# Design

**Dynamic Modelling**

## Sequence Diagram:

A kind of diagram that arrange the interaction of the object in the time sequence. It is also called the event diagram. It describe how the classes are interacting with each other while exchanging message over time. It helps to validate and visualize different runtime scenarios. It contain parallel vertical line that we called the lifeline also contain object or the process simultaneously. (Gary Afable, 2010)

**Justifications:**

We used to draw this diagram at the requirement period of certain period. It is important to create this diagram because it helps to show the behavior of different object which we are going to develop in our certain projects. Through this the collaboration of the object can be shown.

**Advantages:**

* It is quite helpful for the developer and analyst of the business to get common or the basic understanding.
* It is used to validate class diagrams against use case in object orientated analysis as well as it helps to show the timing interacting of objects or the entities.

**Disadvantages:**

* It must be elaborate and define for every possible scenario and need fully defined class module.

**Notation used:**

It also contain different following notations while creating the basic sequence diagram.

* + **Class Roles or Participants**

The main role of participant is to show how an object is going to behave in context.

Object symbol - Sequence diagram

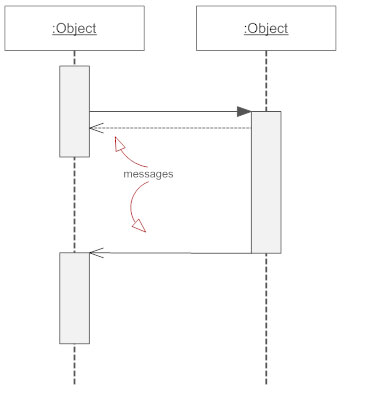
* + **Activation or Execution occurrence**

It is the vertical boxes that used to place on the lifeline and helps to show how much time an object needed for the completion of task.



* + **Messages**

It represent the arrows that can be dotted or non-dotted and helps to shows the communication between various objects.



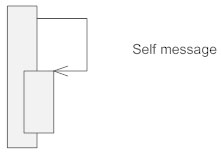
**There are different kinds of messages which are listed below:**

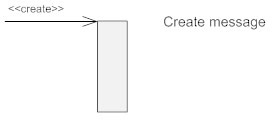
Synchronous message - Sequence diagram

Simple message

Asyncrhonous message

Reply messages





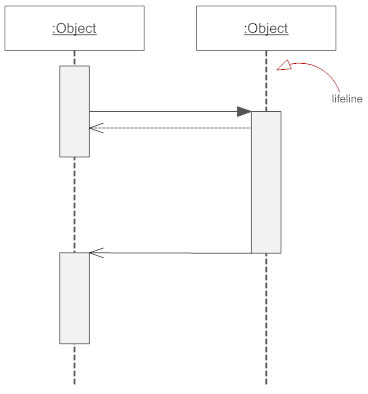
Delete message

Found message

Lost message

* + **Lifelines**

The vertical dotted line that we connect through the object represent the lifeline.



