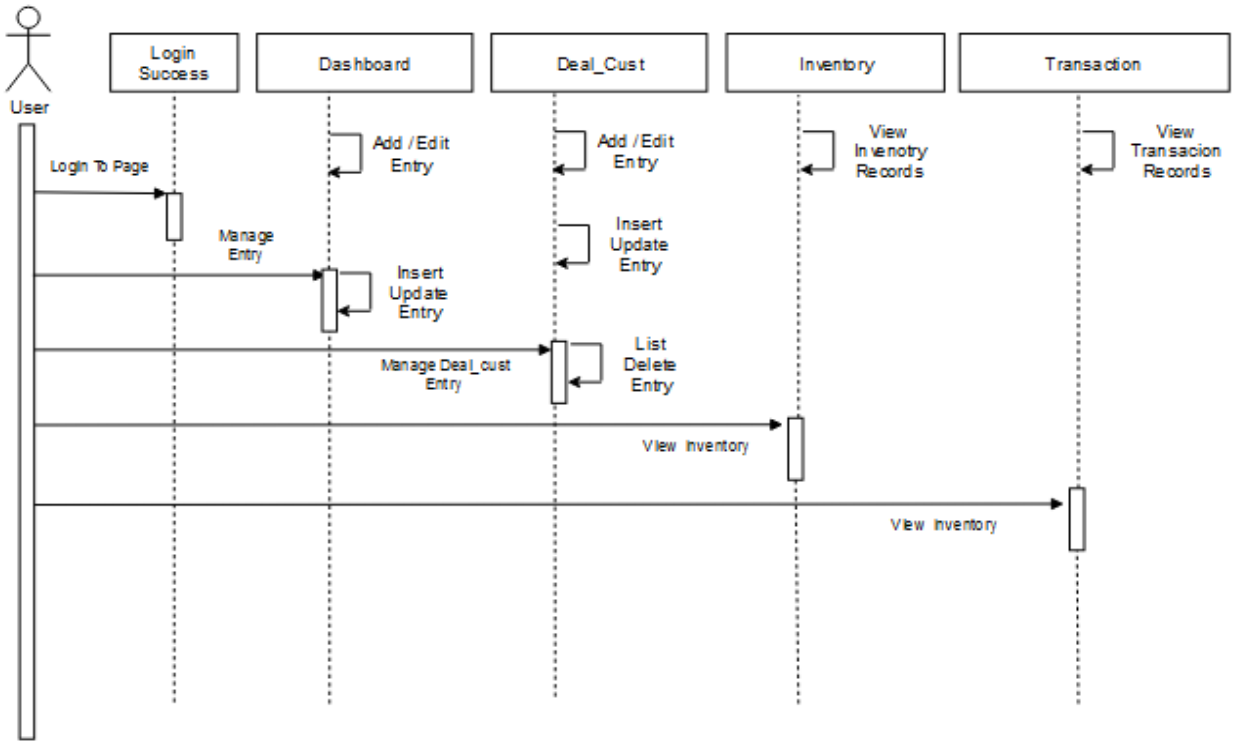
Use of Sequence Diagrams :-

1. Used to model and visualise the logic behind a sophisticated function, operation or procedure.
2. They are also used to show details of UML use case diagrams.
3. Used to understand the detailed functionality of current or future systems.
4. Visualise how messages and tasks move between objects or components in a system.

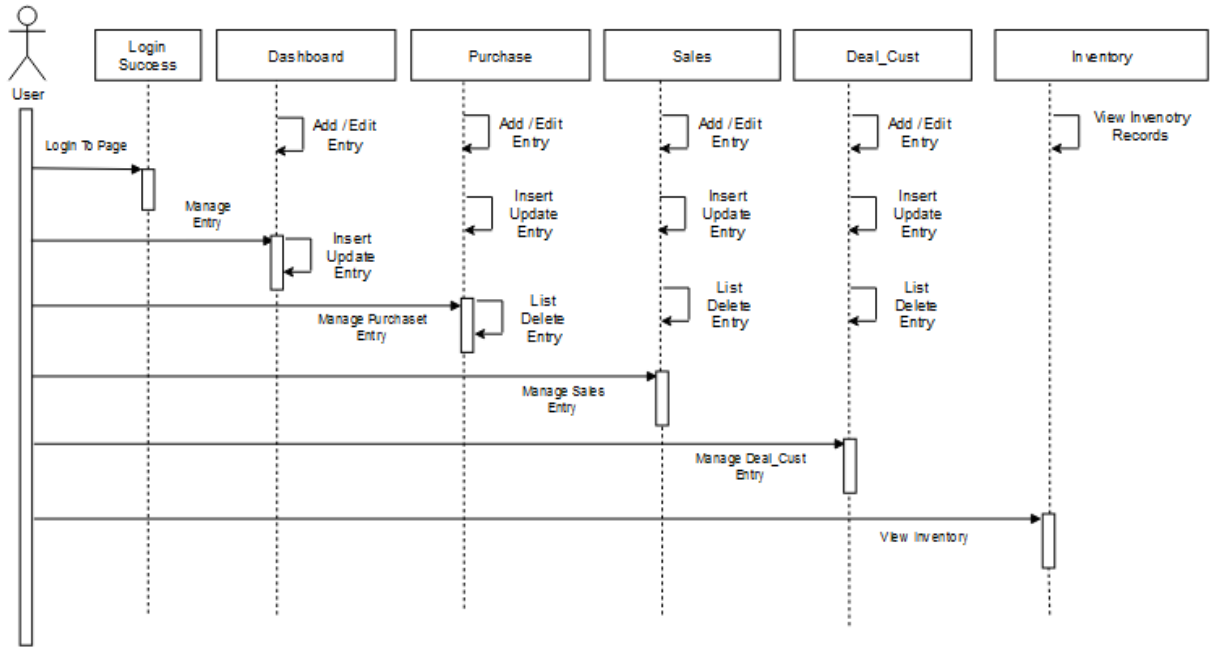
Admin Sequence 1 :-



Admin Sequence 2 :-



User Sequence :-



Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

Activity is a particular operation of the system. They are used to construct the executable system by using forward and reverse engineering techniques.

The purpose of an activity diagram is:

1. To Draw the activity flow of a system.
2. Describe the sequence from one activity to another.
3. Describe the parallel, branched and concurrent flow of the system.

