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	201900607			
محمد اسامه محمد سلامه	201900630			
مصطفي محي الدين	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 search accommodation nearby the work place. It was difficult earlier to go a place to work which was situated at long-distances and hence had to lose good opportunities, as we don’t know where to stay and where unaware about a particular city. But now it has become easier to find an accommodation nearby the work place in cheap rates. In this online paying guest system, users can find a number of paying accommodation nearby work place or desired place. This project aims to facilitate the process of reserving places by providing the ability for the user to search for his preferred location and display the places available for rent. Providing accounts also enables the user to announce his own property that he wants to display to tenants. These systems will be available to all users to communicate with the owner, the ability to browse more than one publication, the ability to advertise and its full details

The system will be designed to work to display advertisements to users and the places available to them and to provide the opportunity to add ads to the user's favorite list and to provide the ability to communicate with the owner.

1.3. 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
Admin	The person who has access to system, modify into stuff and show and upload report to manger
Manger	The person who has access to modify meal, (add – update – delete) staff, (add -update – delete) meal

1.4. 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.

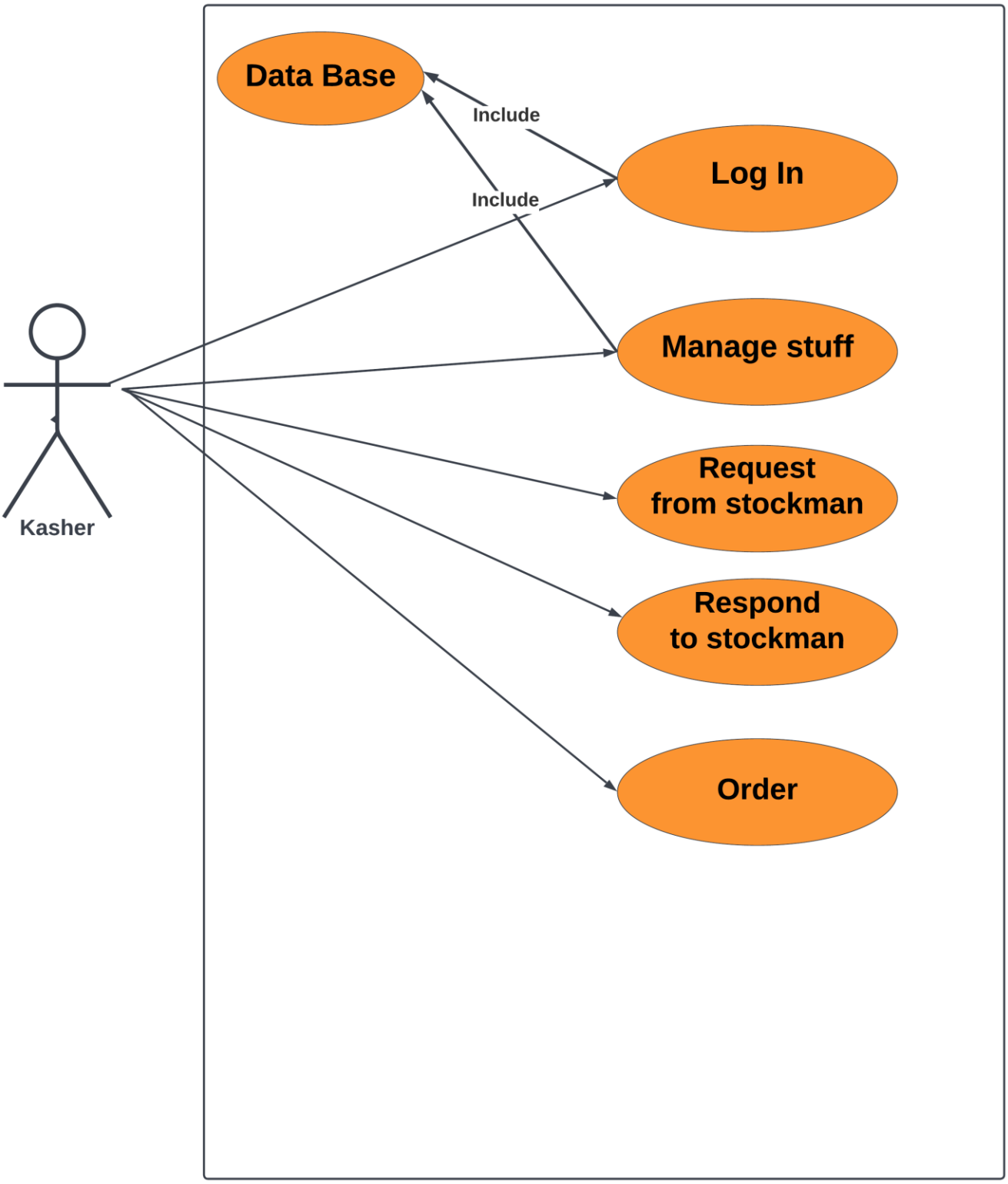
3.2.3. Usability requirements

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

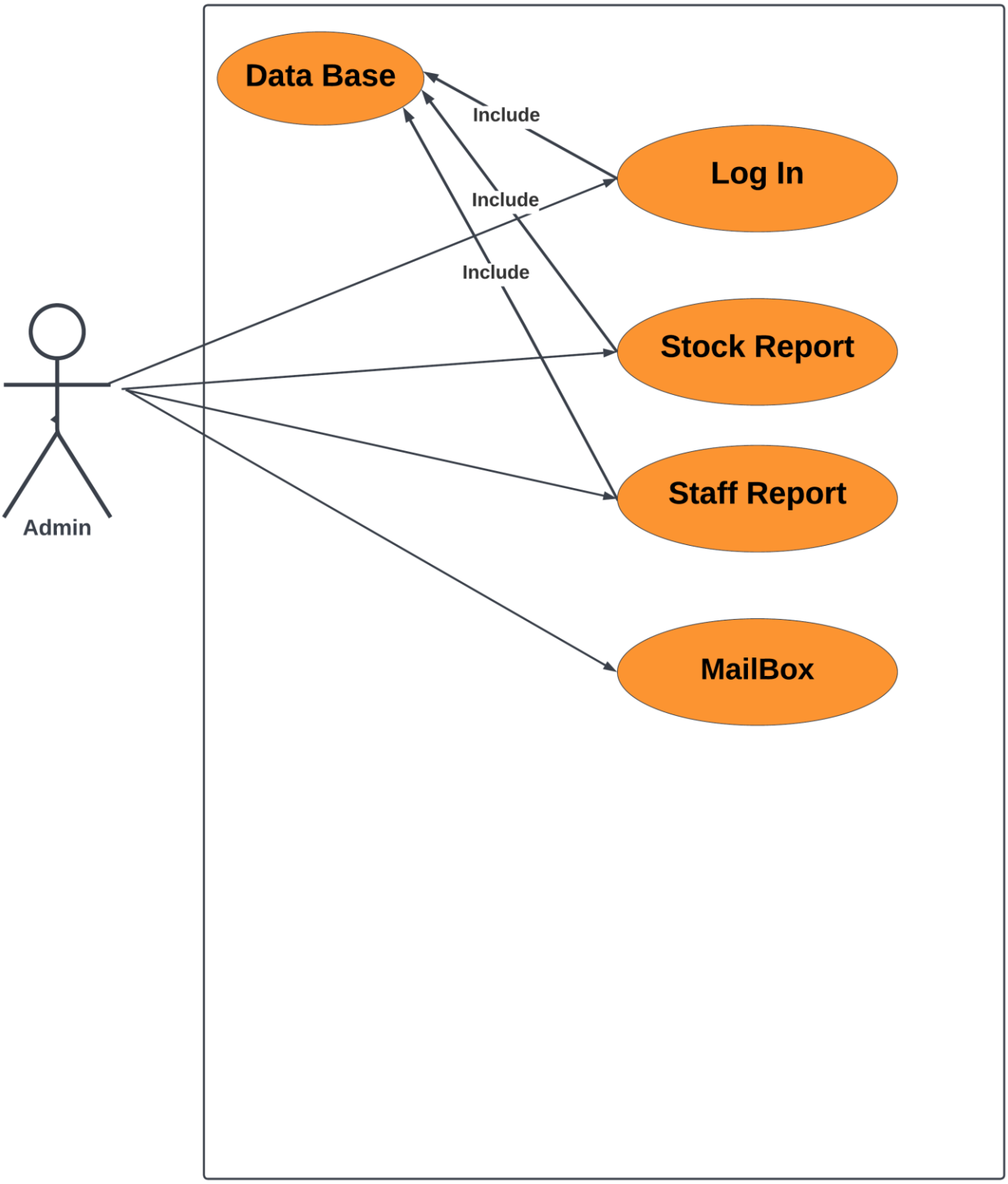
---

Use case:

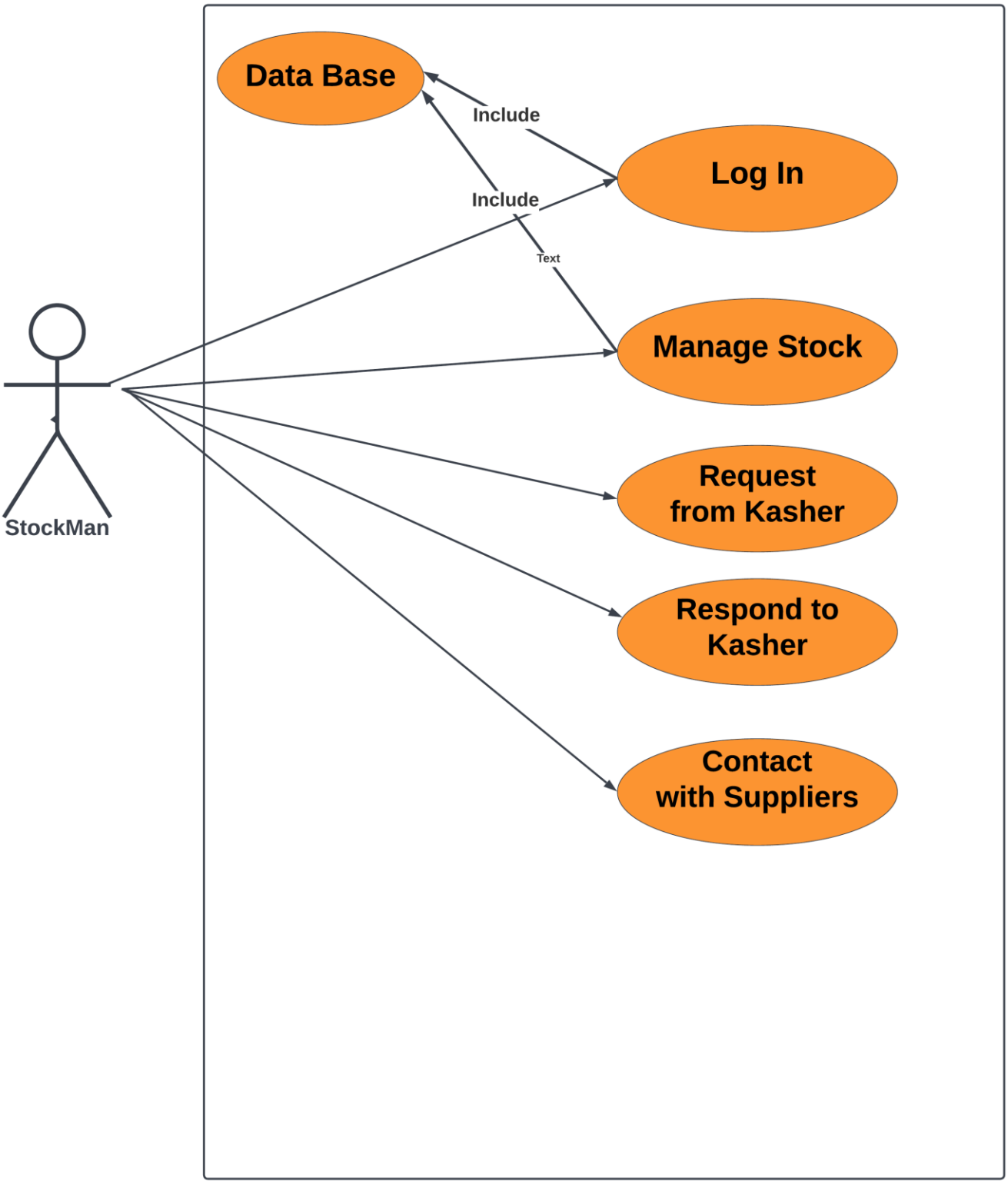
Casher:


Data Base
Include
Include
Log In
Manage stuff
Request
from stockman
Respond
to stockman
Order
Kasher

