

1. Overview

Service desk application is a tool used for solving the queries or the issues faced by employees of different departments, Service desk application is served as a centralized location for connection or as a link between different departments and helps in solving the issues in a seamless manner. Service desk application acts as a connection between the employees of various department and will help in solving the queries quickly which is directly intended towards good user experience. Quick solving of the issues will help in gaining the users trust, which will help in the growth of the company.

2. Objectives

The main objectives of the project are:

- Developing user friendly, single point of contact web application for management of queries raised by the users.
- Developing a centralized application for queries management.
- Management and quick resolving of the issues raised.
- Connecting various departments in a single platform for faster resolving of issues.
- E mail notification and Power BI dash board implementation for taking quick actions.

3. Scope

- Implementation of the of the application by using APS.Net MVC (Model View Controller) architecture.
- Designing and using SQL database for backend to store the user details and the raised ticket details.
- Using web API's for enabling the development of HTTP services to reach out the client.
- Using E-mail notification for knowing the assignment of ticket or for giving the response of solving the query.
- Hosting the application on azure for better scalability and low-cost maintenance.
- Building a data model to transfer the data using Azure Data Factory and use it as a backend for Power BI Reporting.
- Using Power BI dashboard for showing the ticket status, this helps in solving the issues quickly by seeing the unsolved queries in visualization form.

4. Modules

➤ Admin

- Admin can maintain all the employees' details.
- Admin can view all the sections and keeps the track of every activity.
- Admin can assign the ticket to a particular group for solving the issue.

➤ Lead

- Lead manages the tickets generated by a particular group.
- Lead takes care of assigning and working the tickets raised by groups.
- Team lead monitors the team level activities.