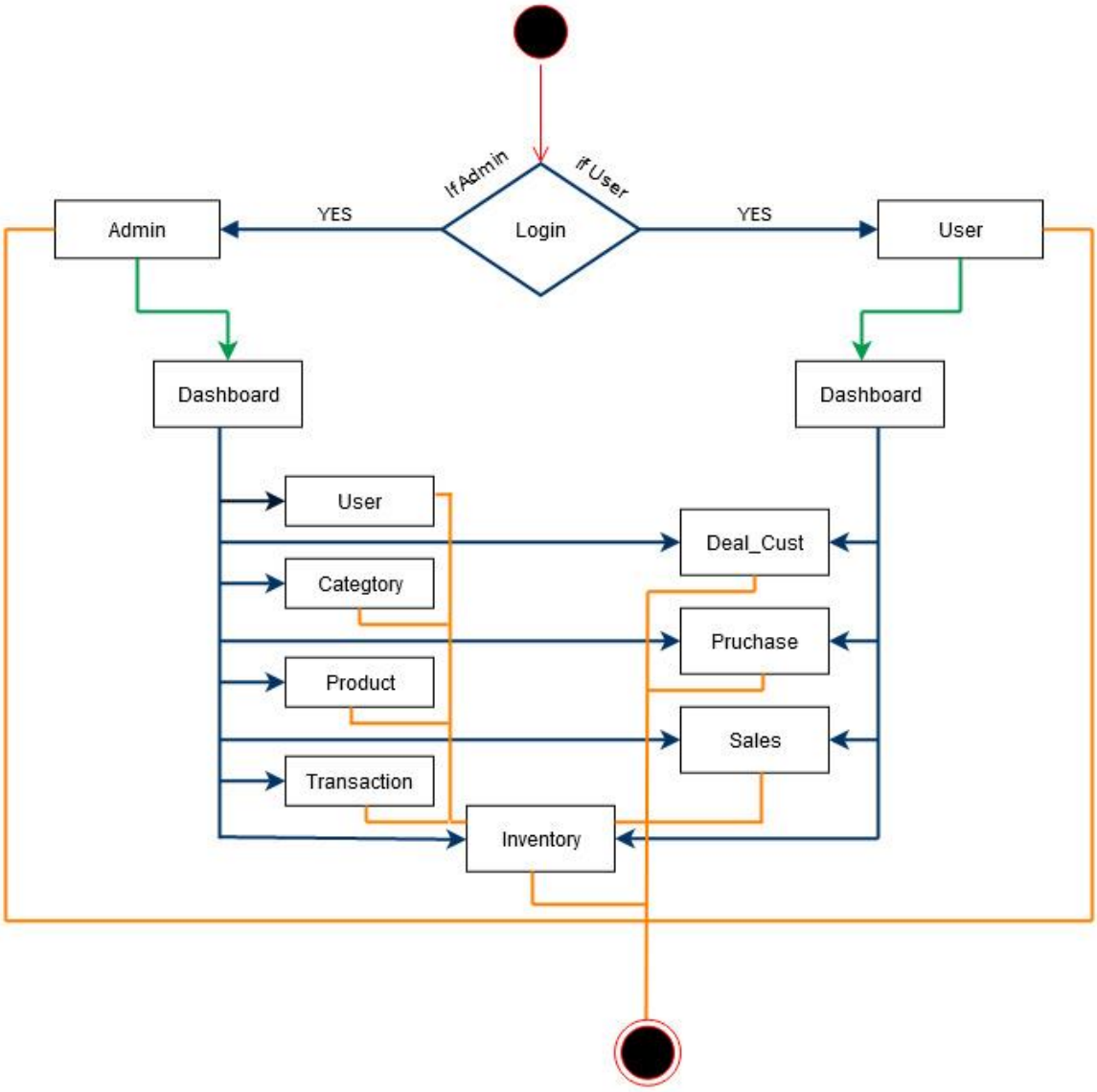
Activity Diagram consists of following elements −

1. Activities
2. Association
3. Conditions
4. Constraints

Activity diagram is suitable for modeling the activity flow of the system. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

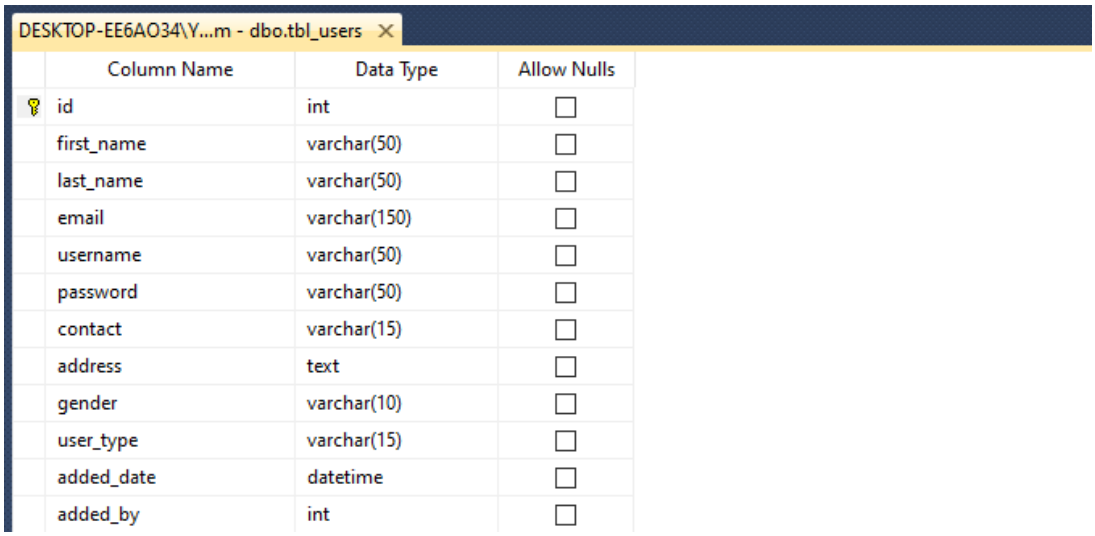
Activity diagram can be used for −

1. Modeling workflow by using activities.
2. Modeling business requirements.
3. High level understanding of the system's functionalities.
4. Investigating business requirements at a later stage.