**Sequence Diagram**

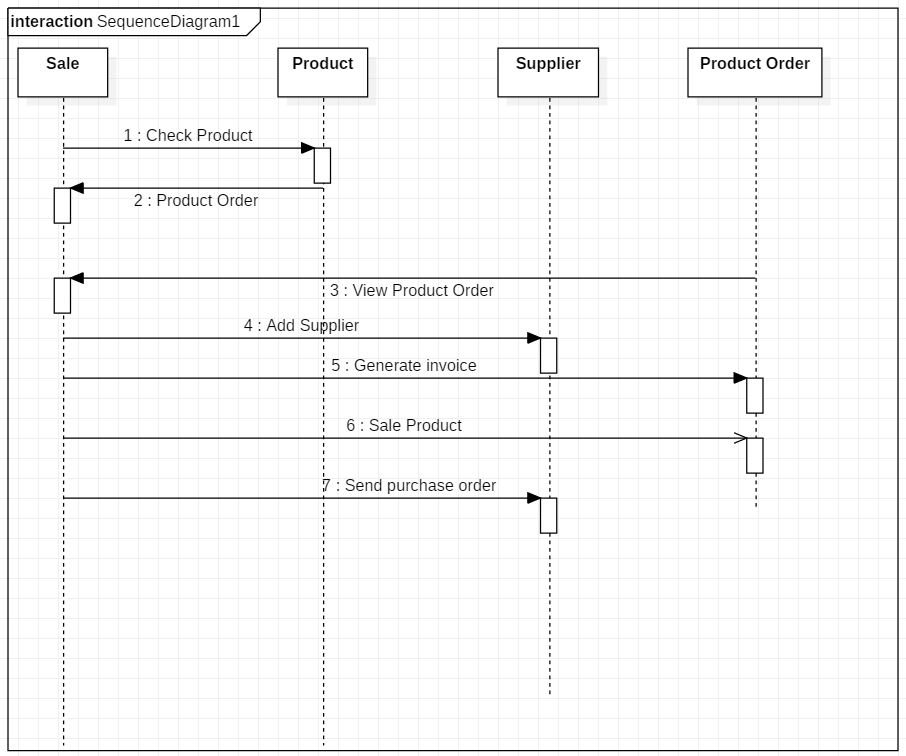


Figure 7: Item Sequence Diagram

**Description**

It interact among classes or entities above the diagram that are Sale, Product, Supplier, and Product Order in terms of exchanging the message or showing the activity done to manage the business through the inventory management system. It tell about each and every action that user have to do after accessing the Inventory Management system.

## Activity Diagram:

Activity diagram is also known as the behavioral diagram that describes all the dynamic aspect that contained by the system. It is more advanced then the flow chat diagram. It describe the flow of the system or the flow of activity. (Ivory-Ndiaye, 2013)

**Justifications:**

They are used for all the stages of system or the software development. Activity diagram helps to describe how the system will show the flow of one activity to another activity. It make the developer as well as the user to understand the system easily. It is quite similar to a flowchart or data flow diagram.

**Advantages:**

* It deals with the control flow, it means how the system flow logically.
* It will show how the system will behave when it is interacted.
* It helps the user or the developer to know how the operations are getting performed.

**Disadvantages:**

* Activity diagram does not describe in detail, that how the system behave and because of this problem they may not be used in lieu sequence diagram.

**Notation used:**

While creating the activity diagram different notation are used which are listed below:

* + **Activity**

It helps to show the set of actions



* + **Action**

Used to perform the task.

Activity Diagram Notation - Action

* + **Control flow**

It represent the sequence of flow or execution.

Activity Diagram Notation - Control Flow

* + **Object flow**

Show the flow of object from one action to another.

Activity Diagram Notation - Object Flow

* + **Initial Node**

Used for beginning the diagram.

Activity Diagram Notation - Initial Node

* + **Activity final node**

Used to end or stop the diagram.

Activity Diagram Notation - Activity Final Node

* + **Object node**

It helps to represent the objects.

Activity Diagram Notation - Object Node

* + **Decision node**

It decide where action have to flow in activity.

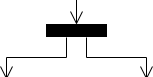
Activity Diagram Notation - Decision Node

* + **Merge node**

Activity Diagram Notation - Merge Node

* + **Fork node**

Behavior split into parallel flow of actions.

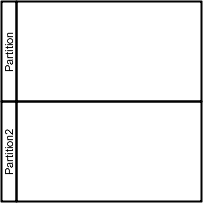


* + **Joint node**

Unite together the set of concurrent actions.

