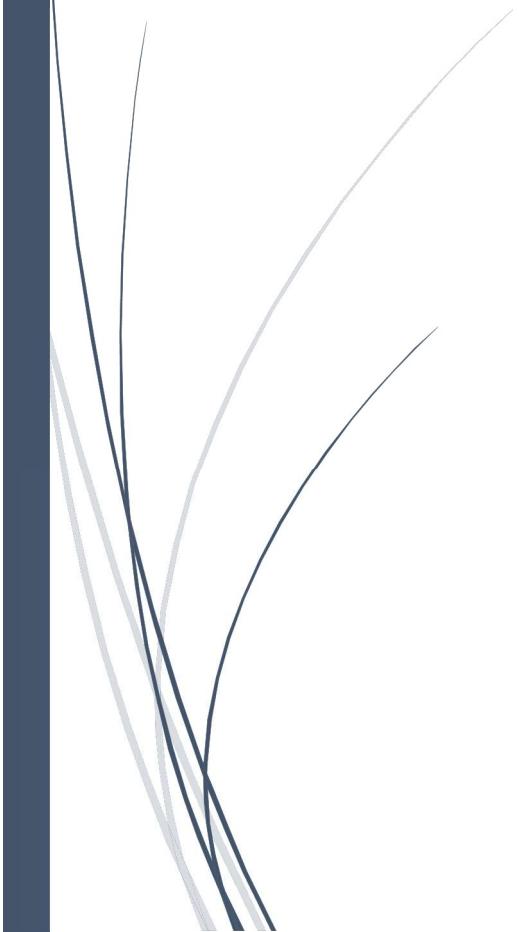




20/04/2020

Workflow Management System

A Project Report



DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF TECHNOLOGY
DHARMSINH DESAI UNIVERSITY, NADIAD

A Project Report
On
Workflow Management System
B. Tech (CE) Semester-VI

In fulfilment of all requirements for the subject of
Object Oriented Software Engineering (OOSE)

Bachelor of Technology
In
Computer Engineering

Submitted by
Yug D. Rajani (CE - 109)
(17ITUOS059)

Sani A. Ranpariya (CE - 112)
(17CEUOF017)

Under the Guidance of
Prof. Prashant M. Jadav



DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF TECHNOLOGY,
DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001
DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT



CERTIFICATE

This is to certify that the project carried out in the subject of Object Oriented Software Engineering titled "**Workflow Management System**" and recorded in this report is the bona fide work of

Yug D. Rajani
ROLL NO: CE-109
ID: 17ITUOS059

Sani A. Ranpariya
ROLL NO: CE-112
ID: 17CEUOF017

Department of Computer Engineering, Semester VI. They were involved in Project Development during the period December-2019 to April-2020.

Prof. P. M. Jadav
Guide, Associate Professor,
Department of Computer Engineering
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia
Head,
Department of Computer Engineering
Faculty of Technology
Dharmsinh Desai University, Nadiad

Table of Contents

Abstract	1
CHAPTER 1 INTRODUCTION	2
1.1 Brief Introduction of Project:	3
1.2 Tools and Technology Used for Development of Project	4
CHAPTER 2 SOFTWARE REQUIREMENTS SPECIFICATION	5
2.1 Scope	6
2.2 System Functional Requirements	6
2.3 Other Non-Functional Requirements:	12
CHAPTER 3 DESIGN	13
3.1 Use Case Diagram	14
3.2 Class Diagram	15
3.3 Sequence Diagrams	16
3.4 Activity Diagrams	18
3.5 State Diagram	20
3.6 ER Diagram	21
3.7 Data Dictionary	22
CHAPTER 4 IMPLEMENTATION DETAILS	23
4.1 Description of Modules	24
4.2 Function Prototypes	25
CHAPTER 5 TESTING	27
CHAPTER 6 SCREENSHOTS	30
CHAPTER 7 CONCLUSION	34
CHAPTER 8 LIMITATIONS AND FUTURE ENHANCEMENTS	36
8.1 Limitations	37
8.2 Future Extension	37
CHAPTER 9 BIBLIOGRAPHY	38