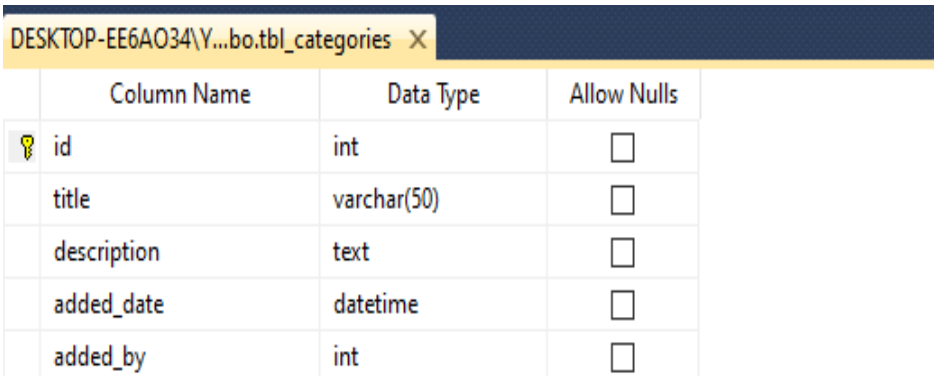


Activity Diagram

Tbl_Users :-



Tbl_Categories :-



Tbl_Deal_Cust :-

DESKTOP-EE6AO34\Y...dbo.tbl_dea_cust

Column Name	Data Type	Allow Nulls
id	int	☐
type	varchar(50)	☐
name	varchar(150)	☐
email	varchar(150)	☐
contact	varchar(15)	☐
address	text	☐
added_date	datetime	☐
added_by	int	☐

Tbl_Product :-

DESKTOP-EE6AO34\Y...dbo.tbl_products*

Column Name	Data Type	Allow Nulls
id	int	☐
name	varchar(50)	☐
category	varchar(50)	☐
description	text	☐
rate	decimal(18, 2)	☐
qty	decimal(18, 2)	☐
added_date	datetime	☐
added_by	int	☐

Tbl_Transaction_Details :-

DESKTOP-EE6AO34\Y...transaction_detail

Column Name	Data Type	Allow Nulls
id	int	☐
product_id	int	☐
rate	decimal(18, 2)	☐
qty	decimal(18, 2)	☐
total	decimal(18, 2)	☐
dea_cust_id	int	☐
added_date	datetime	☐
added_by	int	☐

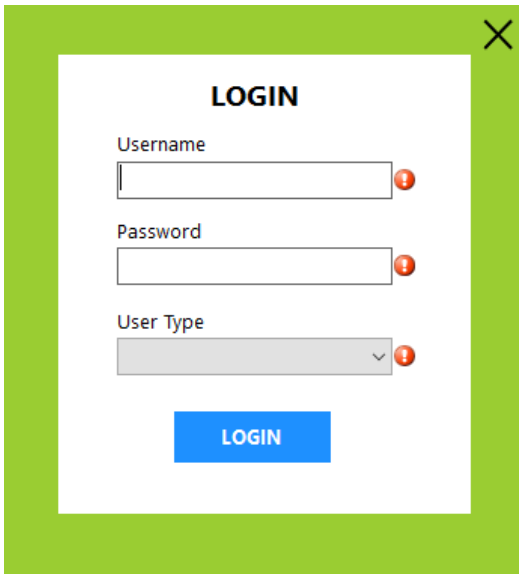
Tbl_Transaction :-

DESKTOP-EE6AO34\Y...o.tbl_transactions

Column Name	Data Type	Allow Nulls
id	int	☐
type	varchar(50)	☑
dea_cust_id	int	☑
grandTotal	decimal(18, 2)	☑
transaction_date	datetime	☑
ig	decimal(18, 2)	☑
cg	decimal(18, 2)	☑
sg	decimal(18, 2)	☑
igamount	decimal(18, 2)	☑
cgamount	decimal(18, 2)	☑
sgamount	decimal(18, 2)	☑
discount	decimal(18, 2)	☑
discountamount	decimal(18, 2)	☑
added_by	int	☑