Activity Diagram Notation - Join Node

* + **Swim lane or partition**



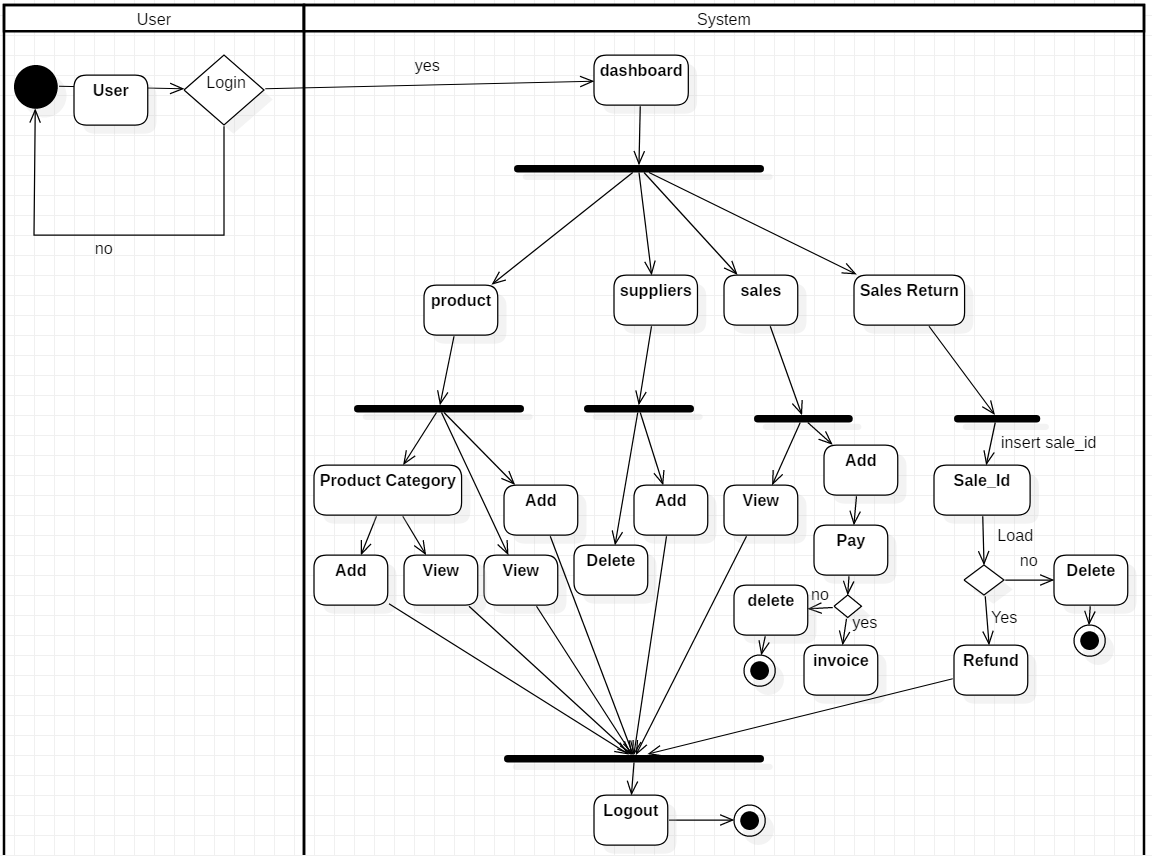
**Activity Diagram**

Figure 8: Activity Diagram

**Description of the system**

According to the system first of all the user need to ‘login’ the system, then only the user will able to open the ‘dashboard’. In dashboard you will have different option to manage the system. The option like Stock Supplier, Product, sales return etc. Also, in all particular option it contain Add, View, Edit, Delete the particulars data like, ‘Supplier’ option contain Add Supplier, Delete supplier, Edit supplier etc. Product also contain Add Product, View Product, Delete Product, after clicking the option product. Beside these it have some other options which do their particular work and finally the user can ‘logout’ after the completion of management.

**Structural Modelling**

## Final Class Diagram:

It the kinds of diagram that made to provide some overview as well as its logical structure of the software in terms of attributes, in terms of classes, methods, as well as relationship between the various classes. It helps to show the structure of the system that we are developing.

