**Table of Contents**

[**1.** **Overview** 2](#_Toc92754570)

[**2.** **Objectives** 3](#_Toc92754571)

[**3.** **Scope** 3](#_Toc92754572)

[**4.** **Project Features** 3](#_Toc92754573)

[**5.** **Project Requirements** 5](#_Toc92754574)

[**5.1** **Functional Requirements** 5](#_Toc92754575)

[**5.2** **Non-Functional Requirements** 6](#_Toc92754576)

[**5.3** **Software Requirements** 6](#_Toc92754577)

[**5.4** **Tools Used** 7](#_Toc92754578)

[**6.** **Modules** 7](#_Toc92754579)

# **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.

# **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.

# **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.

# **Project Features**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature ID** | **Feature Name** | **Feature Description** | **Feature Details** |
| 1 | Login | In Login page user should be able to login using their credentials. Credentials are username and password. | * Login page is the gateway for users to enter into the application. * By entering correct credentials in the login page users gets the access to the application. * Login page includes two text boxes for entering the credentials and a button to submit the data. * When user clicks on the button the credentials are validated and if the credentials are correct then the user will be allowed to access the application. |
| 2 | Landing Screen | This is the home page of the application where user can see and use the different options | * Different sections user can see in landing screen are  1. Profile 2. My Tickets 3. New Ticket   4) Group Tickets   1. Department Tickets 2. Employees  * Admin can all 6 sections. * Manager can access first 5 screens. * Lead can access first 4 sections. * User can access first 3 sections. * In Profile section user can check out his profile information. * In My Tickets section user can see the queries raised by him and also the status of the queries. * New Tickets section user can raise new query tickets. * In Group tickets section includes the queries raised by a group. * Department tickets section included the tickets raised by departments. |
| 3 | Profile Screen | In profile screen page user can view and edit his profile details. | * Profile screen includes  1. Employee ID 2. Employee name 3. Gender 4. E-mail 5. Change Password 6. Mobile Number 7. User Role 8. Department 9. Group  * User can view and edit his information in this page, after saving his edited information user can see updated information. |
| 4 | My Tickets Screen | In My tickets user can see the information of the tickets raised by him. | * User can view the details of all the raised tickets by him. * User can edit the information of the ticket raised. * User can view the status of the ticket raised. * User can see whom the raised ticket is assigned. * Based on the user’s group or department he can view the particular queries raised by the groups or departments. |
| 5 | New Tickets Screen | In this page user can raise new tickets | * In ticket raising page user mention all the information of the issue. * User selects the group or department he belongs. * After raising the ticket, a mail or a message goes to the respected department or group. * After receiving the query admin or Manager assign the task to a proper team to solve the issues * Raised tickets details can be viewed and edited in My Tickets page. |
| 6 | Employees Screen | All the companies existing employee’s information can be viewed in this page. | * This page can only be accessed by the admin. * Admin can view and edit the existing employee’s information. * Admin can also add new employees and also can delete employees’ details. |

# **Project Requirements**

## **Functional Requirements**

* **Login** – User should login properly after entering proper login credentials.
* **Landing screen with all the sections** – Landing screen should include all the required section and by clicking on those section user should land in the page of that section.
* **New tickets –** The application should generate new tickets in new tickets page.
* **Editing and Updating tickets** – User should be able to edit and update his tickets.
* **E mail Notification** – Department should receive a notification whenever a new ticket is raised from the users.
* **Ticket Assigning** – Whenever a user raises new ticket it should be assigned to a person or team who can solve that issue.
* **Ticket response** – Whenever an issue is raised, user should receive the solution quickly from the assigned team.
* **Profile Screen** – In profile screen user should see the information entered by him and he should be able to update his information.
* **Employee adding** – Admin should be able to add, delete and update the employees’ details in employees’ page.

## **Non-Functional Requirements**

* **Availability:** All the different features and services of the application should available at any point of time.
* **Usability:** Users can enjoy the services safely and effectively. The interface is user friendly.
* **Maintainability –** Application should be easily maintainable.
* **Portable –** Application should perform similarly in different devices and in different environments.
* **Feasible –** Application should be feasible in terms of both economical and operational.

## **Software Requirements**

* **Framework** - ASP .Net 4.8
* **Back End** – MS SQL
* **Front End** – HTML, CSS and Boot Strap

## **Tools Used**

* Visual Studio 2019
* MS SQL Server Management Studio 18
* MS Azure
* MS Azure Data Factory
* Power BI

# **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 details from profile page.