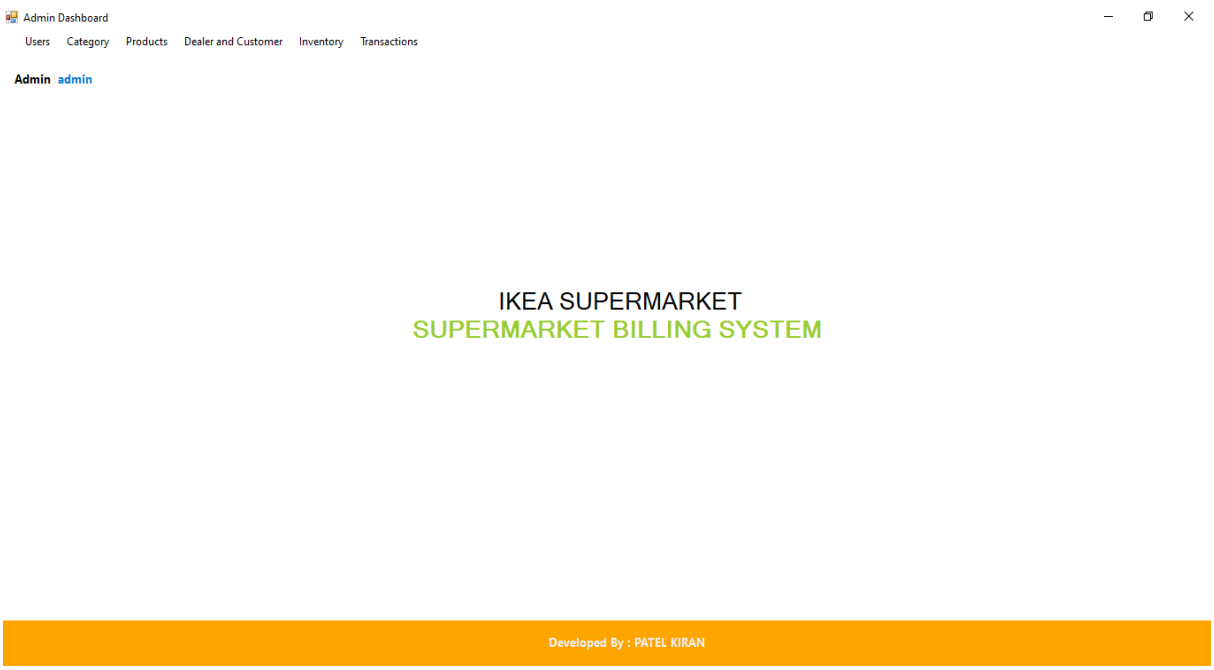
Project Screenshots

Login Form :-



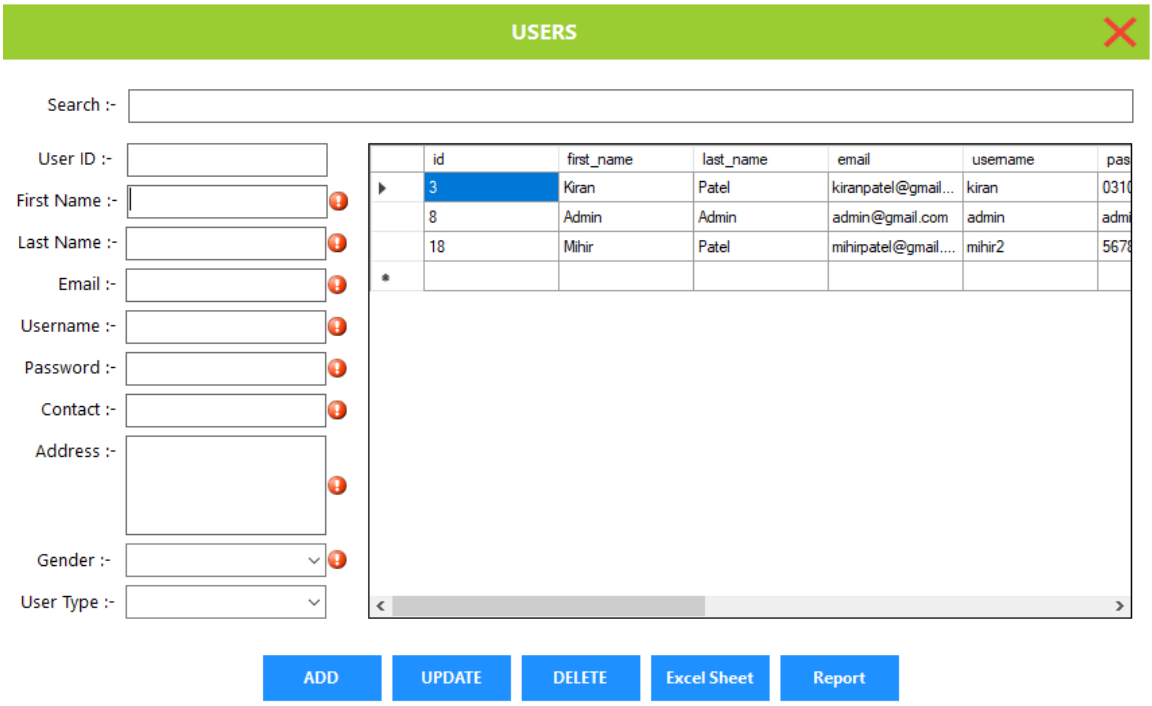
- Two Members Are Login :-
  - Admin
  - User (Staff Member)

Admin Dashboard :-



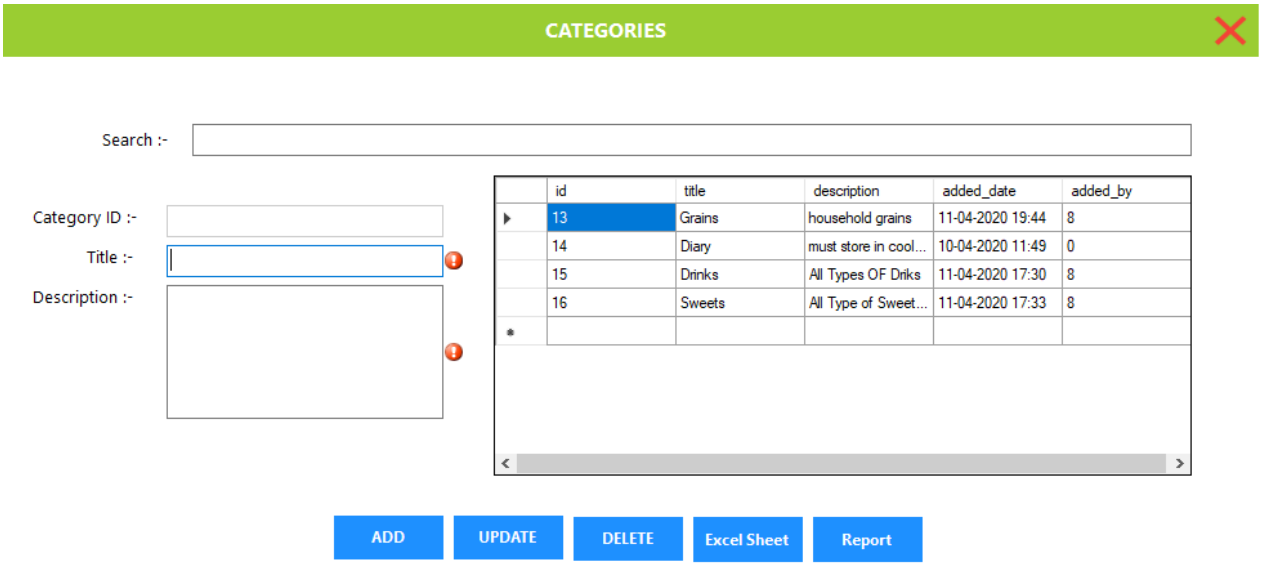
Admin can access all the form like user , category , product , deal_cust , Inventory , Transaction.

User Form : -



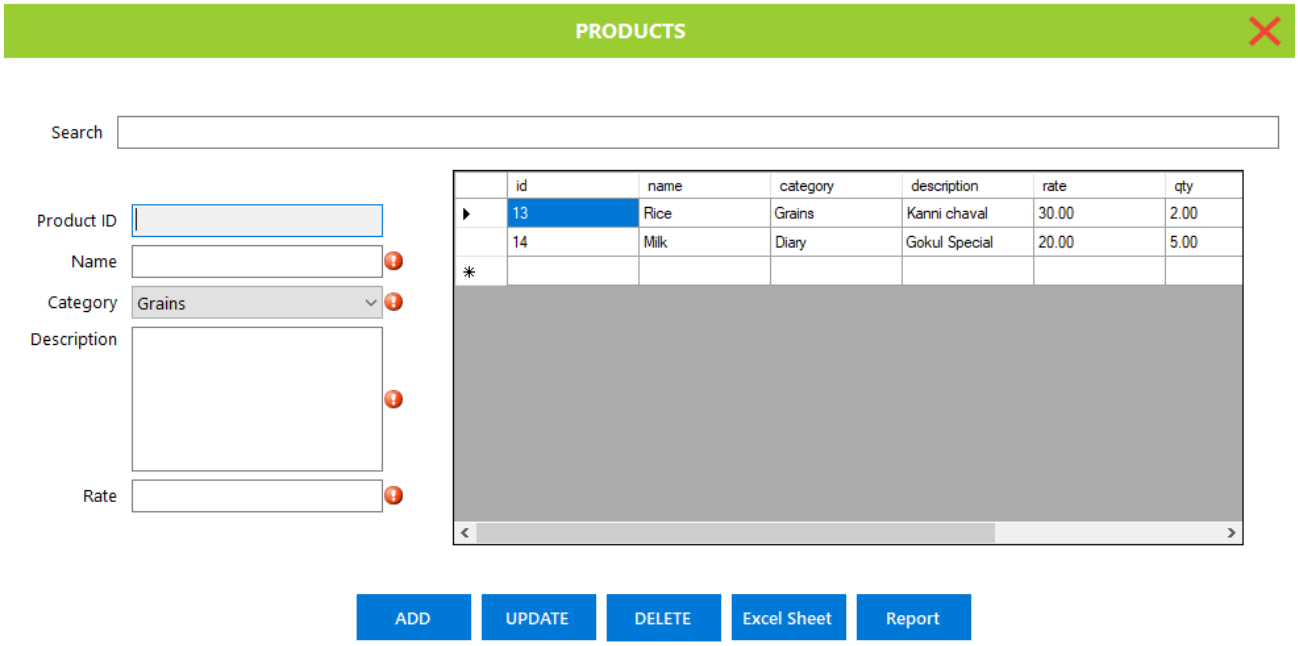
Admin add the user

Category Form :-



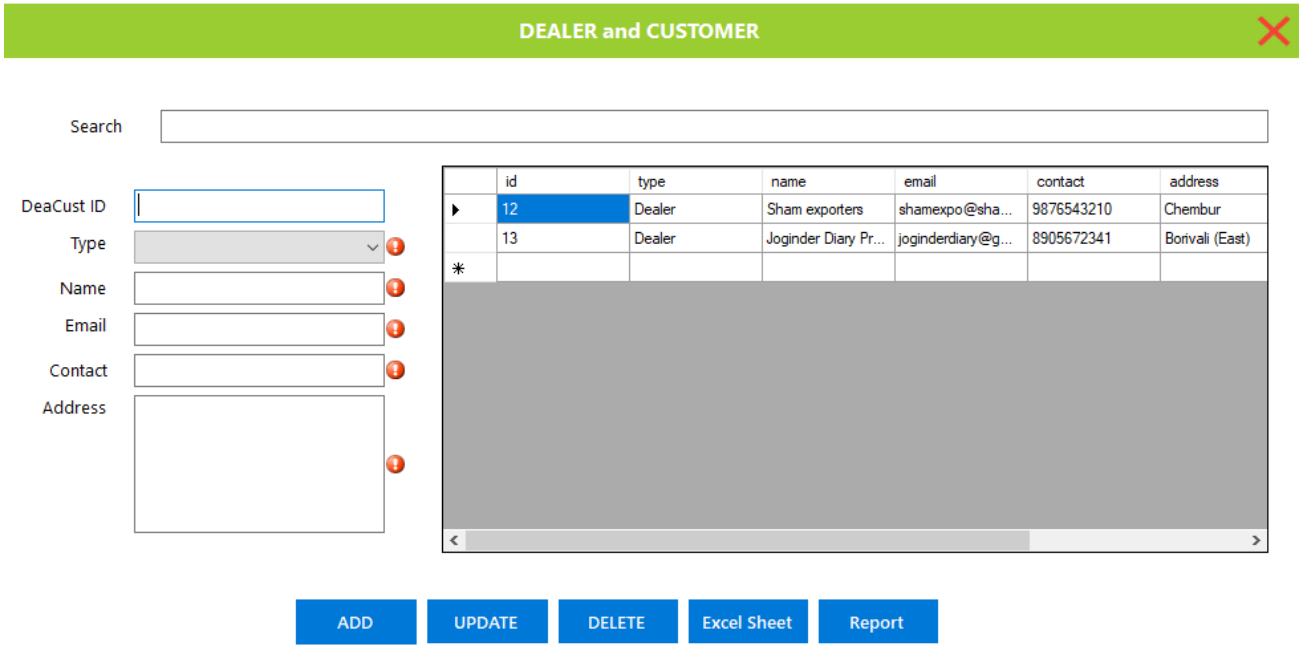
Admin add different types of category such as Food , Drinks , Chocolate , Fruits etc

Product Form :-



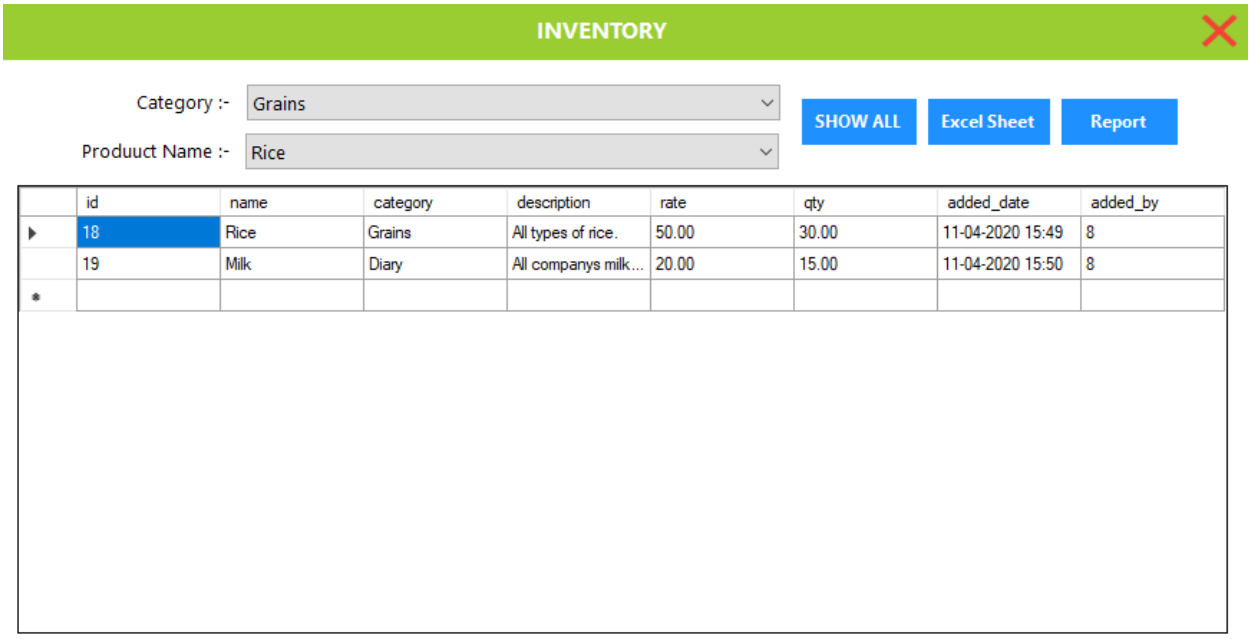
Admin add different types of category such as Drinks – Fruite , Chocolate = Parle etc .