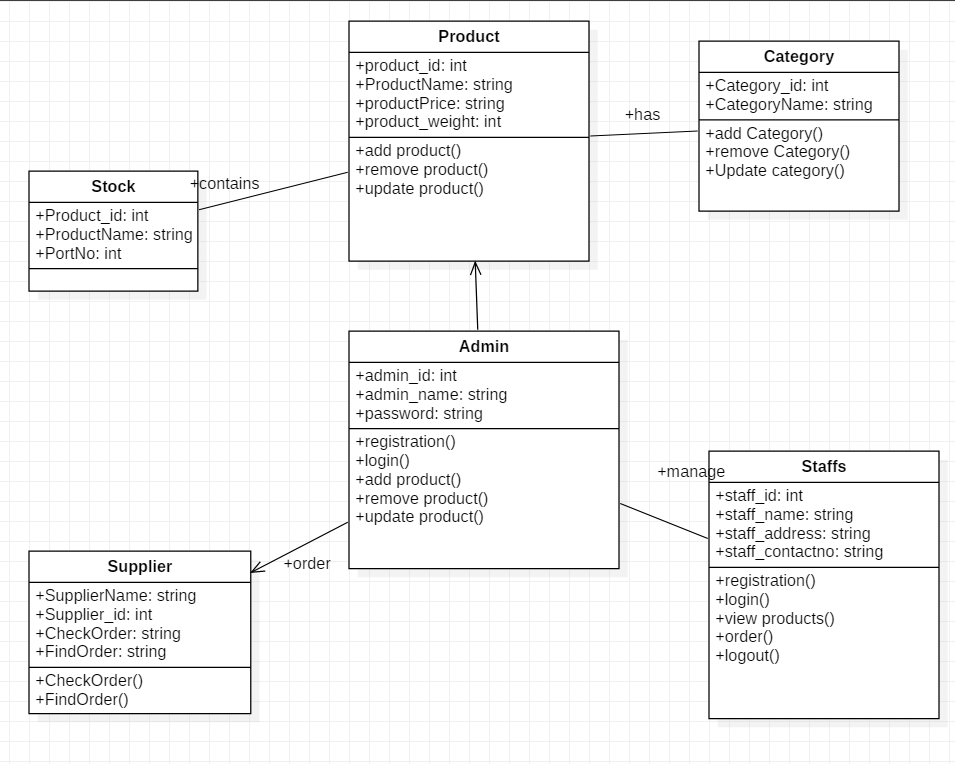


Figure 9: Final Class Diagram

**Description of Diagram**

The stock contains product. One stock contains more than one product. The product has category. In stock there is staff and admin which manage the product. All the product is manage by admin. The product is supplied by supplier.

## DFD:

DFD means the data flow diagram that show the follow of data and information in a certain system and how the system process it in terms of input and output. It really focus on data and information of the certain system where it come and where it go. It became so modified and popular in 1970s for the development of the system and software. (jira, 2010)

**Why DFD:**

Through data flow Diagram, we can give the overview of the software or the system without going the deep detail of the system.

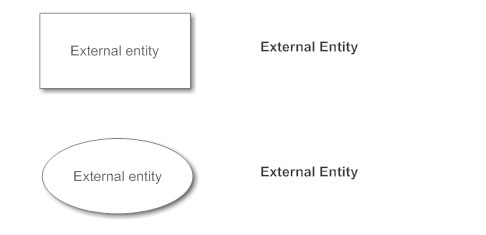
**Some important point:**

* Name should be unique.
* Decision paths (diamond node) represent logical expression.
* DFDs depict flow of data and not order of events like a flowchart.

**Notations used:**

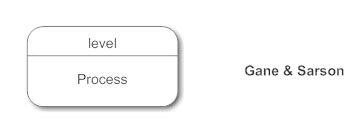
1. **External entity:**

These are the object that are used in communication and found outside of the system. They are so much helpful in sending and receiving the data and information as input and output.



1. **Process Notation:**

It is used to change the data. Here incoming data flow got changed into outgoing data flow.



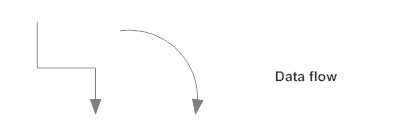
1. **Data Store Notation:**

Data store is like a place where data and information of the system store. It is like a repository for the data.



1. **Data Flow:**

It is the way where the data and information used to flow. They are like Pipeline which are denoted by the arrows.



