** **

****











**Online Dairy products**

**By**

**Virat Tiwari**

**Pratik Singh**













**Acknowledgement**

It gives me great pleasure in presenting this project report. Its justification will never sound good if I do not express my vote of thanks to our C.H.M. College and respective Principal. I would also like to thank our HOD Mrs. Ritika Sachdev for her timely support in this completion of this project.

I thank our internal project guide Ms. Razia Khan*,* who has done a lot to keep this project systematically and on schedule. I am thankful for their valuable guidance to every stage of the project. Finally, I would thank to all others, who give there extended support to complete this project.





**B Project Documentation………………………………………………………………..47**











**ONLINE DAIRY**

**PRODUCTS**

Welcome to the world of Online Dairy Products, where convenience meets quality in the realm of fresh and wholesome goodness. In today's fast-paced lifestyle, the accessibility and reliability of obtaining dairy essentials have taken a significant leap forward. Our online platform redefines the way you experience dairy shopping, offering a diverse array of products that cater to your dietary preferences and nutritional needs, all from the comfort of your own home. From creamy milk and artisanal cheeses to velvety yogurts and decadent butter, we bring the charm of a traditional dairy aisle right to your digital doorstep. Join us on a journey where traditional flavors blend seamlessly with modern convenience, as we deliver the richness of dairy products directly to you.

Dairy products are a great source of protein, calcium, and other nutrients. They are also a versatile ingredient that can be used in many different dishes. In the past, it was necessary to go to the grocery store to buy dairy products. However, online shopping has made it possible to buy dairy products from the comfort of your own home.

There are many benefits to buying dairy products online. First, you can save time by not having to go to the grocery store. Second, you can choose from a wider variety of dairy products than what is typically available in grocery stores. Third, you can often find better prices for dairy products online. Finally, you can have dairy products delivered to your home, so you never have to worry about running out.





**Objective of Project on Online Dairy Products :**

The objective of this project is to develop an online platform for buying and selling dairy products. The platform will allow users to browse a wide variety of dairy products from different Company, compare prices, and place orders. The platform will also provide users with information about the dairy products .

The main objective of the project on ONLINE DAIRY PRODUCTS is to manage the details of Product, Category, Place Order, Confirm Order, Add to Cart, Online Paymrnt etc..

To provide users with a convenient and easy way to buy dairy products online.



* The Current System can be extended to allow the users to create accounts and save products in to wish list and also add address.
* The users could subscribe for price alerts which would enable them to receive message when price for products fall below a particular level.
* The current system is confirmed only to the shopping cart process. It can be extended to have a easy to use check out process.
* Development of an online platform for buying and selling dairy products.
* The platform will allow users to browse a wide variety of dairy products from different compare prices, and place orders.



**Economic Feasibility :** The project is economically feasible as the only cost involved is having a computer with the minimum requirement mentioned earlier. For the users to access the application, the only cost involved will be in getting access to the internet.

**Technical Feasibility :** To deploy the application the only technical aspects needed are mentioned below –

**Platform :** HTML, CSS, JS, Bootstrap, Django

**Database :** MYSQL

**For Users:**

Internet Browser

Internet Connection

**Behavioral Feasibility :** The application requires no special technical guidance and all the views available in the application are self explanatory. The users are well guided with warning and failure message for all the actions taken.



**Hardware Specification**:

**Processor** : 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz

**RAM** : 8.00 GB

**Minimum Space required :**  1GB

**System Type :** 64-bit operating system, x64-based processor

**Software Specification:**

**Platform :** HTML, CSS, JS, Django

**Database :** MYSQL





* Admin log into the system.
* Admin add/update/delete details of events.
* Admin view the booking status.
* Admin generates the receipt.
* Admin view reports of event.
* New customers register into the system.
* Register customer log into the system.
* Customers search the products.
* Customer book an event.
* Customers pay the payment.
* Customers give the feedback.



* **Increased convenience for users:** Users will be able to buy dairy products from the comfort of their own homes, without having to go to the any store. This will save them time and effort.