Deal_Cust :-



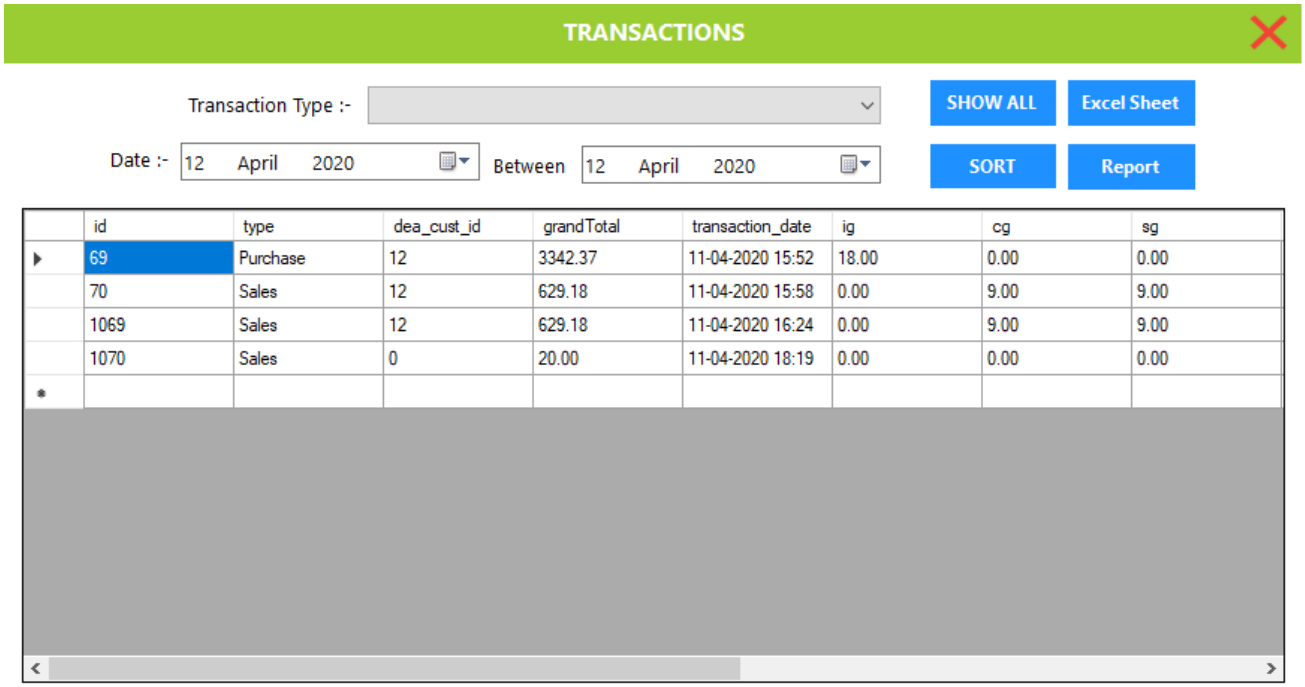
Admin can add dealer or customer details

Inventory Form :-



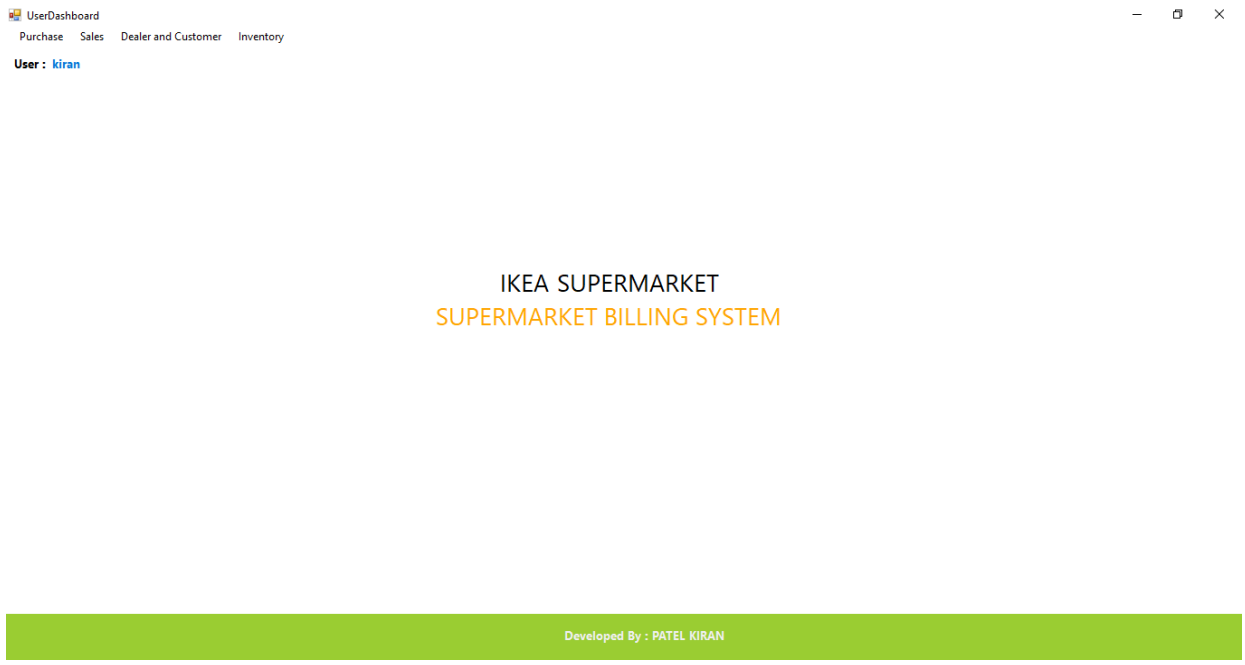
Admin can view Inventory Records of category

Transaction From :-



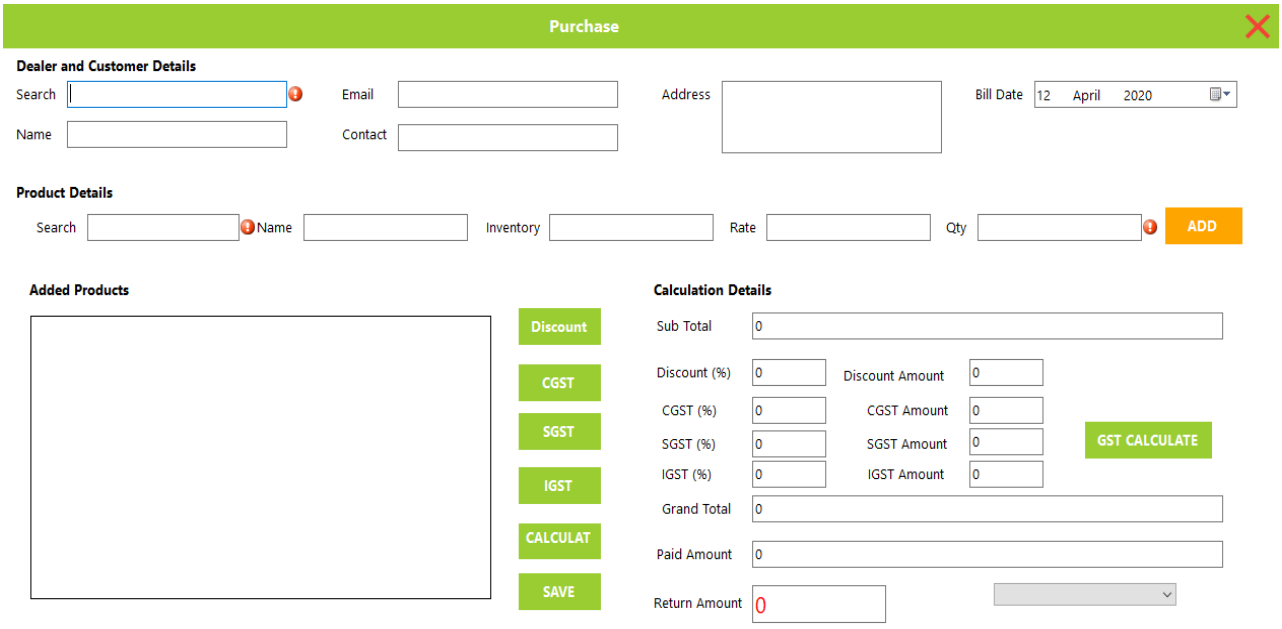
Admin can view all the transaction records of purchase & sales

