

## SW2 – Project Evaluation Form

- Each team must submit the following Documentation that contains:
  - Project Description in detail.
  - Class Diagram. And Database Schema.
- Each team must submit the project via GitHub:
  - Source Code.
  - Video Demo for running (2 – 5 Minutes).
  - Documentation and Evaluation Form
- The Evaluation will start with giving all teams 30 marks then check the following criteria:

Violation Level	Full	Medium	Small	Grade
Documentation	-5	-2	-1	
Not Apply MVC (it does not Separate Business logic from GUI). Example of violation: write the implantation for a method such as an inset item into the database inside the Button Action method)	-6	- 3	-1	
Violate clean code – Variables	-2	-1	-.05	
Violate clean code – Functions	-2	-1	-.05	
Violate Single-responsibility Principle	-2	-1	-.05	
Violate Open-closed Principle	-2	-1	-.05	
Violate the Liskov Substitution Principle	-2	-1	-.05	
Violate Interface Segregation Principle	-2	-1	-.05	
Violate Dependency Inversion Principle	-2	-1	-.05	
Not Upload code to GitHub	-1			
Only One Branch Without Merge (GitHub)	-2			
Only One Contribution (GitHub)	-2			
Total Minus from Grade				
Design Pattern Bounce	+4			
Bounce on Overall Work	+2			
Total Team Grade / 30				

Name (Arabic)	ID	Individual Bounce +2	Grade	Grade
محمد احمد رفاعي عبدالحليم	201900622			
محمد علي محمود محمد	201900712			
مازن ربيع عبدالحميد ربيع	201900607			
محمد اسامه محمد سلامه	201900630			
مصطفى محي الدين مصطفى احمد	201900834			
امنيه محمد احمد محمد	201900176			

# **Software2 Document**

**Version 1.0**

## **Resturant Management System**

---

Helwan University

Faculty of computers and artificial intelligence

**CS352 Software Engineering- 2**

**DR. Ahmed Hisham**

### **1. Introduction**

#### **1.1. Purpose**

The purpose of this document is to present a detailed description of the restaurant management system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

---

## **1.2. Scope of project**

It has become easy to add employee and work on system and keeping the accuracy of restaurant .

These systems will be available to all employees to communicate with the owner,

The system will be designed to work to organize the restaurant and to get data about employees or meals and get reports easily , improve the system

and to provide the ability to communicate with the owner.

## Glossary

Terms	Definition
Database	Collection of all the information monitored by this system.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document
Casher	The person who has access to system and he can (sign out-sign in) to all stuff And take orders
Stock Man	The person who has access to system and transfer goods to kitchen stock
Stuff	The person who has an account and access to system
Manger	The person who has access to modify meal, (add – update – delete) staff, (add -update – delete) meal

### 1.3. Overview of document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the system. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the system.

Both sections of the document describe the same software system in its entirety, but are intended for different audiences and thus use different language.

## **3.2. non-Functional requirements**

### **3.2.1. Implementation requirements**

In implementing whole system, it uses c# language which will be used for database , the database part is developed using MySQL.

---

### **3.2.2. Reliability requirements**

The system should accurately perform accommodation registration, user validation.

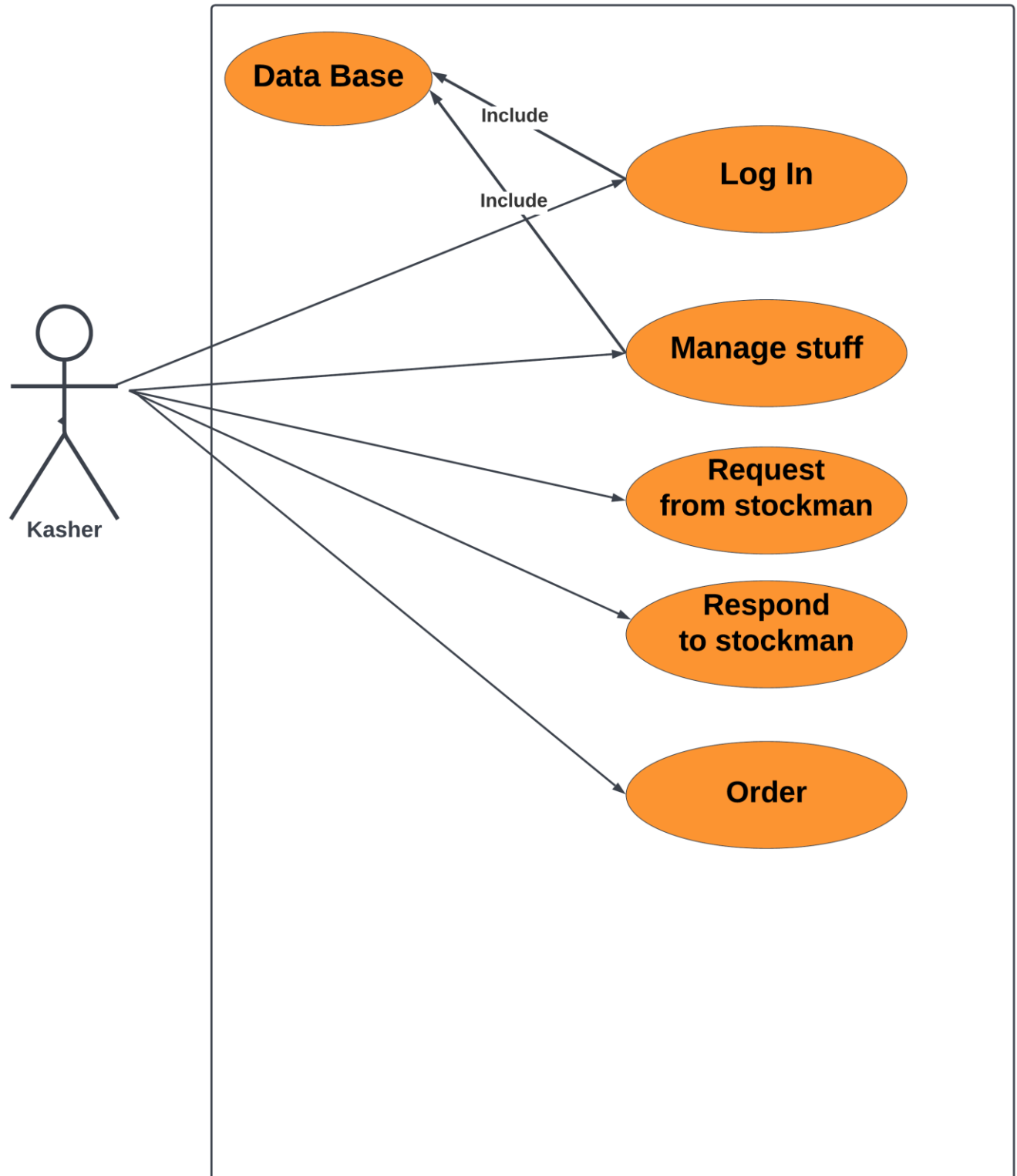
---

### **3.2.3. Usability requirements**

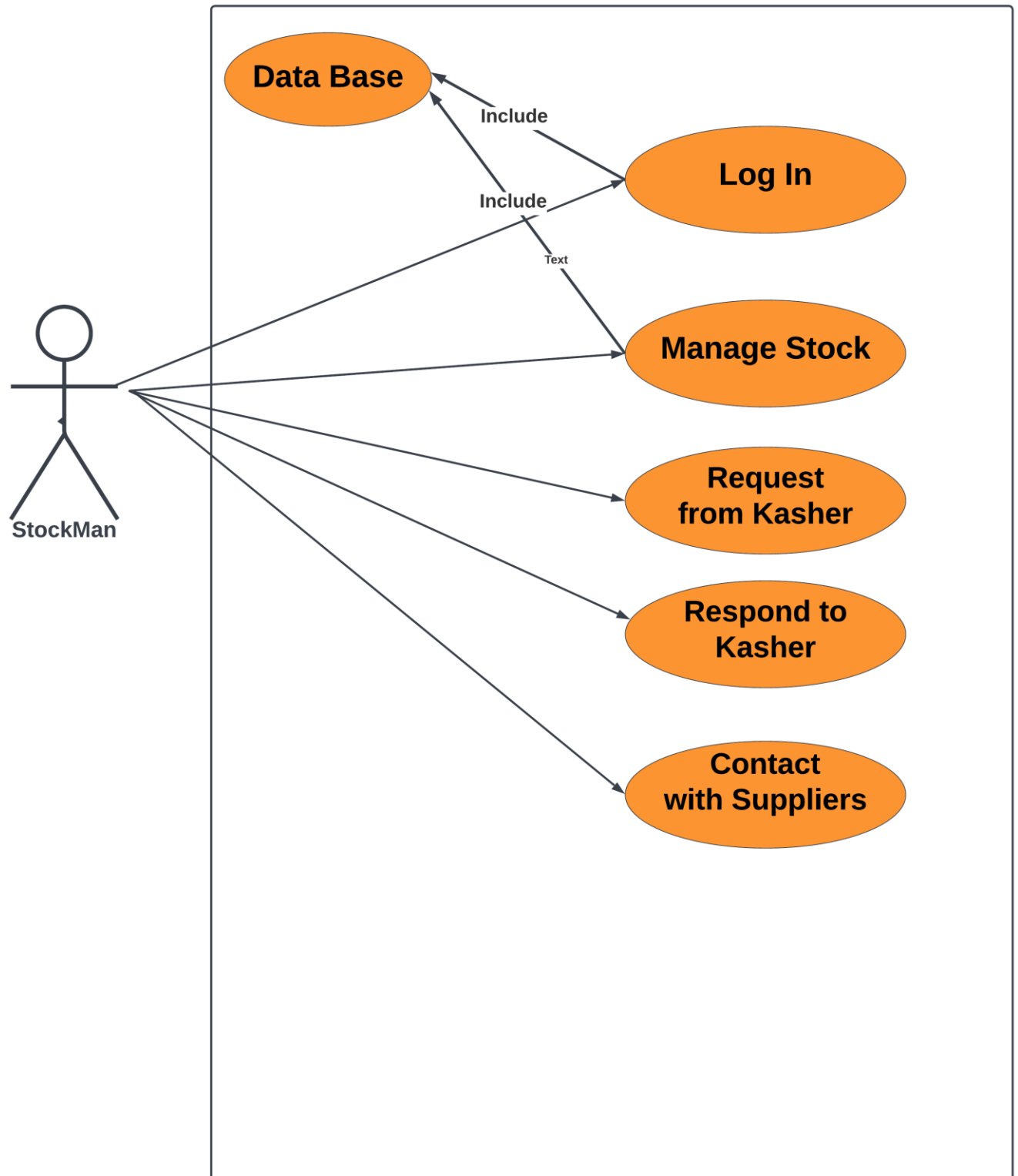
The system is designed for employees and Administration to work easily and faster and keeping accuracy of work