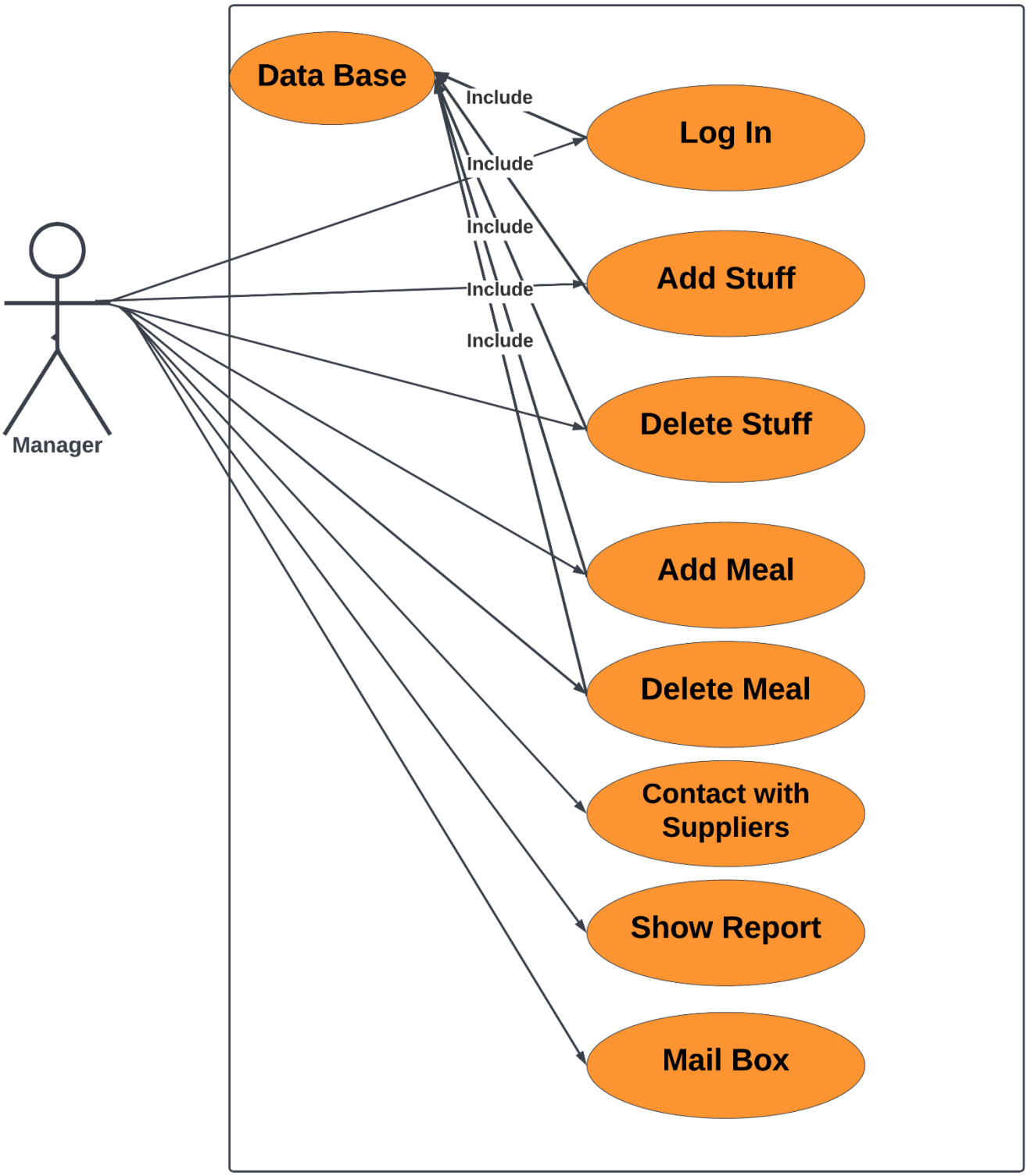
Admin:


Data Base
Include
Log In
Include
Include
Stock Report
Staff Report
MailBox
Admin

Stock Man:


Data Base
Include
Log In
Include
Text
Manage Stock
StockMan
Request from Kasher
Respond to Kasher
Contact with Suppliers

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

<-->

2 - Admin:

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

Include	Database

Use Case Name	StockReport
Precondition	Admin must have login
Basic Path	1- Admin Enter in label of StockReport 2- Write Report about Stock
Alternative Paths	No alternative
Postcondition	All Reports about Stock Done
Include	Database

Use Case Name	Staff Report
Precondition	Admin must have login
Basic Path	1- Admin Enter in label of Stuff Report 2- Write Report about All Stuff
Alternative Paths	No alternative
Postcondition	All Reports about Staff Done
Include	Database

Use Case Name	Mail Box
Precondition	Admin must have login
Basic Path	1-Admin in label of mail box 2-Admin send mail to manager
Alternative Paths	No alternative
Postcondition	All Emails send
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