User Dashboard :-



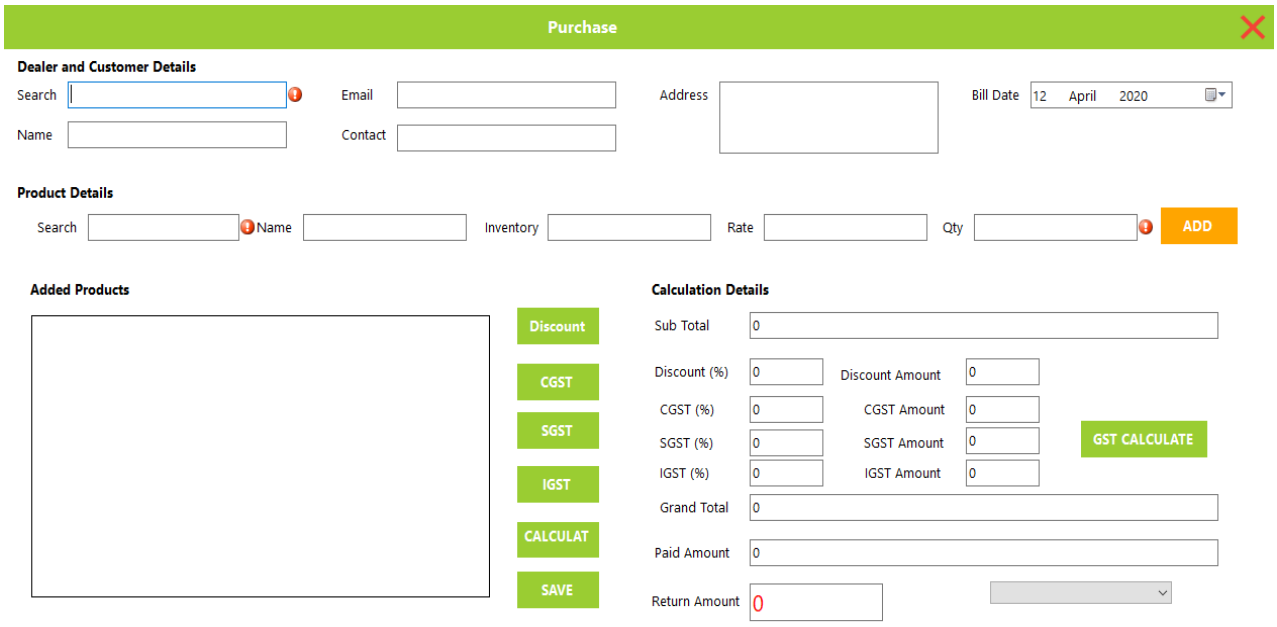
User can access purchase , sales , Inventory , Transaction Form

Purchase Form :-



User can access purchase form for purchase different types of product from dealer

Sales Form :-



User can acces sales form for sale different types of product from customer or dealer.

Payment Receipt :-

IKEA SUPERMARKET

Payment Voucher

Voucher No. : 31211 **Date :** 20 April 2020

Sum of Rupeees (₹)	780
In Words :	Seven Hundred and Eighty Only
Paid To :	Kishor Kunbi
On Behalf Of :	Mitesh Gada

Prepared By **Received By**

Sales Receipt :-

IKEA SUPERMARKET

Shop No. :- 2,3,4,5, Beside Modern Cafe, D.R.
Road, Kurla (West) 400 001
Contact Nos. :- (022) 789546 / 456123

Bill No. :	1141
Date :	20 April 2020

Name : Patel Tejal

Address : Thane Kalher Bhiwandi Road Kapurbawdi (E)

ProductName	Qty	Rate	Total
Milk	10	20.00	200.00
	SUB TOTAL		200.00
	DISCOUNT 2 %		4
	CGST (9%)		16.51
	SGST (9%)		16.51
	IGST (18%)		0
	GRAND TOTAL		229.0200

Notes :-

1) Goods once sold will not be taken back..

THANK YOU FOR YOUR VISIT.

Prepared By

Testing And Validations

Why Is Testing Important?

This is the most important part of the software life cycle. It provides better quality of software to end users; therefore, those end users won't come across software issues. Testing of any software is very important for validating functionality of the software. Testing will provide the following information: It finds issues during early phases, which can be fixed before finalization. It assures stability and reliability of software in different conditions. It helps to provide issue- free software for delivery. Any application must be tested with different methodologies. If the application is not tested properly, then some faulty application will be delivered to customers. Delivering such quality of application will reduce credibility, and the customers will be not delighted with application. Testing is usually conducted by development and quality assurance teams. This testing validates the functionality of the application.

For this **Inventory Management System**, Validation Testing is performed. Various validation checks are implemented to ensure efficient and apt functioning of the System.
Validations are executed with the use of ERROR-PROVIDERS tool in Visual Studio. Main Validation Checks that are performed are:

a. **RequiredField Validation :-** It ensures that the the field on which this particular error-provider has been assigned, should not be left void.