* **Wider variety of products:** Users will be able to choose from a wider variety of dairy products than what is typically available in any stores. This includes products from different brands, countries, and types.

* **Better customer service:** Online retailers can typically offer better customer service than any stores. This is because they have more control over the customer experience and can be more responsive to customer feedback.
* **Increased demand for dairy products:** The dairy industry as a whole will benefit from the increased demand for dairy products through online retailers. This will lead to increased production and employment in the dairy industry.

Overall, the expected outcomes of a project on online dairy products are positive for both users and suppliers. It is a project that has the potential to make dairy products more convenient, affordable, and accessible to a wider audience.





<https://getbootstrap.com/><https://cdnjs.com/libraries/jquery><https://razorpay.com/>

# 11. Entity Relationship Diagram (ER-Diagram)

* 1. Entity relationship diagram can express overall logical structure of database logically.
* 2. ER Diagrams are simple and clear.
* 3. ER Diagrams represents entities and tables and their relationship with one another.

Components of ER diagram:

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. |  | Shape | Description |
| 1 | Rectangle |  | Represents entity set. |
| 2 | Ellipse |  | Represents attributes. |
| 3 | Diamond |  | Represents relationship. |
| 4 | Flow lines |  | Represents link between 2 entities set. |
| 5 | Double ellipse |  | Represents multivalve attributes. |
| 6 | Dashed ellipse |  | Denotes derived attributes. |
| 7 | Double Rectangle |  | Represents weak entity set. |
| 8 | Double Diamond |  | Represents relationship set for weak entity set. |

User Name

**Login:**

Password

Login

**Dairy Products:**

Milk type

product code

Event place

Status

Rate

Images

Dairy product

**Recept:**

Recept

Dairy Product

Product Type

payment

Case

Card

User Name

Email ID

Item

Total price

**Customer:**

Customer

Locality

Name

Email ID

Mobile No.

State

Address

City

Pin Code

Customer

Bill

Admin

Dairy product

Login

Pay

Order

Add

Manages

As

Generate

Alert

By

View

As

Reports

# 12.Use Case Diagram

use case description

**Admin Subsystem:**

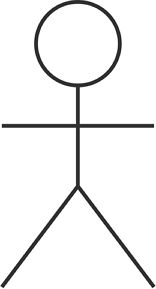
* Admin log into the system.
* Admin add/update/delete details of Products.
* Admin view the order status.
* Admin generates the receipt.
* Admin view reports of item.

**Customer Subsystem:**

* New customers register into the system.
* Register customer log into the system.
* Customer Order a Product.
* Customers pay the payment.
* Customer receive Ordered Milk Product Receipt
* Customers give the feedback.

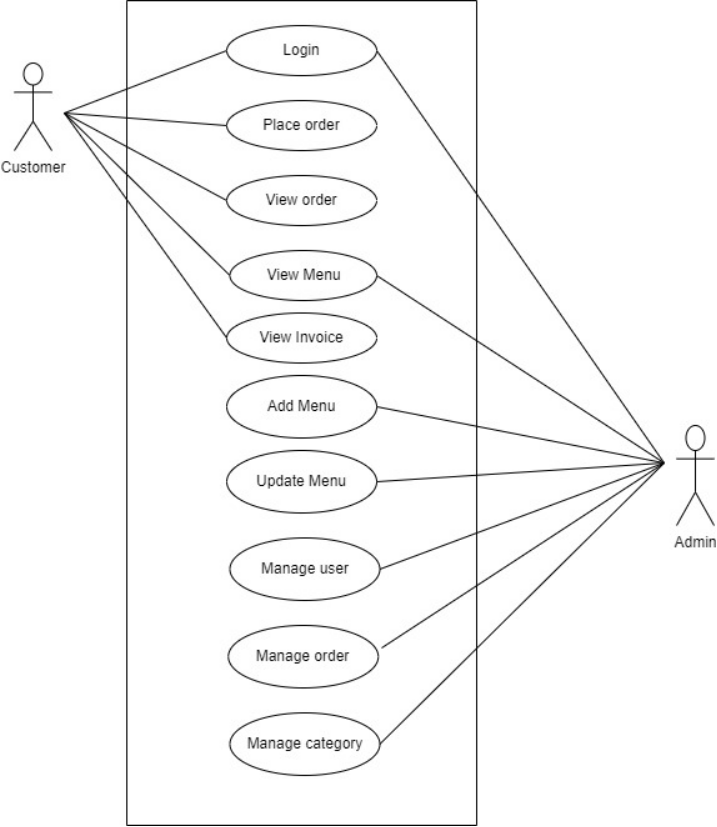
use case diagram

A use case diagram is a set of scenarios that describing an interaction between user and system. A use case diagram displays the relationship among actors & use cases. The 2 main components of use case diagram are use case and actor.

Actor Use Case

An actor represents a user or another system will interact with the system that you are modelling. A use case is an external view of the system that represents some action that might perform in order to complete a task.



13.Activity Diagram

● Activity diagram describes the workflow behavior of the system.

● 2. Activity diagram are similar to state diagram because activities are the state of doing something. ● The diagram describes the state of activities by showing the sequence of activities performed.

● Activity diagram can resolve activities that are conditional or parallel.

●Activity diagram should be used in conjunction with other modeling techniques such as interaction diagram and state diagram.

● Main reason to use the activity diagram is to model workflow behind the system being designed.

**Components of Activity Diagram:-**

|  |  |  |
| --- | --- | --- |
| 1 |  | Represents the activity. |
| 2 |  | Merge |
| 3 |  | Fork |

view

Details

Add to Cart

Order decorative

items

Payment

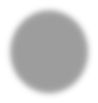
Verification

Cancel

Order

Receive

Receipt



New Register

Login

No

Add New

product Type

Add

New product

Update R

ecord

View

Order

Generate Receipt

Delete Records

No

Login

Ye

s

Admin

User

## 14.Class Diagram

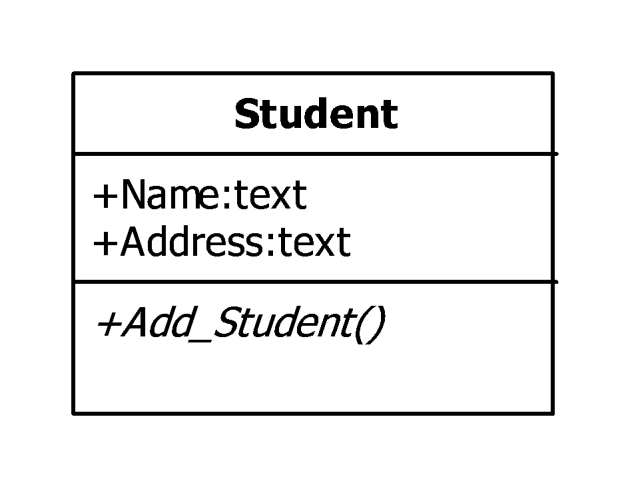
Class diagrams are widely used to describe the types of objects used in system and their relationship. Class diagram models class structure and contents using design elements such as classes and packages and objects.

Class diagram describes 3 different perspectives when designing a system. These perspectives become evident as the diagram is created and help solidify the design.

Classes are composed of 3 things:

1. Class name.
2. Attributes and
3. Operations.

**For Example Diagram:**



**Class Name**

**Attribute**

**Operations**

14.Class Diagram

Class diagrams are widely used to describe the types of objects used in system and their relationship. Class diagram models class structure and contents using design elements such as classes and packages and objects.

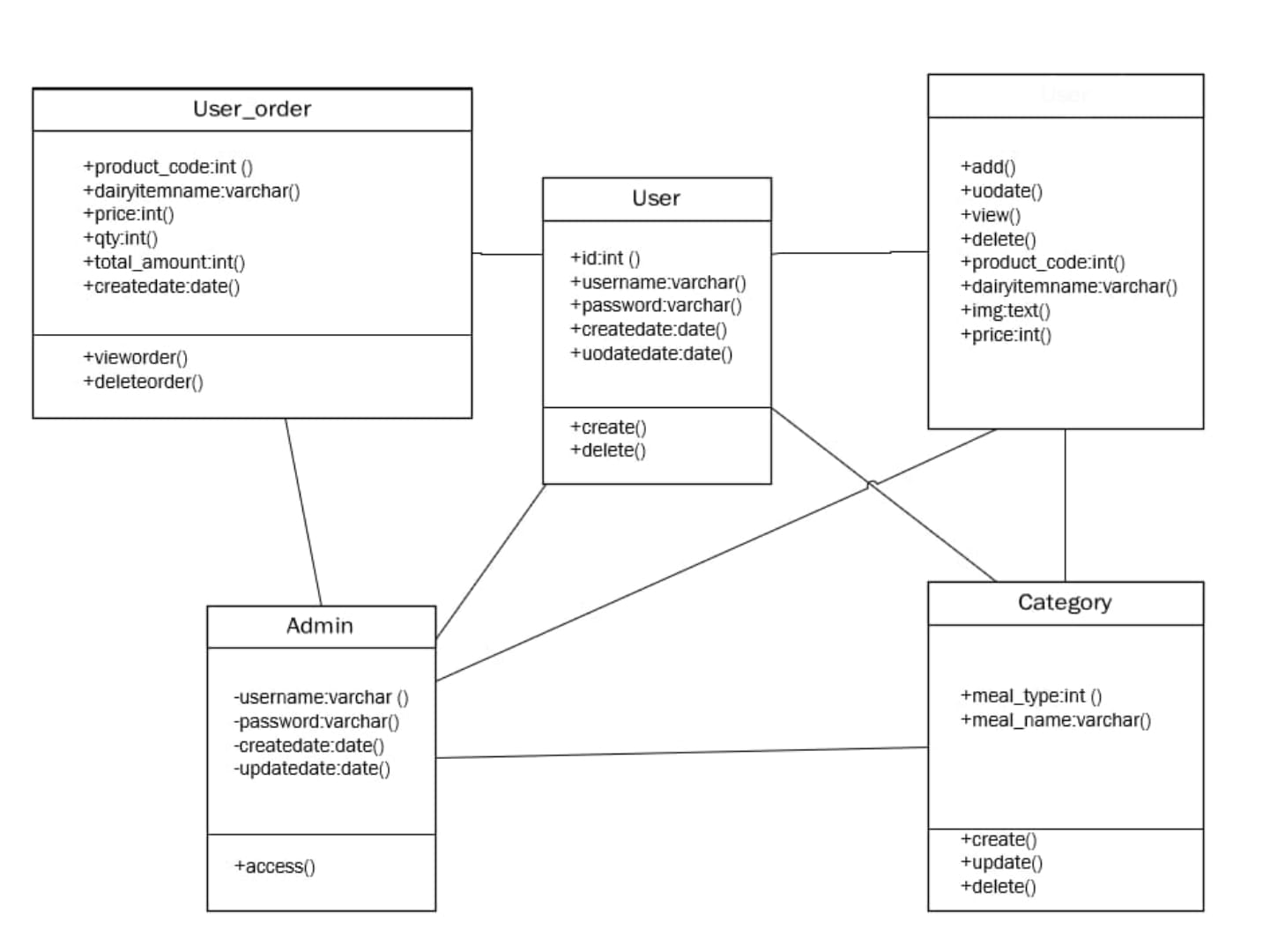
Class diagram describes 3 different perspectives when designing a system. These perspectives become evident as the diagram is created and help solidify the design.

Classes are composed of 3 things:

1. Class name.

2. Attributes and

3. Operations.



Dairy item

## **15.Object Diagram**

## 

Object diagram are same as that of class diagram. Instead they contain the values in place of data types.

The object diagram describes 3 different perspectives when designing a system. This perspective becomes evident as the system is created & helps solidify the design.

They are composed of 2 things:

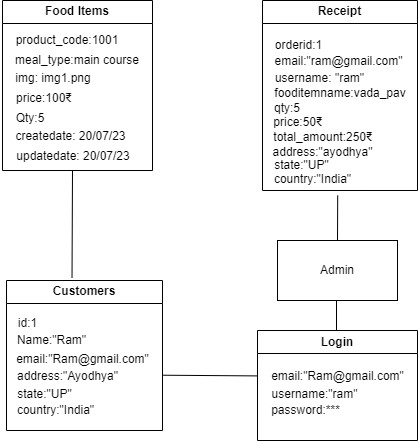
1. Class name.
2. Attributes.

For Example Diagram:



**Class Name**

**Attribute**



**Dairy item**

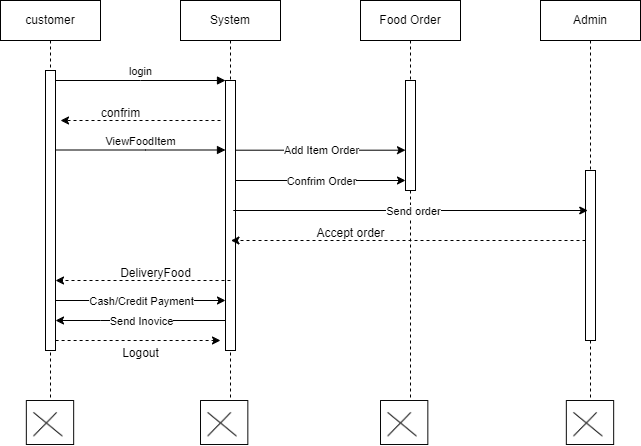
## **16.Sequence/Collaboration Diagram**

Sequence diagram demonstrates the behavior of the objects in a use- case by describing the objects and the messages they pass. The diagrams are read left to right & descending.

**Components of Sequence Diagram:-**

|  |  |  |
| --- | --- | --- |
| 1 |  | Represent object activation. |
| 2 |  | Represents the objects of case study. |
| 3 |  | Represents life of objects. |
| 4 |  | Represents end of objects. |

Product Order



View dairy item

## **17.State Diagram**

* State diagram are used to describe the behavior of the system.
* State diagram describe all of the possible state of an object as an event occurs.

* Each diagram usually represents objects of single class and track the different state of its object through the system.
* We use the state diagram to demonstrate the behavior of an object through many use cases of the system.
* We use the state diagram for classes where it is necessary to understand the behavior of the object through the entire system.

**Diagram:**

State Diagram

Activity

# Transitions

**Customer**

Order a Dairy item

View

product

Details

Order a décor item

View

Order

Pay the Bill

Payment Verification

Cancel

Order

Receive

Receipt



New Register

Login

End

No

Yes

**Admin:**

Add New Dairy Type

Add New

Decor\_Type

Add

New Décor item

Update Records

View

Order

Generate Receipt

Delete Records



End

No

Login

Yes

## 18.Component Diagram

* The component diagram contains component & dependencies.
* Component represents the physical packaging of the module of the code.
* The dependencies between the components show how changes made to one component may affect each other components in the system.
* Dependencies in a component system are represented by dashed lines between 2 or more components.
* Component diagram can also show the interface used by the component to communicate with each other.

**Diagram:**

method1

()

+

()

method2

+

Object1

+

method1

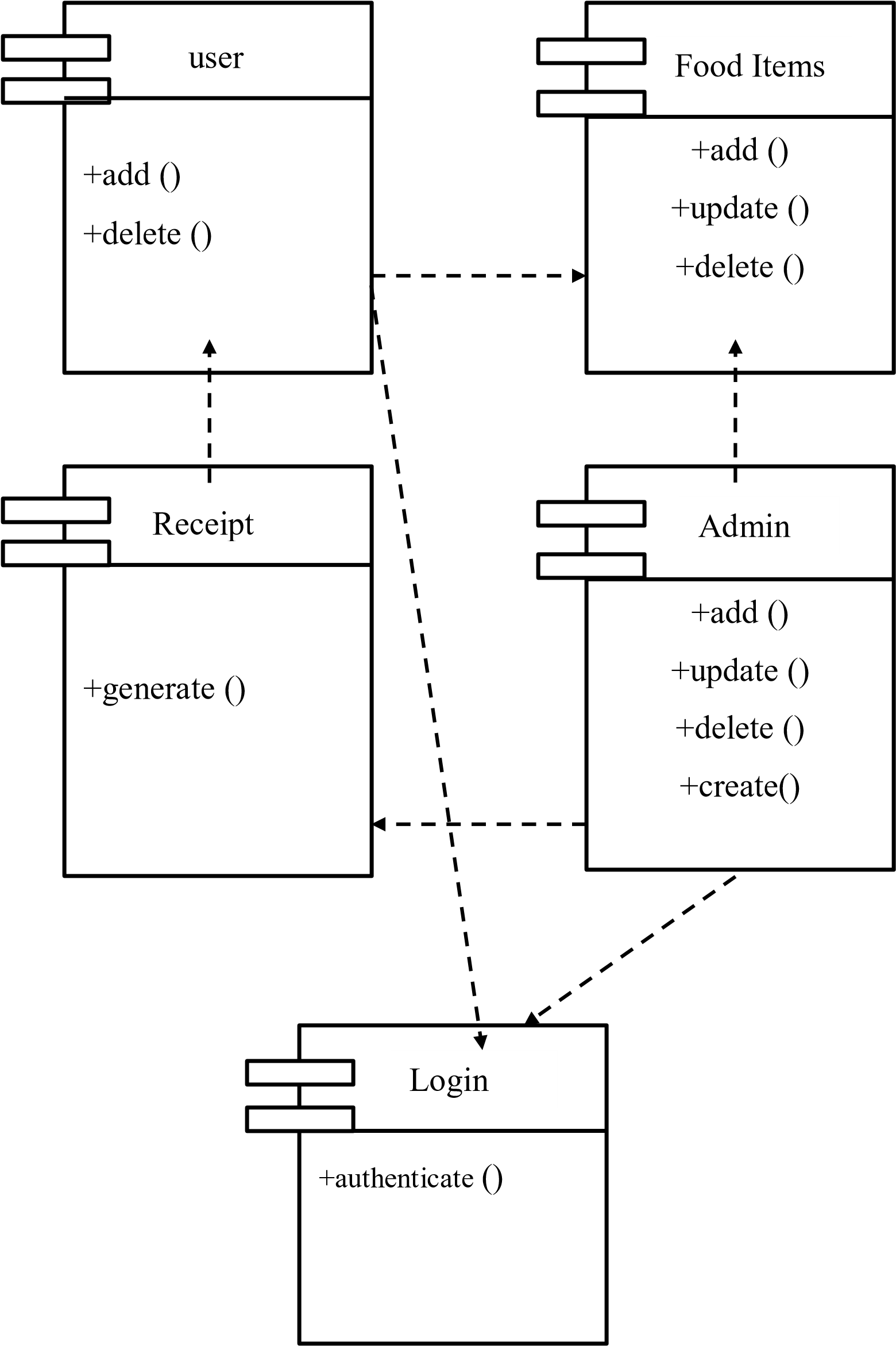
()

+

method2

()

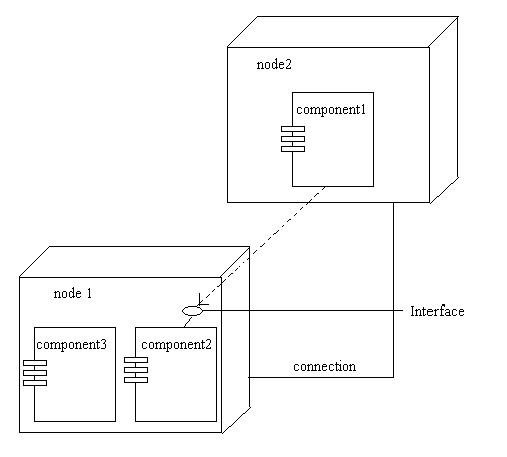
Object2

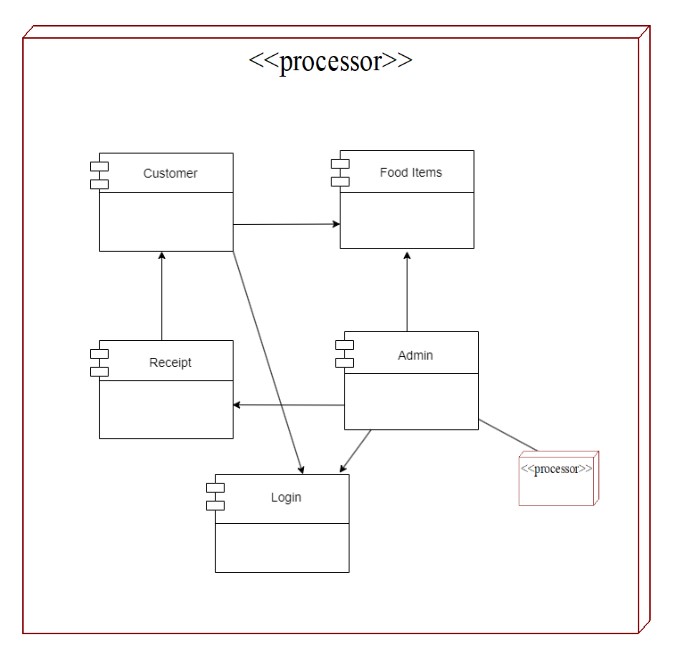


Dairy

## **19.Deployment Diagram**

* The deployment diagram contains nodes & connections.
* A node usually represents a piece of hardware in the system.  A connection depicts the communication path used by the hardware to communicate & usually indicates a method such as TCP/IP.
* **Diagram:**





Dairy item

**20 Package Diagram**

• In addition to standard UML dependency relationship there are two special types decencies defined between packages:

Package Import

Package Merge

• A package imports a relationship between an import namespace and a package indicating that importing namespace adds the names of members an unlabeled dependency between two packages an interpreted as a package import relationship. In this relationship elements within the target package will be imported in source package.

• A package merge is a directed relationship between two packages that indicates that the contents of two packages that are to be combined. It is very similar to generalization.

**Elements of package Diagram:**

1. Package: It is a general purpose mechanism for organizing model contains elements is designed diagrams into groups. It provides an encapsulated namespace within which all names must be unique. 2. Class: It is representation of objects that reflects their structure and behavior of system.

3. Interface:It is specification of behavior. By implementing interface classes are required to support the behavior

4. Object: It is instance of class. It often used in analysis to represent numerous artifacts and item that exist.

5. Table: It is stereotyped class.

Dairy item

Online Dairy Products

Customer

Decor

Items

Receipt

Admin

Login

Database

## **21.Menu Tree**

**Login:**  Create new account

**Main:**

* Customer
* Admin

**Activities:**

* Order a dairy products
* View dairy Items
* Register Customer
* Dairy products Details
* Ordering Report
* Receipt Report
* Sign up
* Log in
* View Dairy Order Information
* Order a dairy producs
* Exit

**22 Database and List of Tables with descriptions**

Database: django

Name of the database: Online dairy product

1. **Table name: User**

Description: Stores the information of the account created by the Admin and Author.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data Type** | **Description** |
| user\_id | int | id generated for user |
| first\_name | varchar | first name of user |
| last\_name | varchar | last name of user |
| username | varchar | unique user name created by user |
| password | varchar | password created by user |
| role | int | Role of admin user |

1. **Table name: category**

Description: Stores the information of the category .

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data Type** | **Description** |
| category\_id | int | Unique Id of the category is generated |
| category\_name | varchar | Name of the category i.e sports,GK,etc |
| post | int | Images posted by the author/admin |

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**

**File Name:.**

**Source Code:-**