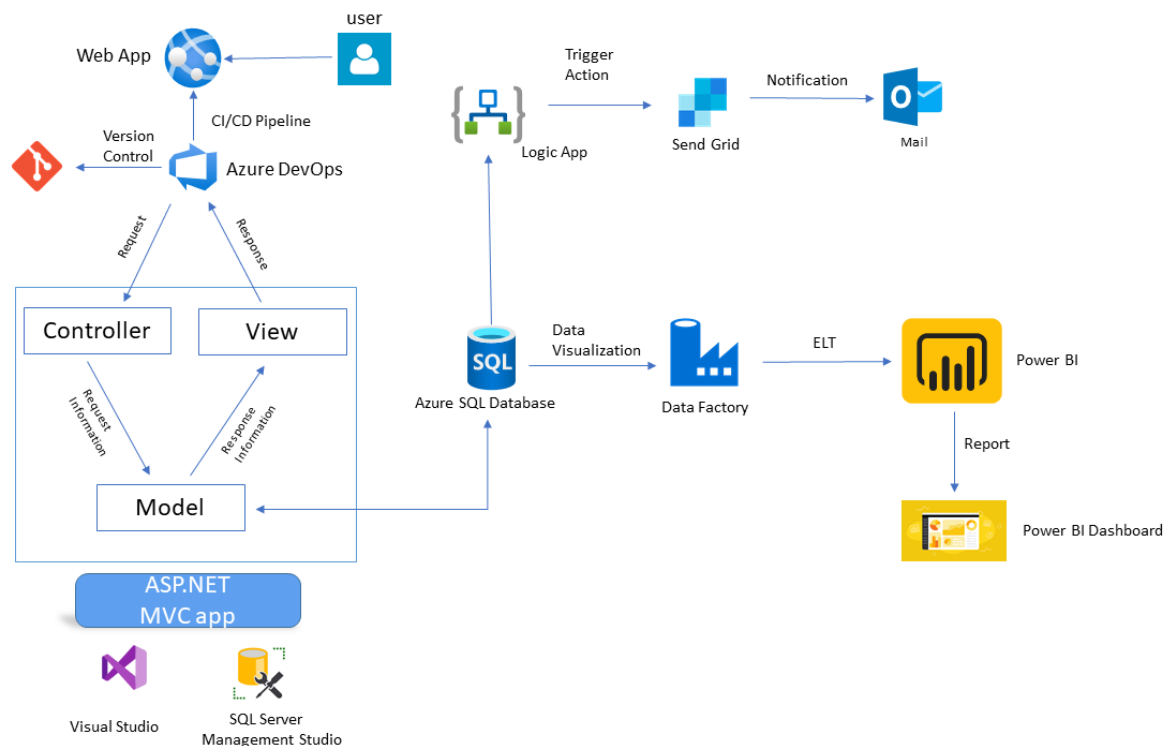
➤ Manager

- Manager manages the tickets generated by the particular departments.
- Manager takes care of department level activities related to the raising and solving of queries.

➤ User

- User can raise new queries and can keep the track of his already queries.
- User can modify already created queries.
- User can view and update his profile.

5. Application architecture



1. User

User is the person who uses the application generally the users are registered employees and team leads, managers and admin. User uses web application to interact with the application.

2. Azure Web App

Web Apps are one of the most widely used Azure service that enables us to host our web applications in Azure. It can host a User Interactive application or even a backend service like a WCF Service or Web APIs. With Web Apps, we can leverage the power of Microsoft Azure and build a multi-functional, immensely scalable, highly secured and seamlessly accessible internet or intranet-based applications.

3. ASP.Net MVC

ASP.NET MVC is an open-source software from Microsoft. It is a web development framework combines the features of MVC (Model-View-Controller) architecture.

- **Model:** Model represents the shape of the data. A class in C# is used to describe a model. Model objects store data retrieved from the database.
- **View:** View in MVC is a user interface. View display model data to the user and also enables them to modify them. View in ASP.NET MVC is

HTML, CSS, and some special syntax (Razor syntax) that makes it easy to communicate with the model and the controller.

- **Controller:** The controller handles the user request. Typically, the user uses the view and raises an HTTP request, which will be handled by the controller. The controller processes the request and returns the appropriate view as a response.

4. Visual Studio

Microsoft Visual Studio is an IDE made by Microsoft and used for different types of software development such as computer programs, websites, web apps, web services, and mobile apps. It contains completion tools, compilers, and other features to facilitate the software development process.

5. Azure SQL Database

Azure SQL is a family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud. Azure SQL Database: Support modern cloud applications on an intelligent, managed database service, that includes serverless compute.

6. SQL Server Management Studio

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases.

7. Azure Storage

An Azure Storage Account is a secure account, which provides you access to services in Azure Storage. The storage account is like an administrative container, and within that, we can have several services like blobs, files, queues, tables, disks, etc. And when we create a storage account in Azure, we will get the unique namespace for our storage resources. That unique namespace forms the part of the URL. The storage account name should be unique across all existing storage account name in Azure.

8. Azure Data Factory

Azure Data Factory is Azure's cloud ETL service for scale-out serverless data integration and data transformation. It offers a code-free UI for intuitive authoring and single-pane-of-glass monitoring and management. You can also lift and shift existing SSIS packages to Azure and run them with full compatibility in ADF.

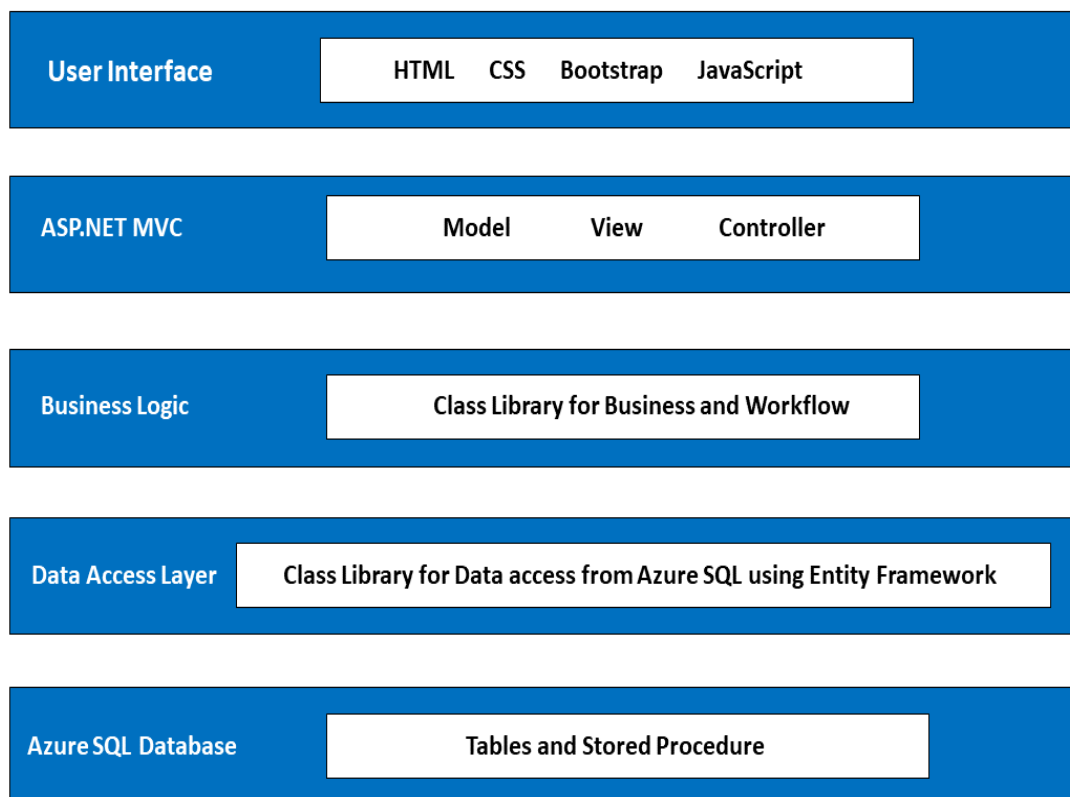
9. Azure Logic Apps

Azure Logic Apps is a cloud-based platform for creating and running automated workflows that integrate your apps, data, services, and systems. User can do the tasks like moving uploaded files from an SFTP or FTP server to Azure Storage. Monitor tweets, analyse the sentiment, and create alerts or tasks for items that need review.

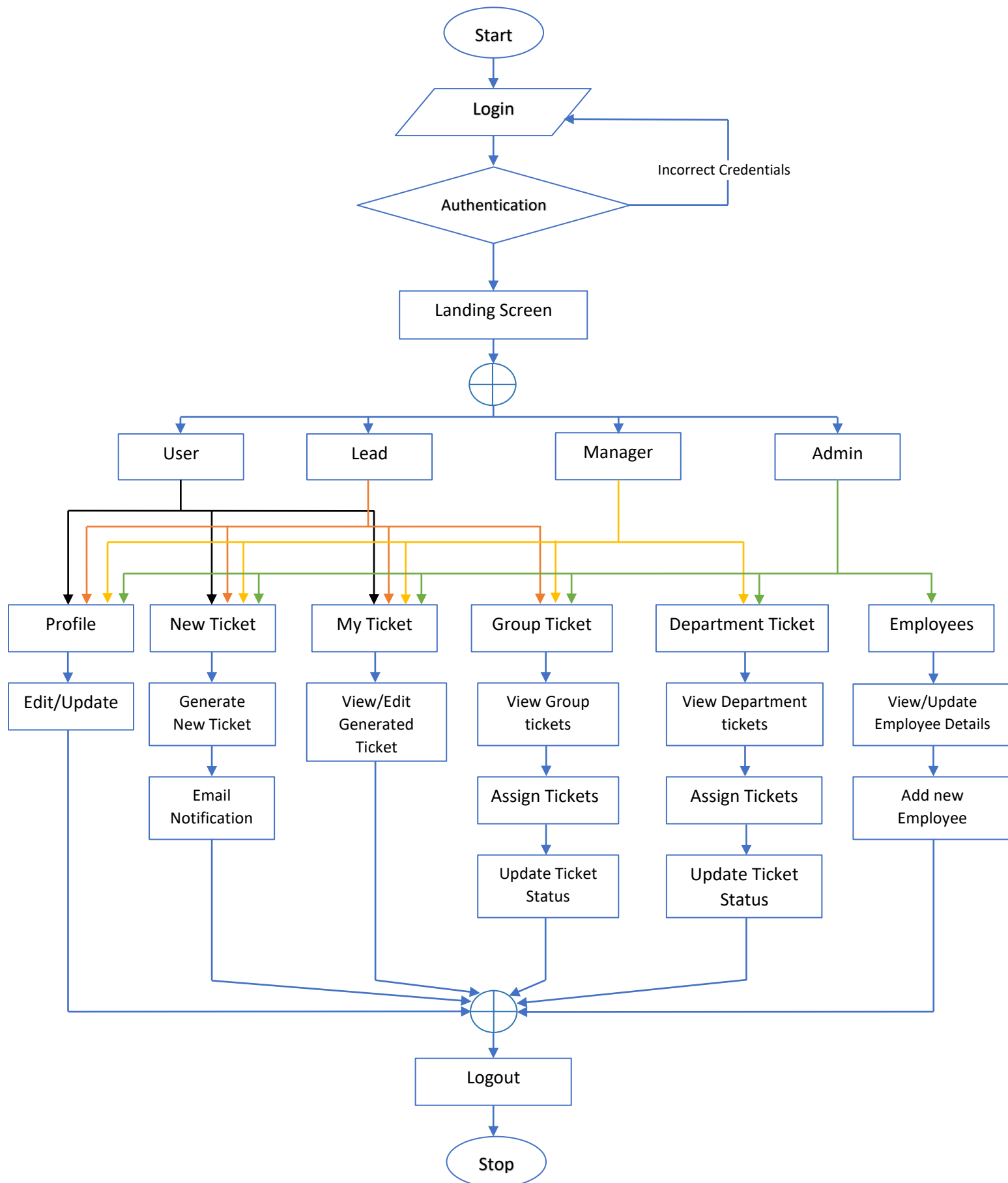
10. Power BI

Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights, data may be an Excel spreadsheet, or a collection of cloud-based and on-premises hybrid data warehouses. Power BI helps to easily connect to your data sources, visualize and discover what's important, and share that with anyone or everyone you want.

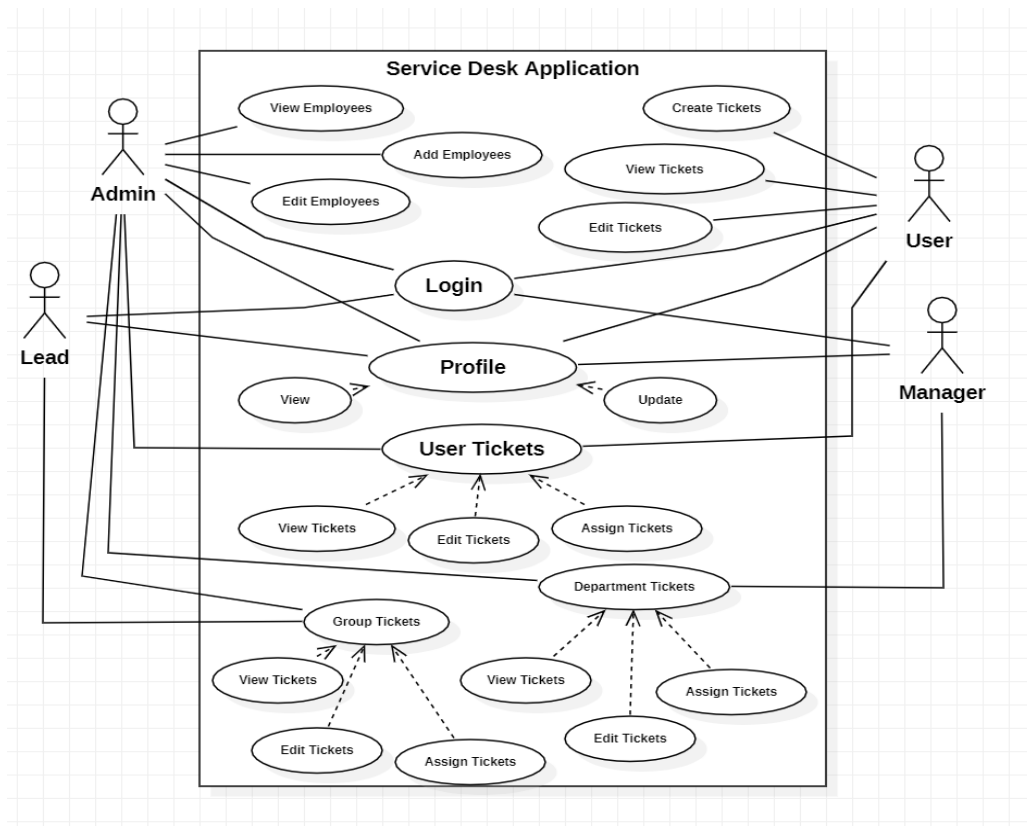
6. Layered layout



7. Flowchart



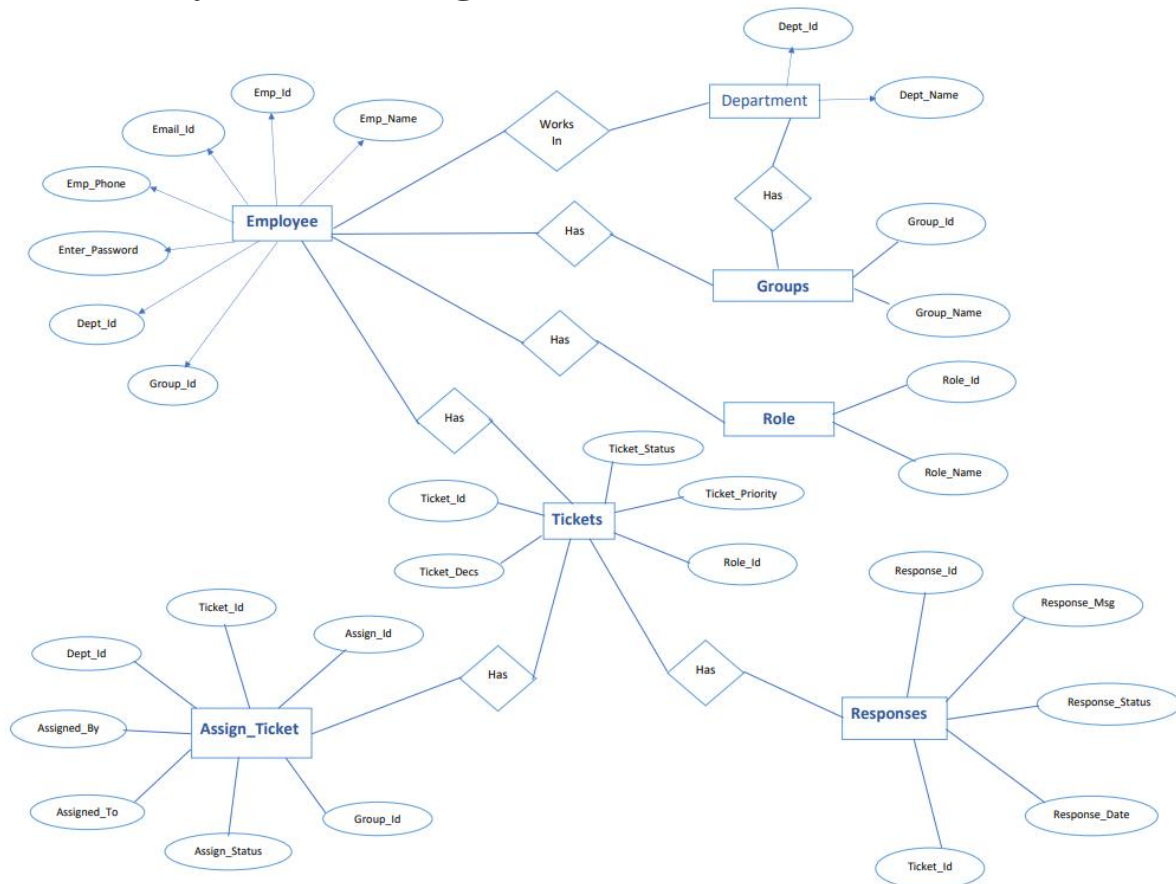
8. Use case Diagram



Actors and Use cases

SL No	Actors	Use cases
1	Admin	<ul style="list-style-type: none"> Admin can add, update and remove user details. Admin can view user profiles. Admin can view user, group and department tickets. Admin can assign tickets. Admin can raise tickets.
2	Lead	<ul style="list-style-type: none"> Lead can view user tickets. Lead can view and assign group tickets. Lead can view and edit his profile Lead can raise tickets.
3	Manager	<ul style="list-style-type: none"> Manager can view user tickets. Manager can view and assign department tickets. Manager can view and edit his tickets. Manager can raise tickets.
4	User	<ul style="list-style-type: none"> User can raise tickets. User can view raised tickets. User can view and update his profile.

9. Entity Relation Diagram



The tables of the database are

- **Employees**
 - Emp_Id – This is a primary key value used to uniquely identify the Employees.
 - Emp_Name – This field is used to store the name of the employee.
 - Emp_Email – This field is used to store the Email of the employee.
 - Emp_Phone – This field is used to store the Phone Number of the employee.
 - Emp_Password – This field is used to store the Password of the employee.
 - Group_Id – This field represents foreign key relationship from Group table.
 - Dept_Id - This field represents foreign key relationship from Group table.
- **Roles**
 - Role_Id – This is a primary key value used to uniquely identify the Roles table.
 - Role_Name – This field is used to store the Role names of the employee.
- **Departments**
 - Dept_Id – This is a primary key value used to uniquely identify the department.
 - Dept_Name – This field is used to store the name of the department.

- **Groups**

- Group_ID – This is a primary key value used to uniquely identify the group.
- Group_Name – This field is used to store the name of the group.
- Dept_Id – This field represents foreign key relationship from Departments table.

- **Tickets**

- Ticket_ID – This is a primary key value used to uniquely identify the tickets.
- Status – This field is used to store the status of the ticket.
- Ticket_Desc – This field is used to store the description of the ticket.
- Emp_ID – This field represents foreign key relationship from Employees table.

- **Assign_Tickets**

- Assign_ID – This is a primary key value used to uniquely identify the Assigned Tickets.
- Ticket_ID – This field represents foreign key relationship from Tickets table
- Assign_By – This field is used to store the name of the person who assigned the ticket.
- Group_ID – This field represents foreign key relationship from Groups table.
- Dept_ID – This field represents foreign key relationship from Departments table.
- Assign_To – This field is used to store the name of the person whom the ticket is assigned.
- Status - This field is used to store the status of the ticket.

- **Responses.**

- Response_Id – This is a primary key value used to uniquely identify the Responses.
- Response_Msg – This field represents the responses message sent by the person whom the ticket is assigned.
- Response_Status – This field represents the status of the ticket, whether the ticket is resolved or yet to be resolved.
- Response_Date – This field represents the date when the response is given.
- Ticket_Id - This field represents foreign key relationship from Ticket table.

10. Database Structure