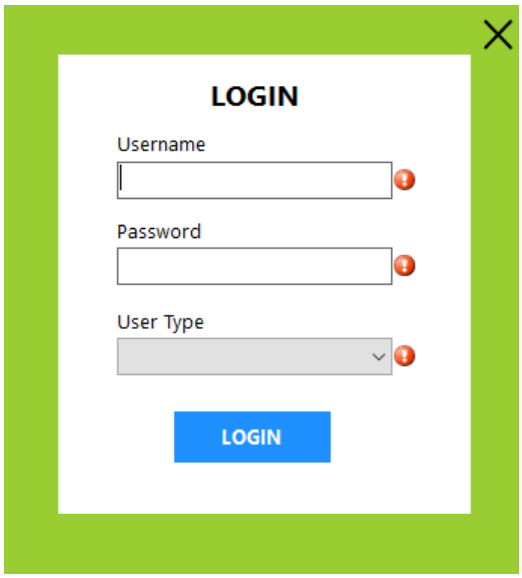
b. **RangeValidation :-** It ensures that the number of characters that a textbox can take should be within the specified Range.

c. **Custom Validation :-** This Validation can be customized to check on the desired input. It is executed for E-mailID and ContactNumber (so that textbox accepts only digits).

d. **CompareValidation :-** This validation is used to compare and check whether the user has entered the authentic Password.

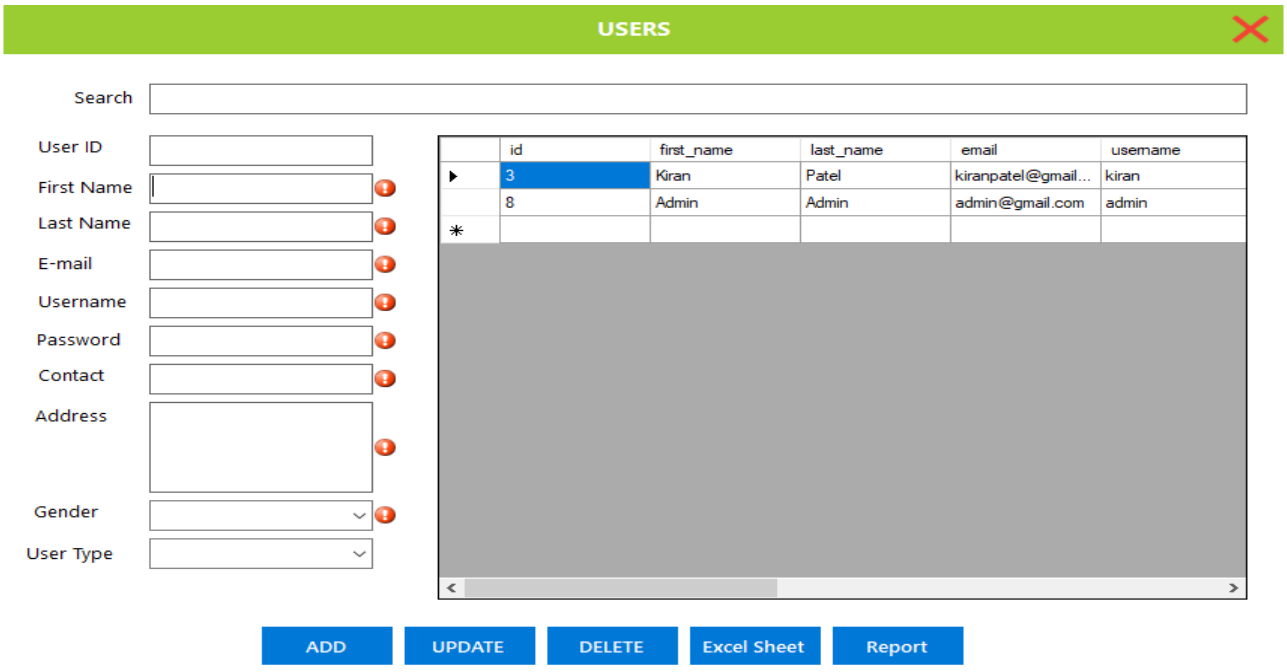
Below are the ScreenShots that depict the implementation of Validations in our System:

Validation Screenshot 1 : -



RequiredField Validation is performed on UserName Field , Password , and USerType. They cannot be left blank.

Validation Screenshot 2 :-



RequiredFieldValidations are performed on: Employee First Name, Employee Last Name, EmailID and Contact Number Custom Validation is executed on EmailID whereas Range Validation alongwith Custom Validation is checked for ContactNumber such that it accepts 10 digit Number Only.

System Maintenance

System Maintenance stands for all the modifications and updations done after the delivery of software product. The software requirements vary according to client needs. Hence the software must be customizable to be able to meet Client Needs.

The Need for Modification in our System can be of the following form:

Market Conditions :- Policies, which changes over the time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger need for modification.

Client Requirements :- Over the time, customer may ask for new features or functions in the software.

Host Modifications :- If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.

Organization Changes :- If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.

In Employee Management System lifetime, type of maintenance may vary based on its nature. It may either be just a routine maintenance task as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature. Following are some types of maintenance based on their characteristics:

Corrective Maintenance :- It includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.

Adaptive Maintenance :- It includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.

Perfective Maintenance :- It includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.

Preventive Maintenance :- It includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

Limitations

Limitations :-

The Inventory Management System makes use of Crystal Reports which are not portable. For the clients, the plug-in needs to be installed on their machine with Visual Studio and Sql Management Studio compatible versions. System design is hard to be customized for Client Needs. Since IMS is Desktop application, remote access is not possible.

Future Enhancement :-

This is a prototype of Inventory Management System. In future software can enhanced or expanded. It can also incorporate different probability and classification techniques.

E.g :- Decision Tree, other AI techniques like Record Report , Bill Generating Receipts using Crystal Report prescribed authorized by software owner. It will work with product rating using bar-code , sentiment analysis & opinion mining.

Conclusion

As you can see the importance of inventory management is very serious, it is one of the most important aspects of any business. The aspect of this part of the business is whether or not you can satisfy the demand of your customers if you aren't sure if you have all the materials available to make the final product Without Wheeled the proper inventory management they would not be able to supply their customers with their ordered ambulance.

And this product is what their entire business is based on, so it is of great importance When they are choosing from the different types of programs or automated systems to help with keeping records accurate, Wheeled to keep in mind that the customer is not concerned with which materials are needed to complete the finished product, but the product is operating as promised based on the contract. This is why they need to make sure that any processes or programs that they do decide to use are going to be beneficial to their needs as well as the needs to service their customers.

In addition, the plans for the maintenance of having proper inventory levels need to be in place and also adjusted when the company grows and as the business dictates If Wheeled the new suggestions they will be on the right track to having a well established business.

Bibliography

- For UML Diagrams :-
  - staruml and lucidcharts :
  - https://staruml.io https://www.lucidchart.com

- Shared Libraries(.dll) used :-
  - Bunifu-UI,
  - Nugget Packages
  - SAP Crystal Reports in Visual Studio

- For Coding Reference :-
  - C#: The Complete Reference Book. Author: Herbert Schildt
  - https://www.youtube.com(Vetrivel D. UI Academy, Fox Learn)
  - https://stackoverflow.com
  - https://github.com
  - https://www.quora.com
  - https://www.c-sharpcorner.com