# Use case:

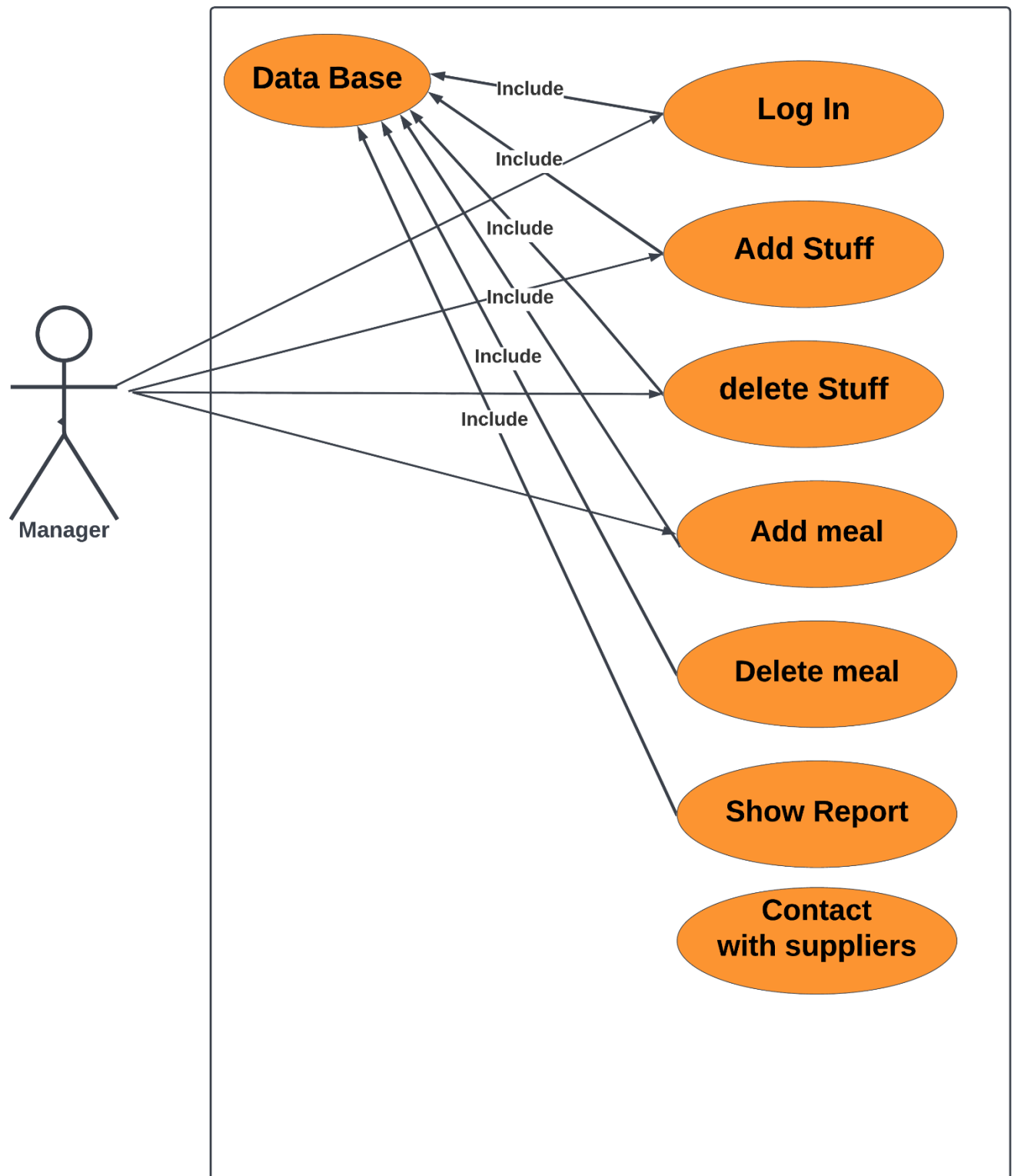
Casher:



Stock Man:



### 3- Manger





# Use case Description:

## 1- Casher

Use Case Name	Log in
Precondition	Must have account
Basic Path	1- Casher login to system 2- System check the account 3- Casher go to casher dashboard
Alternative Paths	No alternative
Postcondition	Casher go to casher dashboard
Include	Database

Use Case Name	Mange staff
Precondition	Casher must have login
Basic Path	1- Casher enter to label of staff 2- Casher mane log in of staff 3- Casher mange logout of staff
Alternative Paths	No alternative
Postcondition	All staff logout from system after every day
Include	Database

Use Case Name	Request from stockman
Precondition	Casher must have login
Basic Path	1- Casher enter to label of request 2- Casher send request to stockman
Alternative Paths	No alternative
Postcondition	Casher send request to stockman
Include	

Use Case Name	Respond to stock man
Precondition	Casher must have login
Basic Path	1- Casher enter to label of request 2- Casher send respond to stockman
Alternative Paths	No alternative
Postcondition	Casher send respond to stockman
Include	

Use Case Name	Order
Precondition	Casher must have login
Basic Path	1- Casher enter to label of order 2- Casher take order 3- Casher save order in system
Alternative Paths	No alternative
Postcondition	All orders save in system
Include	Database



### 3-StockMan

Use Case Name	Login
Precondition	Must have account
Basic Path	1- Stockman login to system 2- System check the account 3- stockman go to stockman dashboard
Alternative Paths	No alternative
Postcondition	stockman go to stockman dashboard
Include	Database

<b>Use Case Name</b>	Manage stock
<b>Precondition</b>	stockman must have login
<b>Basic Path</b>	1- Stockman enter label of stock 2- Stock man show and update all data about stock
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	All data of stock are updated
<b>Include</b>	Database

<b>Use Case Name</b>	Request from cashier
<b>Precondition</b>	Casher must have login
<b>Basic Path</b>	1- Stock man enter to label of request 2- Stock man send request to
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	stockman send request to cashier
<b>Include</b>	Database

<b>Use Case Name</b>	Respond to cashier
<b>Precondition</b>	Stock man must have login
<b>Basic Path</b>	1- Cashier enter to label of request 2- Cashier send respond to stockman
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	stockman send request to cashier
<b>Include</b>	Database

Use Case Name	Contact with suppliers
Precondition	stockman must have login
Basic Path	1- Stockman enter to label of suppliers 2- Stockman have data of all supplier 3- Stock man show all suppliers and contact with him
Alternative Paths	No alternative
Postcondition	Stock man show all suppliers's data
Include	Database



#### 4- Manager:

Use Case Name	Login
Precondition	Must have account
Basic Path	1-Manager login to system 2- System check the account 3-Manager go to Manager dashboard
Alternative Paths	No alternative
Postcondition	Manager go to Manager dashboard
Include	Database

Use Case Name	Add Stuff
Precondition	Must have account
Basic Path	1-Manager go to label stuff 2-Manager Add stuff
Alternative Paths	No alternative
Postcondition	Add All data about stuff
Include	Database

<b>Use Case Name</b>	Delete Stuff
<b>Precondition</b>	Must have account
<b>Basic Path</b>	1-Manager go to label stuff 2-Manager Delete stuff
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	Delete All data about stuff
<b>Include</b>	Database

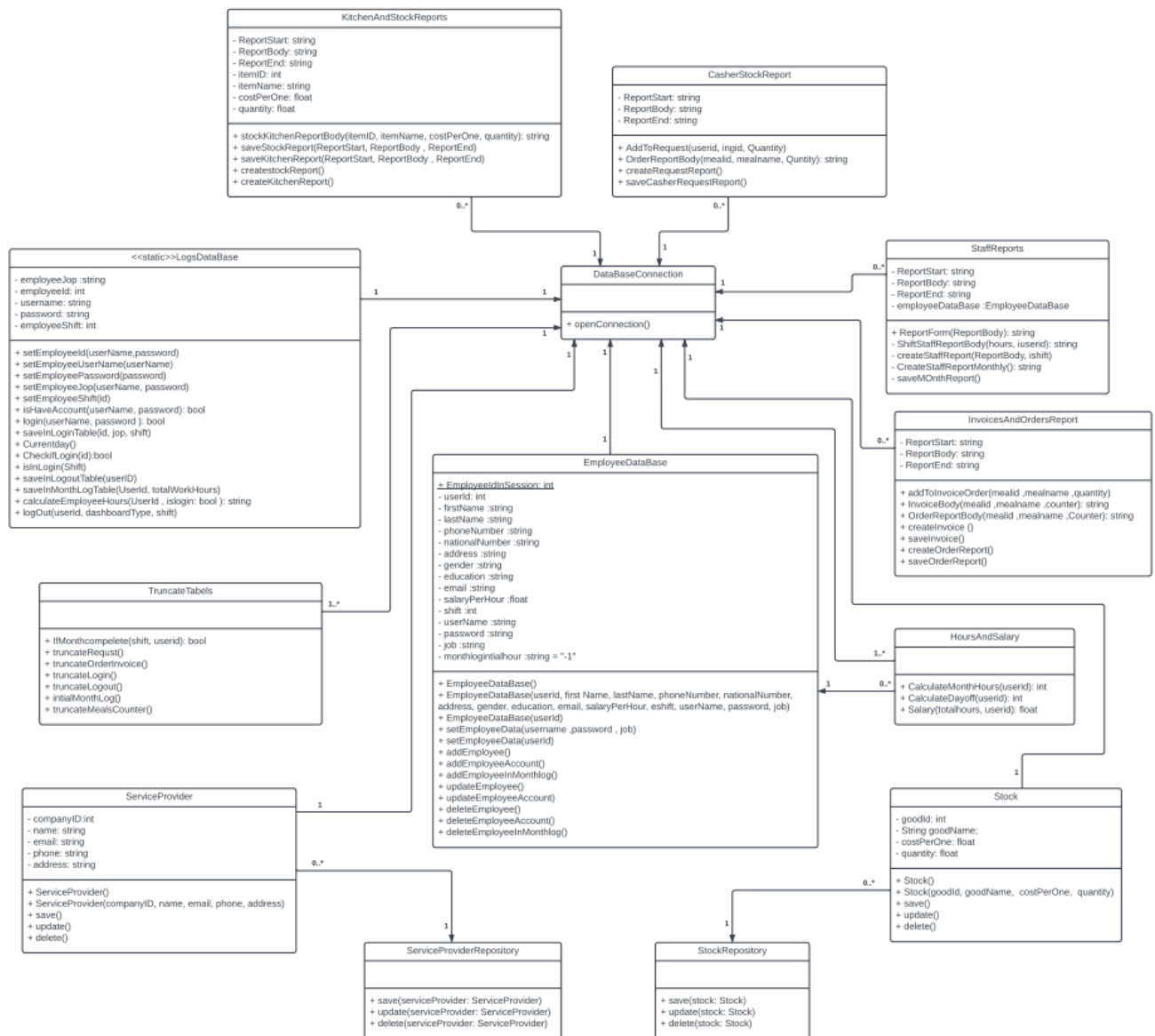
<b>Use Case Name</b>	Add meal
<b>Precondition</b>	Must have account
<b>Basic Path</b>	1-Manager go to label Food Menu 2-Manager Add Meal
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	Add meal to Food Menu
<b>Include</b>	Database

<b>Use Case Name</b>	Delete meal
<b>Precondition</b>	Must have account
<b>Basic Path</b>	1-Manager go to label Food Menu 2-Manager Delete Meal
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	Delete meal to Food Menu
<b>Include</b>	Database

<b>Use Case Name</b>	Contact with suppliers
<b>Precondition</b>	Manager must have login
<b>Basic Path</b>	1- Manager enter to label of suppliers 2- Manager have data of all supplier 3- Manager show all suppliers and contact with him
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	Manager show all suppliers's data
<b>Include</b>	Database

<b>Use Case Name</b>	Show Report
<b>Precondition</b>	Manager must have login
<b>Basic Path</b>	1- Manager enter to label of reports 2- Manager show all reports about staff and stock
<b>Alternative Paths</b>	No alternative
<b>Postcondition</b>	Manager have all reports
<b>Include</b>	Database

# Class diagram



```

classDiagram
    class StockReport {
        UserID
        StockReportID
        Date
        Time
    }
    class Employee {
        UserID
        FullName
        Password
        Nickname
        Address
        Gender
        Birthdate
        Email
        Salary
        Shift
    }
    class Invoices {
        InvoiceID
        Date
        Time
    }
    class Stock {
        StockID
        GoodsName
        CostPerOne
        Quantity
    }
    class MealFromStock {
        MealID
        GoodsID
        Consumption
    }
    class MealMenu {
        MealID
        MealName
        MealCost
    }
    class MealCounter {
        GoodsID
        GoodsName
        CostPerOne
        Quantity
    }
    class MonthLog {
        UserID
        Day1Hours
        Day2Hours
        Day3Hours
        Day4Hours
        Day5Hours
        Day6Hours
        Day7Hours
        Day8Hours
        Day9Hours
        Day10Hours
        Day11Hours
        Day12Hours
        Day13Hours
        Day14Hours
        Day15Hours
        Day16Hours
        Day17Hours
        Day18Hours
        Day19Hours
        Day20Hours
        Day21Hours
        Day22Hours
        Day23Hours
        Day24Hours
    }
    class CashierStockReports {
        UserID
        Report
        Date
    }
    class KitchenReport {
        UserID
        KitchenReport
        Date
    }
    class OrderReports {
        UserID
        OrderReports
        Date
    }
    class ServiceProvider {
        CompanyID
        CompanyName
        CellPhone
        CallPhone
        CallAddress
    }
    class Kitchen {
        GoodsName
        CostPerOne
        Quantity
    }
    class Accounts {
        UserID
        UserName
        Password
    }
    class Login {
        UserID
        Date
        Time
        Log
    }
    class Logout {
        UserID
        Date
    }
    class Request {
        UserID
        Request
        Quantity
    }

    StockReport "0" -- "0" Employee : StockReport
    Employee "0" -- "0" Invoices : Employee
    Employee "0" -- "0" Stock : Employee
    Employee "0" -- "0" MealFromStock : Employee
    Employee "0" -- "0" MealMenu : Employee
    Employee "0" -- "0" MealCounter : Employee
    Employee "0" -- "0" MonthLog : Employee
    Employee "0" -- "0" CashierStockReports : Employee
    Employee "0" -- "0" KitchenReport : Employee
    Employee "0" -- "0" OrderReports : Employee
    Employee "0" -- "0" ServiceProvider : Employee
    Employee "0" -- "0" Kitchen : Employee
    Employee "0" -- "0" Accounts : Employee
    Employee "0" -- "0" Login : Employee
    Employee "0" -- "0" Logout : Employee
    Employee "0" -- "0" Request : Employee
    Invoices "0" -- "0" Stock : Invoices
    Stock "0" -- "0" MealFromStock : Stock
    MealFromStock "0" -- "0" MealMenu : MealFromStock
    MealMenu "0" -- "0" MealCounter : MealMenu
    MonthLog "0" -- "0" CashierStockReports : MonthLog
    MonthLog "0" -- "0" KitchenReport : MonthLog
    MonthLog "0" -- "0" OrderReports : MonthLog
    MonthLog "0" -- "0" ServiceProvider : MonthLog
    MonthLog "0" -- "0" Kitchen : MonthLog
    MonthLog "0" -- "0" Accounts : MonthLog
    MonthLog "0" -- "0" Login : MonthLog
    MonthLog "0" -- "0" Logout : MonthLog
    MonthLog "0" -- "0" Request : MonthLog
    CashierStockReports "0" -- "0" KitchenReport : CashierStockReports
    CashierStockReports "0" -- "0" OrderReports : CashierStockReports
    CashierStockReports "0" -- "0" ServiceProvider : CashierStockReports
    CashierStockReports "0" -- "0" Kitchen : CashierStockReports
    CashierStockReports "0" -- "0" Accounts : CashierStockReports
    CashierStockReports "0" -- "0" Login : CashierStockReports
    CashierStockReports "0" -- "0" Logout : CashierStockReports
    CashierStockReports "0" -- "0" Request : CashierStockReports
    KitchenReport "0" -- "0" OrderReports : KitchenReport
    KitchenReport "0" -- "0" ServiceProvider : KitchenReport
    KitchenReport "0" -- "0" Kitchen : KitchenReport
    KitchenReport "0" -- "0" Accounts : KitchenReport
    KitchenReport "0" -- "0" Login : KitchenReport
    KitchenReport "0" -- "0" Logout : KitchenReport
    KitchenReport "0" -- "0" Request : KitchenReport
    OrderReports "0" -- "0" ServiceProvider : OrderReports
    OrderReports "0" -- "0" Kitchen : OrderReports
    OrderReports "0" -- "0" Accounts : OrderReports
    OrderReports "0" -- "0" Login : OrderReports
    OrderReports "0" -- "0" Logout : OrderReports
    OrderReports "0" -- "0" Request : OrderReports
    ServiceProvider "0" -- "0" Kitchen : ServiceProvider
    ServiceProvider "0" -- "0" Accounts : ServiceProvider
    ServiceProvider "0" -- "0" Login : ServiceProvider
    ServiceProvider "0" -- "0" Logout : ServiceProvider
    ServiceProvider "0" -- "0" Request : ServiceProvider
    Kitchen "0" -- "0" Accounts : Kitchen
    Kitchen "0" -- "0" Login : Kitchen
    Kitchen "0" -- "0" Logout : Kitchen
    Kitchen "0" -- "0" Request : Kitchen
    Accounts "0" -- "0" Login : Accounts
    Accounts "0" -- "0" Logout : Accounts
    Accounts "0" -- "0" Request : Accounts
    Login "0" -- "0" Logout : Login
    Login "0" -- "0" Request : Login
    Logout "0" -- "0" Request : Logout
  
```