Abstract

In an effort to embrace automation, many companies start out with a system to manage their business processes to scale up their productivity. Whatever the state of your workflow management, using the right **Workflow Management System** with the right features is critical. Workflow management is **creating and optimizing the paths** for data in order to complete items in a given process. Workflow management includes **finding redundant tasks, mapping out the workflow** in an ideal state, **automating** the process, and **identifying bottlenecks or areas for improvement**. A Workflow Management System (WMS or WfMS) is a software tool designed to help **streamline routine business processes** for **optimal efficiency**. Workflow Management System involves creating a form to hold data and **automating a sequential path of tasks** for the data to follow until it is fully processed. Thus, such a system allows individuals to automate repetitive processes, follows up automatically on uncompleted tasks in the process, and gives an overall picture of the workflow along with performance metrics.

CHAPTER 1

INTRODUCTION

1.1 Brief Introduction of Project:

Project Name:	Workflow Management System
Project Definition/Aim:	The main aim of project is to provide better management of the workflow of an industry in a well-defined manner.
Developed For:	Various Industries which want to embrace automation
Front End:	Windows Forms Application (with C#)
Back End:	.NET framework 4.6.1
Other Technologies:	Microsoft SQL Server 2017
Documentation tools:	Microsoft Word
IDE:	Microsoft Visual Studio Community 2019

1.2 Tools and Technology Used for Development of Project:

Technology Used

- C#
- Windows Forms
- SQL

Platform Used

- Windows

Tools Used

- Microsoft Visual Studio Community 2019
- ERD Plus
- UMLet
- Microsoft SQL Server 2017

CHAPTER 2

SOFTWARE

REQUIREMENTS

SPECIFICATION

2.1 Scope

The scope of the system is very large as it is applicable to any industries trying to manage workflow and embracing automation, be it a small scale or large scale industry.

2.2 System Functional Requirements

Users:

1. Administrator
2. Project Manager
3. Employee

1. Admin

R1: Login/Logout

R.1.1: Enter credentials

Input: Admin information.

Output: "Successfully logged in/signed up" message.

R.1.2: Logout

Description: When admin wants to leave the site, then he may log out of his account.

Input: User selection.

Output: "Successfully logged out" message.

R2: Manage Users

R.2.1: Add User

Input: User information.

Output: "Successfully added" message along with ID and Password.

R.2.2: Change User Profile

Description: Admin is allowed to change the profile details of the users along with their password upon request by the user.

Input: Details to be updated along with new values.

Output: "Successfully changed" message.

R.2.3: Delete User

Description: Admin may delete the account of any user as and when required.

Input: ID of the user to be deleted.

Output: "Successfully deleted" message.

2. Project Manager

R3: Login/Logout

R.3.1: Enter credentials

Input: User information.

Output: "Successfully logged in" message.

R.3.2: Logout

Description: When user wants to leave the site, then he may log out of his account.

Input: User selection.

Output: "Successfully logged out" message.

R4: Manage Project

R.4.1: Create Project

Description: Project Manager can create a new project for starting a new project and adding members to the project team.

Input: Project details along with selection of employees working in the project.

Output: "Successfully created" message.

R.4.2: Update deadline

Description: Project manager can update the deadline of the project if the owner of the project requests to do the same.

Input: New deadline.

Output: "Successfully updated" message.

R.4.3: Notify Employees

Description: Notification can be sent to the employees after they are added to a new project.

Input: User selection

Output: "Successfully notified" message.

R.4.4: Manage Task

Description: Project manager has the main role of managing the tasks into which the project is divided into and their assignment to appropriate employees.

R.4.4.1: Create Task

Input: Task details

Output: "Successfully created" message.

R.4.4.2: Assign Task

Input: Details of user to which task has to be assigned.

Output: Notification to user.

R.4.4.3: Mark Task as Completed

Input: Task_id to be marked by accepting the request for approval by the employees.

Output: "Successfully marked" message.

R.4.4.4: Update Task Deadline

Input: Date to which the deadline is to be updated to.

Output: "Successfully updated" message.

R.4.5: Manage Queries

Description: The project manager can see and resolve the queries and doubts of the employees and help them accomplish the task easily and quickly.