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

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

Postcondition	stockman send request to casher
Include	Database

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

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

Include	Database

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

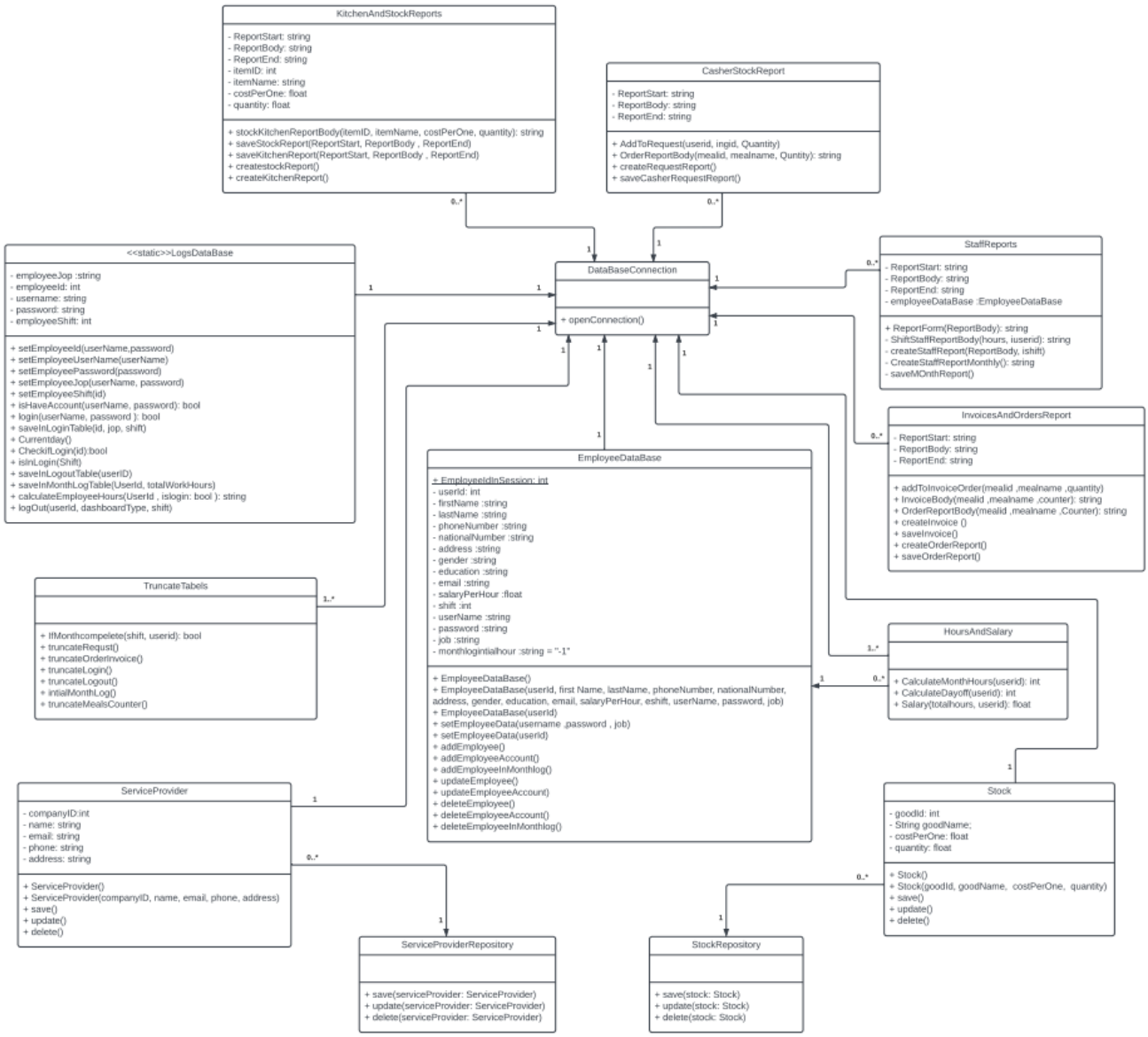
Use Case Name	Delete meal
Precondition	Must have account
Basic Path	1-Manager go to label Food Menu 2-Manager Delete Meal
Alternative Paths	No alternative
Postcondition	Delete meal to Food Menu
Include	Database

Use Case Name	Mail Box
Precondition	Manager must have login
Basic Path	1-Manager go in label of mail box 2-Manager send and recieve mail from admin
Alternative Paths	No alternative
Postcondition	All Emails Respond
Include	Database

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

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

Class diagram



Data base Schema: