# Chapter 3 Design

## 3.1 Introduction to design

**Project design is an early phase of the project where a project’s key features, structure, criteria for success, and major deliverables are all planned** out. The point is to develop one or more designs which can be used to achieve the desired project goals. Stakeholders can then choose the best design to use for the actual execution of the project.

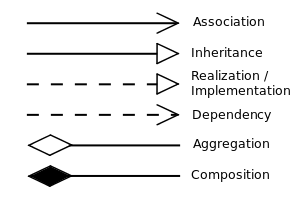
Different design models are used so that easy to understand of that user requirements. For this project I have structural and behavioral models and to flow the system have to done database design and user interface have to done in this phase

## 3.2 STRUCTURAL MODELLING

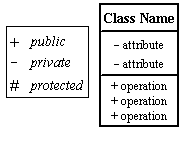
Structural modelling is general solution of application to be developed. It gives solution for the visibility of data being passed. It specifies memory requirement and debugging functions. It declares data, variable, method and their label of privacy like privatize or public or secured protection.

### 3.2.1 CLASS DAIGRAM

A **class diagram** is a type of **diagram** and part of a unified modeling language (UML) that **defines** and provides the overview and structure of a system in terms of **classes**, attributes and methods, and the relationships between different **classes**.



* Aggregation is a special type of **association** in which objects are assembled or configured together to create a more complex object.
* Association end is a connection between the line depicting an association and the icon depicting the connected classifier.
* **Inheritance** is shown as an open arrow between the **diagrams**.
* **Dependency** is a **directed** relationship which is used to show that some UML element or a set of elements requires
* **Composition**. The UML representation of a **composition** relationship shows **composition** as a filled **diamond** shape on the containing **class** end of the lines that connect contained **class**(es) to the containing **class**.



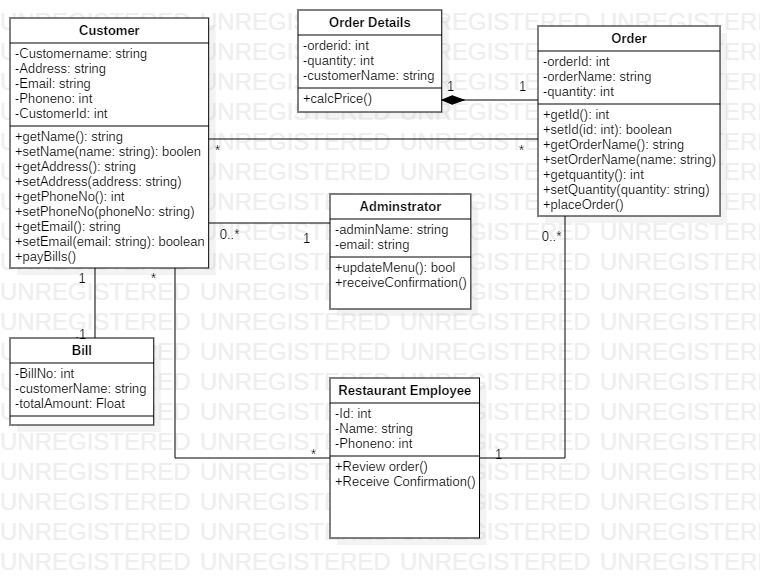
* In this symbols first have class name and their attributes and their operation.(+) public the operation and (-) private attribute.
* To hide anything from people have to use (-) this symbols.

0…\*

* This means one to many.

1-1 = one to one relation

\*--\*= many to many relations



Justification

Class diagram is the most important architectural diagram. In this diagram have to show the relationship between classes. And have to show the connection the classes among them. Class diagram helps to make our work easy. There are many symbols and many meaning of them.

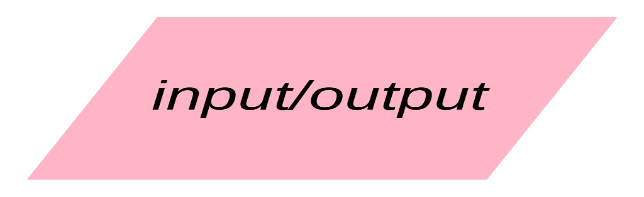
### 3.2.2 FLOW CHART

A **Flowchart** is a graphically representation of the structure of process or system, algorithm or the step-by-step solution of the problem. The **Flowchart** describes the flow of data through an information processing systems and the parts of the flows. The flow is a set of the logic operations that meet the certain requirements.

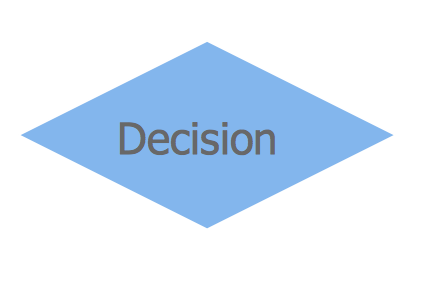
NOTATIONS

1. 

This is the notation of starting and ending point.



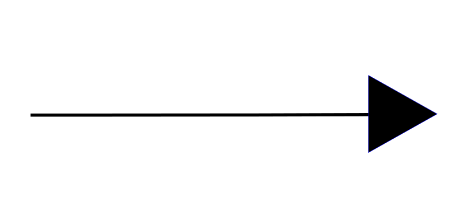
Input/output symbols.



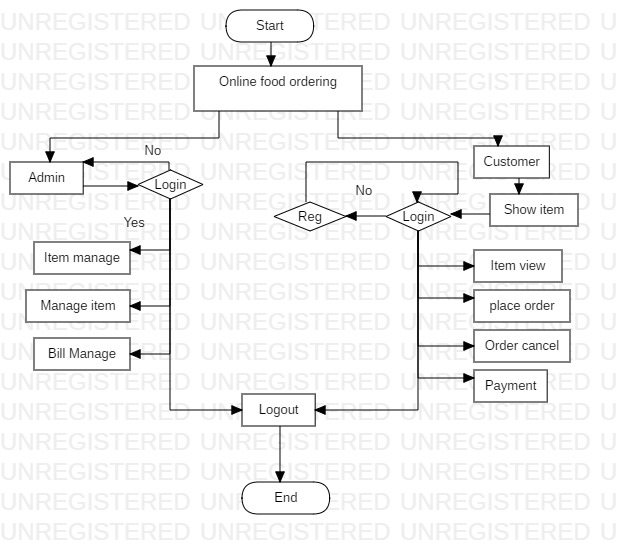
Decision symbols



This is module symbol.



Arrow which helps to connect each other in the flow chart.



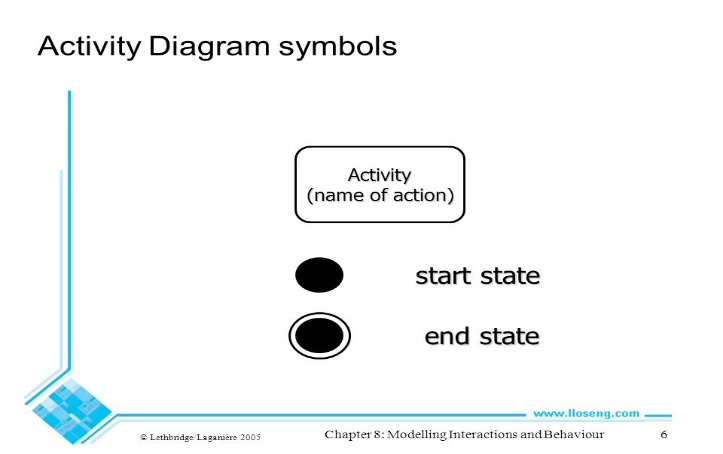
Justification

# 3.3 BEHAVIORAL MODELLING

Behavioral Modelling is the modelling which deals with functionality of entire entity of system. All statements of the system are drafted sequentially with process statement and defines all responses of system.

## 3.3.1 ACTIVITY DIAGRAM

Activity is a parameterized behavior represented as coordinated flow of actions. The flow of execution is modeled as activity nodes connected by activity edges. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents. Activity nodes also include flow of control constructs, such as synchronization, decision, and concurrency control.

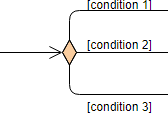


starting and ending node and action symbol

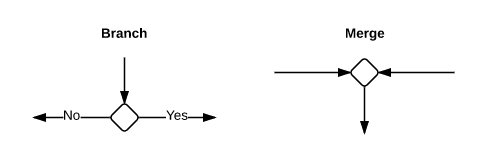
* Final flow node

See the source image

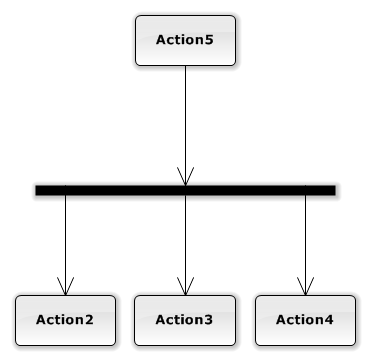
* Decision



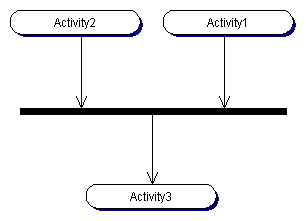
* Merge Node



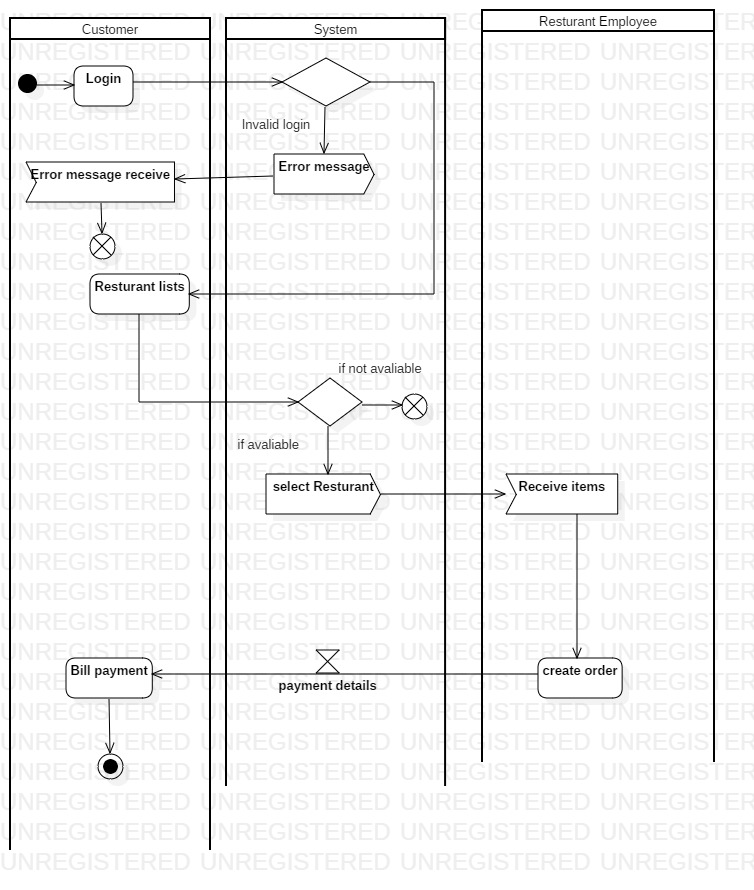
* Fork



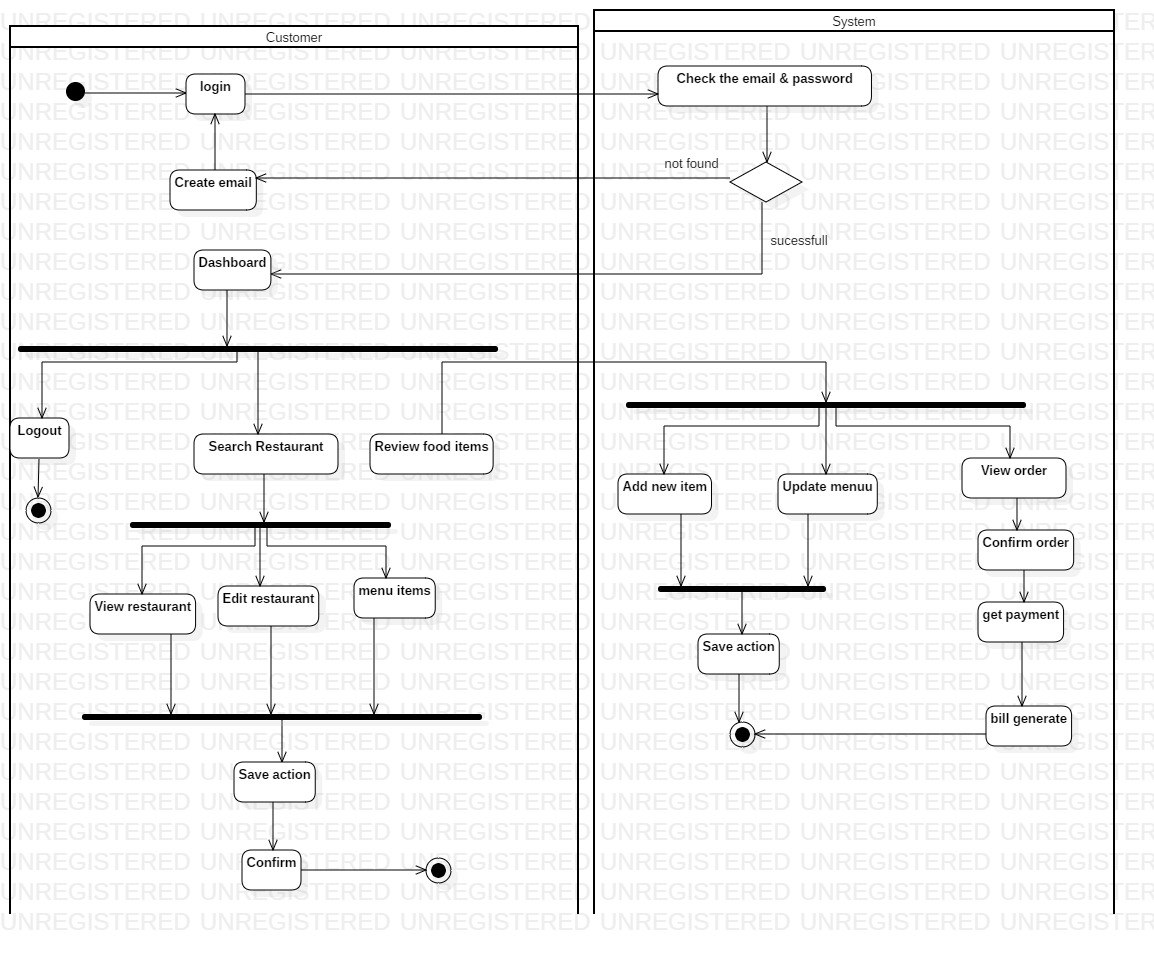
* Join



* Control flow which connect all the actions.



This is the first activity diagram of customer, system and restaurant employee. In this diagram show what is happened when we order the online food and show what is happening in the system. This diagram is important in the project because it helps to know what would be done and what have done in the system. Easy to know about the system.



Justification

This the figure of customer and system. and in this figure shows what the customer and system have rights to do. I have made this figure in the starUML. There are many symbols which have different meaning of them. There are two swim lanes representing customer and system.

## 3.3.2 SEQUENCE DAIGRAM

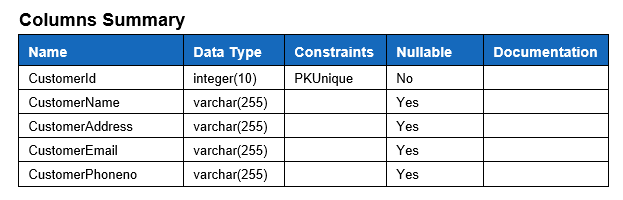
UML **sequence diagrams** are used to show how objects interact in a given situation. An important characteristic of a **sequence diagram** is that time passes from top to bottom : the interaction starts near the top of the **diagram** and ends at the bottom (i.e. Lower equals Later). A popular use for them is to document the dynamics in an object-oriented system.

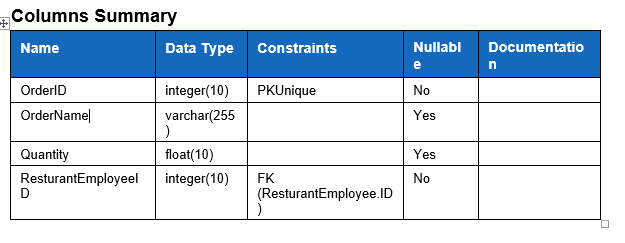
# 3.4 DATABASE DESIGN

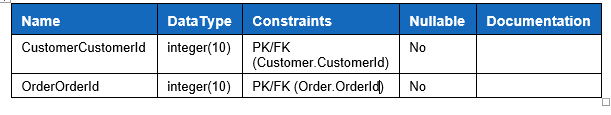
**Database Design** is the process of structuring your **database** in a way that will reduce errors, incorrect data, and redundant **data**. This will give us an optimized, fast, safe, reliable **database**. **Database design** largely implements relationships and normal forms.

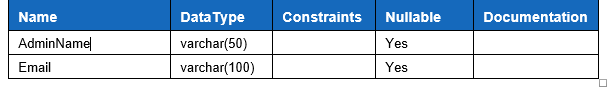
## **3.4.1 DATA-DICTIONARY**

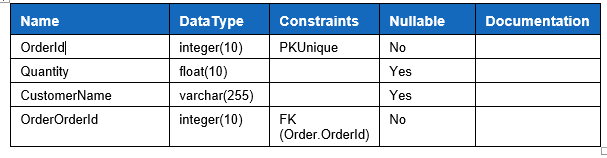
**Data dictionary is record of object of database and their relationships to be implemented. Database administrator interact with data dictionary i.e. no any person would be able to view it while it is crucial component of database and its security. It contains all name of data, data type, constraints and null or not null. The constructed database for my database are as below.**

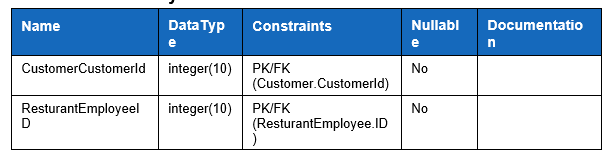


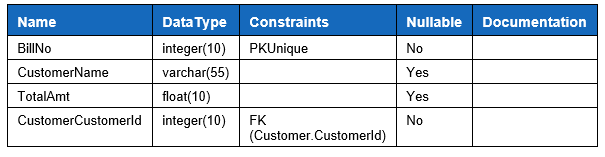


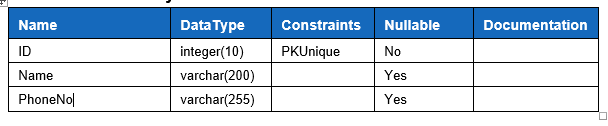












So, these are data dictionary of my project.

## 3.4.2 ENTITY RELATIONSHIP DAIGARM

An entity relationship diagram shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

Key components are:

* Entity:

Entity defines the object that exists. It can be predicted as value like any id or may be any possible object username, name, passwords, date of birth or any specification that can be inserted as attribute information in database within

* Attributes:

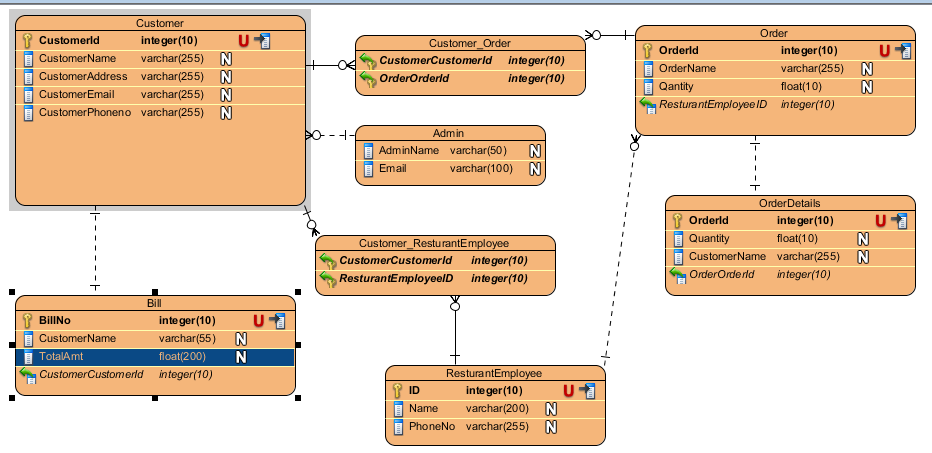
In the term of DBMS, attribute refers database component possessing description of entity. Username, user ID can be taken as example of attribute.

* Relationship:

Different table are associated referencing primary key of related tables as foreign key.

Three types of relationships they are:

* One to one relationship
* One to many relationships
* Many to many relationships



This the ER diagram of my project. This diagram made in visual paradigm. Here the entities are shown along with the relationships that they have with each other.

## 3.5 USER INTERFACE DESIGN

The goal of **User Interface** (**UI**) **design** is to anticipate what users might need to do and ensures that the interface has elements that are easy to access, understand, and use to facilitate those actions.