## **Diagram**

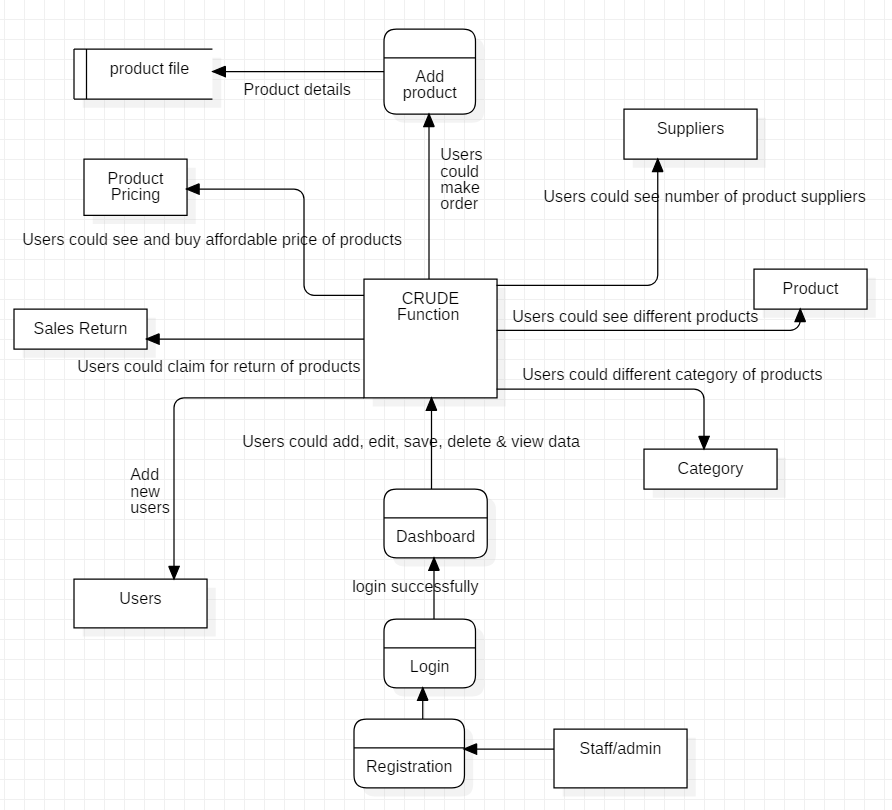


Figure 10: DFD for inventory management system

**Database Modelling**

## ER Diagram:

ER diagram is conceptual module and representational module which is used to show the relationship between different classes which are stored in the database. It is more helpful to design the database. It contain so many symbols and connector and illustrate the logical structure of database. (toyee, 2010)

**Why ER diagram:**

Through ER diagram it make the developer easy to understand the Database. It is high level conceptual module which is easy if we have the idea about entity relationship. It is also the good communication tool to design the Database.

**ER Diagram:**

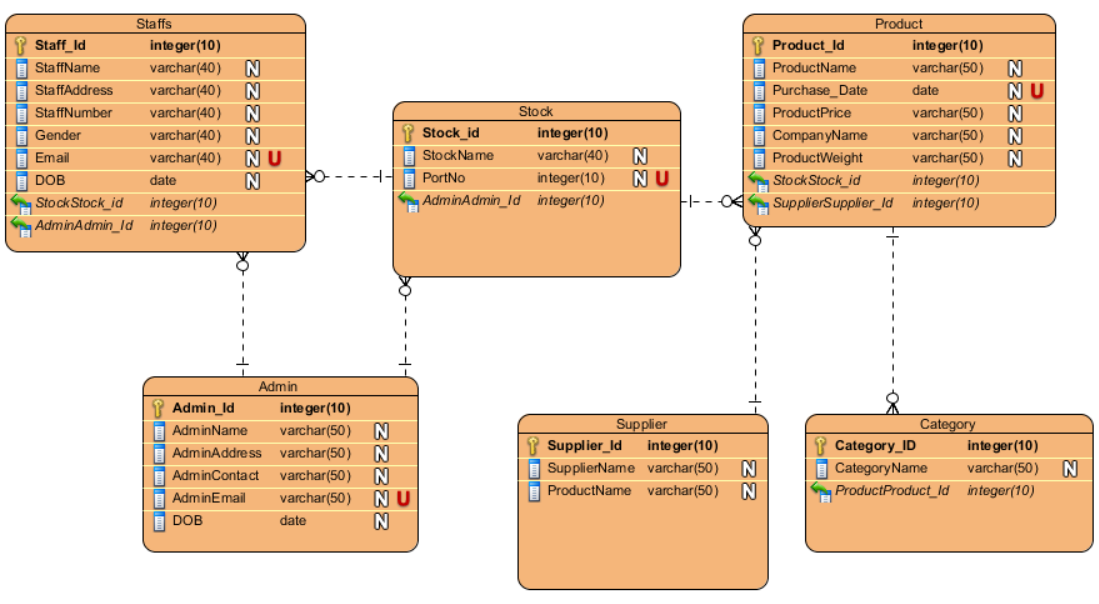


Figure 11: ER Diagram

**Description of the diagram**

Above Entity Relationship diagram contain six class or the Entity named Staff, Suppliers, Products, Admin, Stocks, and the category which illustrate the category of the products. Each and Every particular entity contain one unique primary key, and one unique key. They are connected with each other by one to many and many to one function. Suppose, Product have many category which contain one to many relation respectively.

## Data Dictionary:

Data dictionary is the Place where we used to keep the details of the content of data flow process and data store. It is data about data which define different element like data element. Data element is the smallest unit. It also contain Data structure which handle as a unit.

**Staff Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Data type** | **Length** | **Constraints** |
| 1 | Staff\_id | integer | 10 | primary |
| 2 | Staff Name | varchar | 40 | - |
| 3 | Staff Address | varchar | 40 | - |
| 4 | Staff Number | integer | 10 | - |
| 5 | Gender | varchar | 50 | - |
| 6 | Email | varchar | 50 | Unique |
| 7 | DOB | date | 0 | - |
| 8 | Stock\_id | integer | 10 | - |
| 9 | Admin\_id | integer | 10 | - |

Figure 12: Staff Data Dictionary

**Stock table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Data type** | **Length** | **Constraints** |
| 1 | Staff\_id | integer | 10 | primary |
| 2 | Stoke Name | varchar | 40 | - |
| 3 | PortNo | varchar | 40 | - |
| 4 | Admin\_id | integer | 10 | - |

Figure 13: Stock Data Dictionary

**Product Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Data type** | **Length** | **Constraints** |
| 1 | Product\_id | integer | 10 | primary |
| 2 | Product Name | varchar | 50 | - |
| 3 | Purchase Date | varchar | 50 | Unique |
| 4 | Product Price | integer | 10 | - |
| 5 | Product Weight | varchar | 50 | - |
| 8 | Stock\_id | integer | 10 | foreign |
| 9 | Supplier\_id | integer | 10 | foreign |

Figure 14: Item Type Data Dictionary

**Supplier table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Data type** | **Length** | **Constraints** |
| 1 | Supplier\_id | integer | 10 | primary |
| 2 | Supplier Name | varchar | 50 | - |
| 3 | Product Name | varchar | 50 | - |

Figure 15: Supplier Data Dictionary

**Supplier table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Data type** | **Length** | **Constraints** |
| 1 | Category\_id | integer | 10 | primary |
| 2 | Category Name | varchar | 50 | - |
| 3 | Product\_id | integer | 10 | foreign |

Figure 16: Category Data Dictionary

**Supplier table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Data type** | **Length** | **Constraints** |
| 1 | Admin\_id | integer | 10 | primary |
| 2 | Admin Name | varchar | 50 | - |
| 3 | Admin Address | integer | 10 | - |
| 4 | Admin Contact | varchar | 50 | - |
| 5 | Admin Email | varchar | 50 | - |
| 6 | DOB | date | 0 | - |

Figure 17: Admin Data Dictionary

## UI modelling (Prototype):

Prototype is used to describe the system logically before the development of the system. It helps to make the client easy to understand the system. It also helps to make the developer easy to develop the system. Prototype is known as the roughly drawn diagram to know the different interface of the system.

1. **Login Page:**

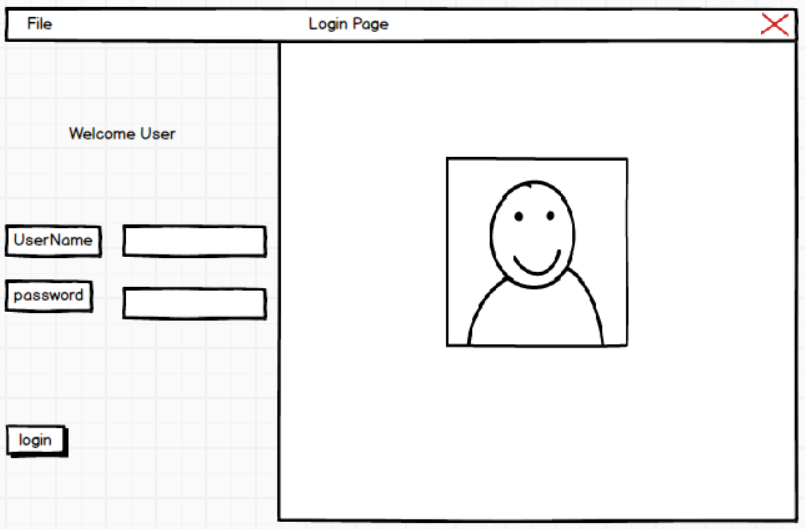


Figure 18: Login Page Prototype

1. **User Page:**

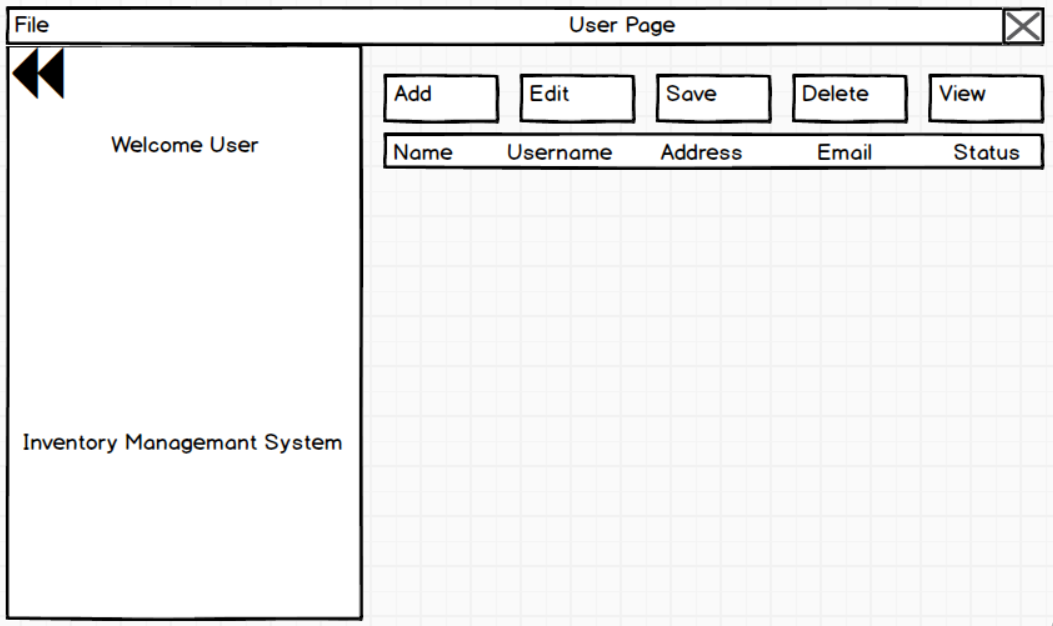


Figure 19: User Page Prototype

1. **Home Page:**

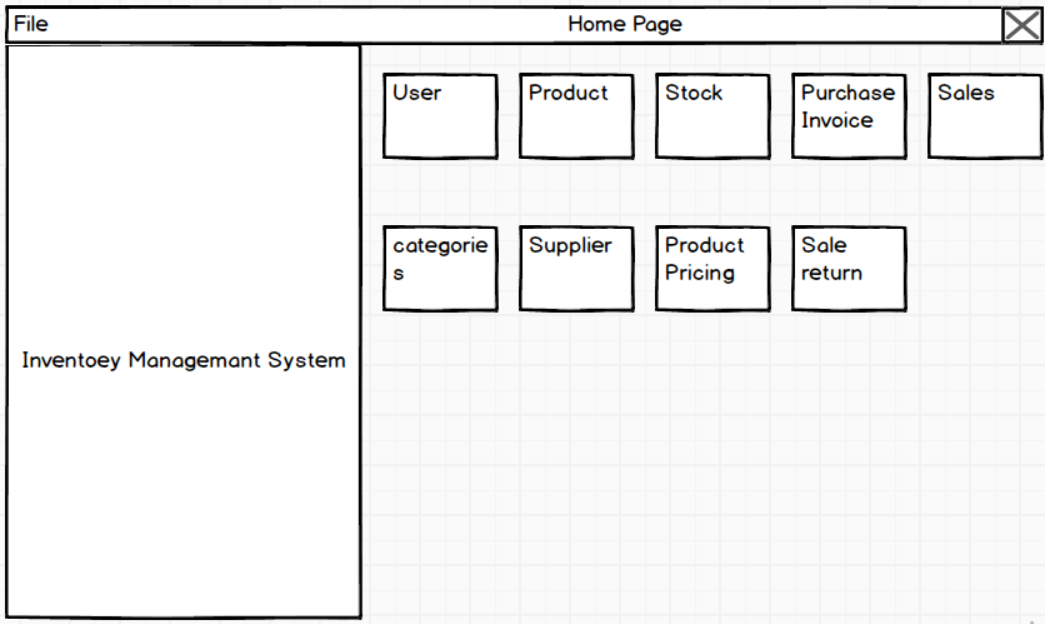


Figure 20: Home Page Prototype

1. **Sales Page:**

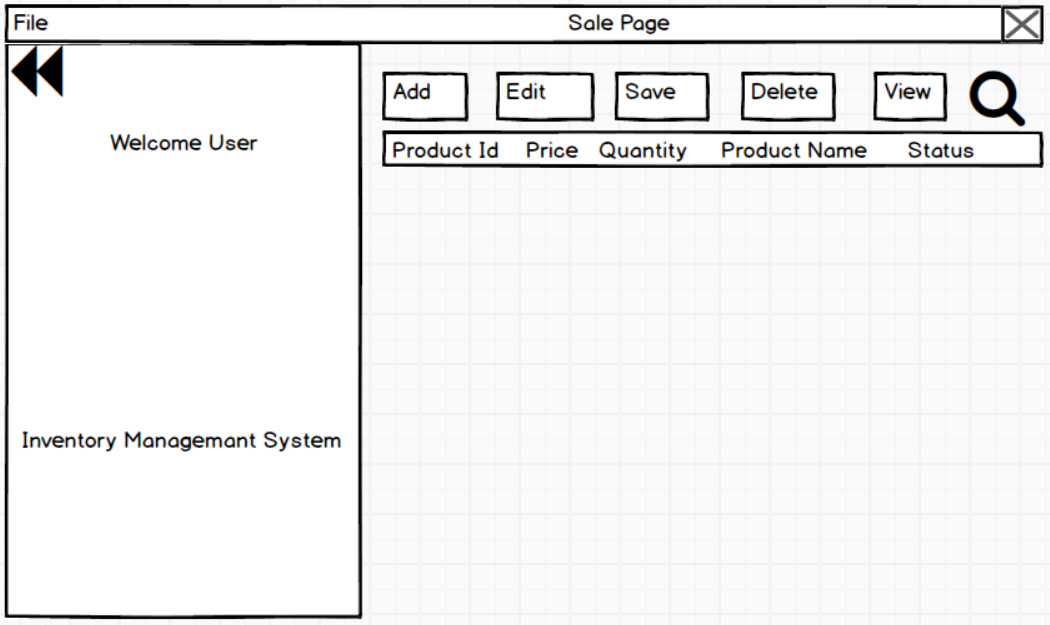


Figure 21: Sales Page Prototype

1. **Supplier Page;**

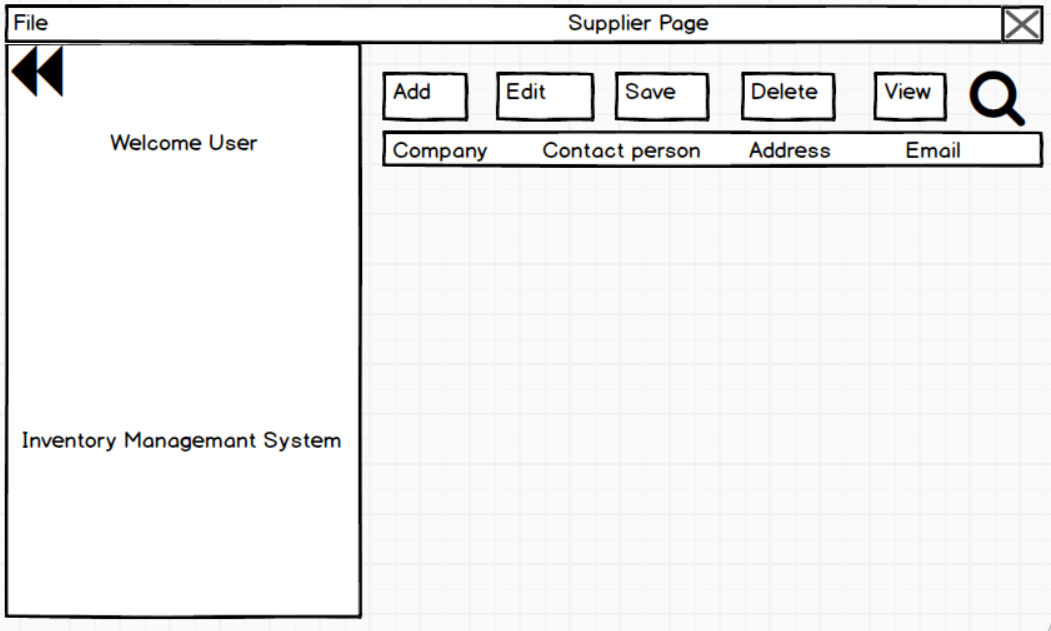


Figure 22: Supplier Page Prototype