R.4.5.1: View Queries

Input: User selection

Output: List of queries posted by the employees.

R.4.5.2: Resolve Query

Input: Solution of the query.

Output: Notification to employee who posted the query.

R.4.6: Mark Project as Completed

Description: The project manager is allowed to mark the project as completed after all the tasks assigned to the employees are completed.

Input: User selection

Output: "Project Status Successfully updated" message.

3. Employee

R5: Login/Logout

R.5.1: Enter credentials

Input: User information.

Output: "Successfully logged in" message.

R.5.2: Logout

Description: When user wants to leave the site, then he may log out of his account.

Input: User selection.

Output: "Successfully logged out" message.

R6: Check Notifications

Description: The employee can check various notifications regarding new project and resolution of a query.

Input: User selection.

Output: List of notifications unseen by the employee.

R7: Select project and task

R.7.1: Select Project

Input: User selection.

Output: List of tasks assigned to the employee for the selected project.

R.7.2: Select Task

Input: Selection of task from the given list.

Output: Task submission box.

R.7.3: Post Query

Input: Detailed description of query for the selected task.

Output: "Successfully posted" message.

R.7.4: Solve Query

Input: Detailed description of solution of the query posted by user.

Output: Notification to the user.

R.7.5: Request for Approval

Input: User selection.

Output: Notification to Project Manager.

2.3 Other Non-Functional Requirements:

1. Performance

The system must be fast and accurate as the main aim is automation and efficiency, it must provide accurate results in less time and using less energy.

2. Safety

As all the data of the employees, managers and projects would be stored in the system, safety is one of the main concerns of the system and hence only authorised users should be allowed to login to the system.

3. Reliability

The data and personal information of any ongoing/upcoming project(s) should not be leaked and must be in safe hands so the system must be reliable.

4. Database

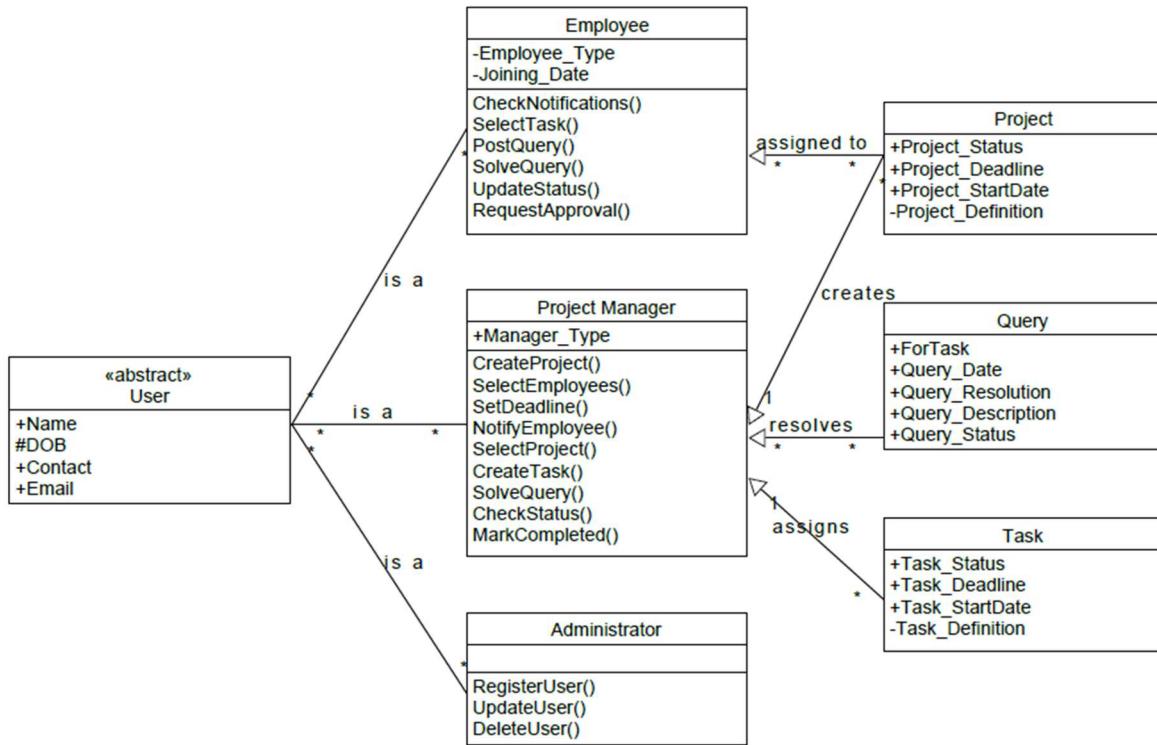
The database of the system should function perfectly, only then the data would be stored into the system correctly thus improving the performance of the system.

CHAPTER 3 DESIGN

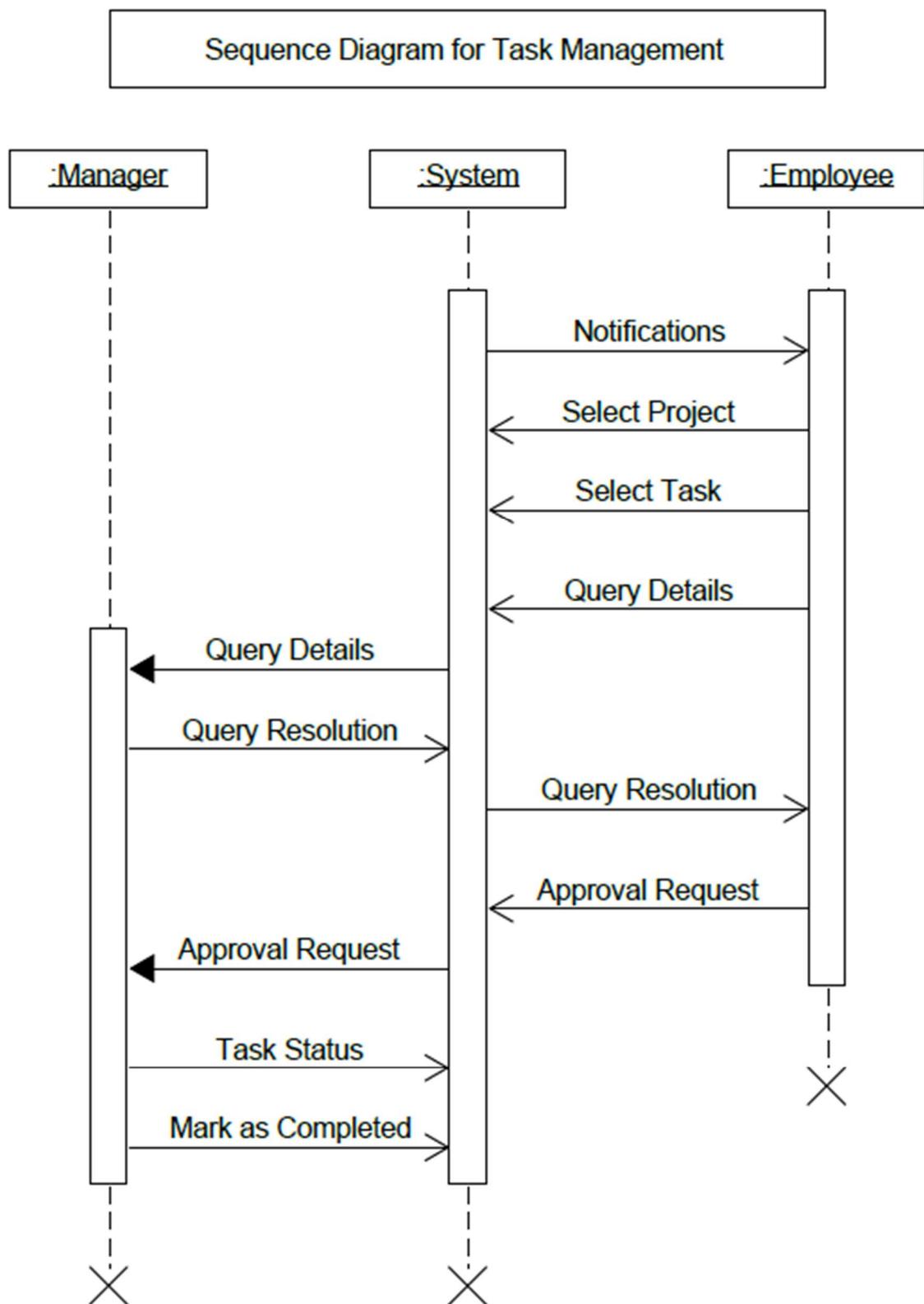
3.1 Use Case Diagram



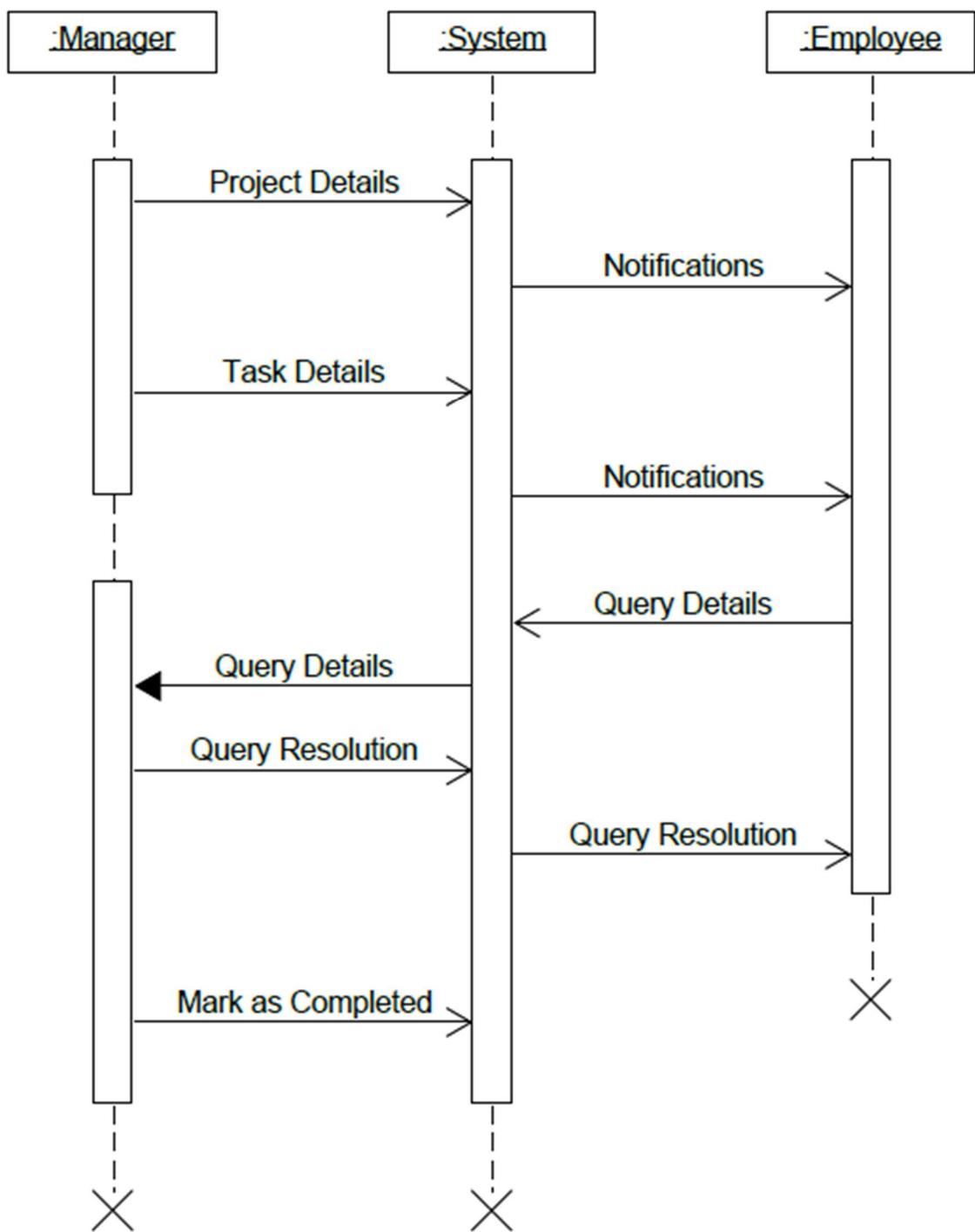
3.2 Class Diagram



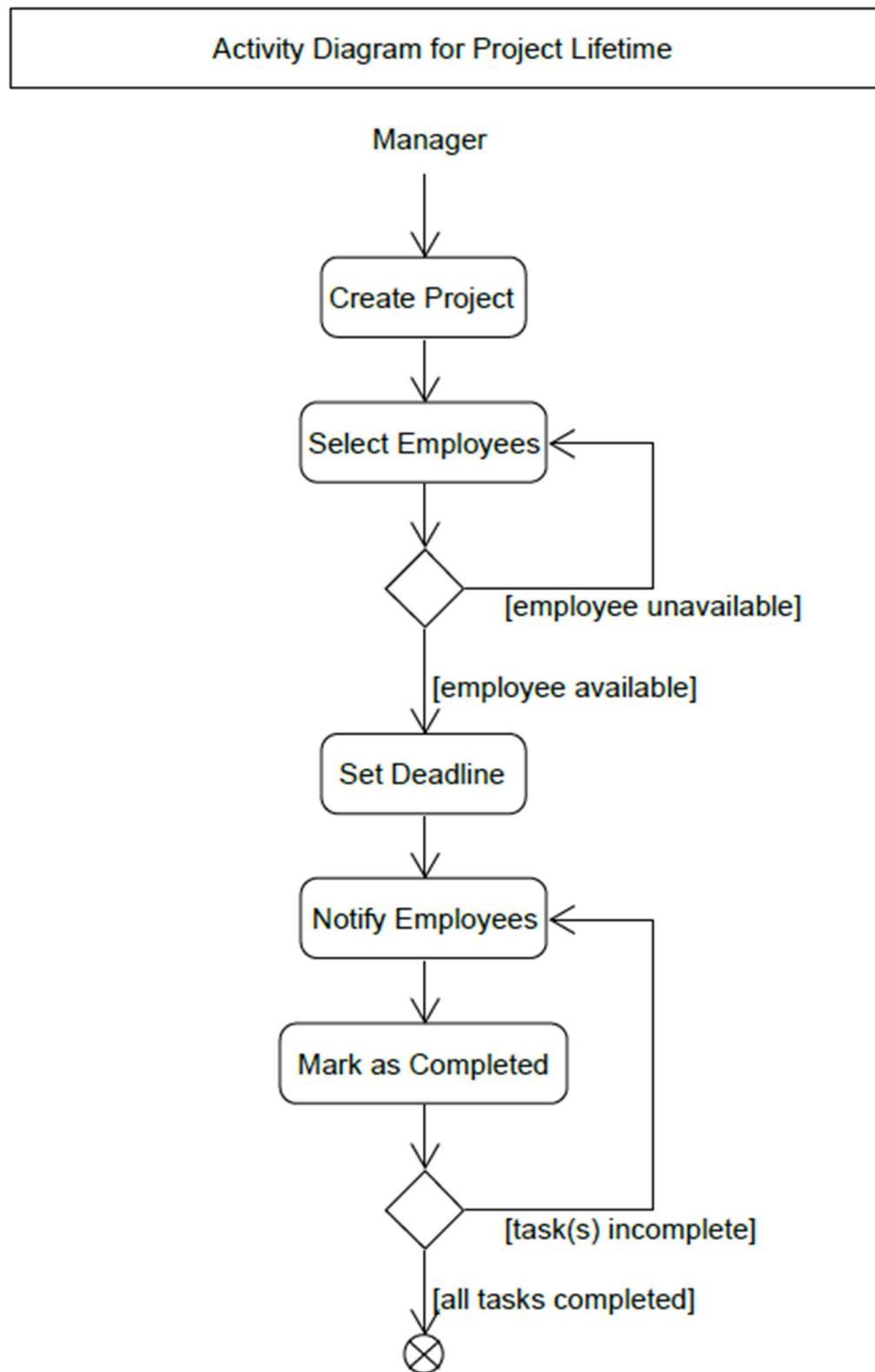
3.3 Sequence Diagrams



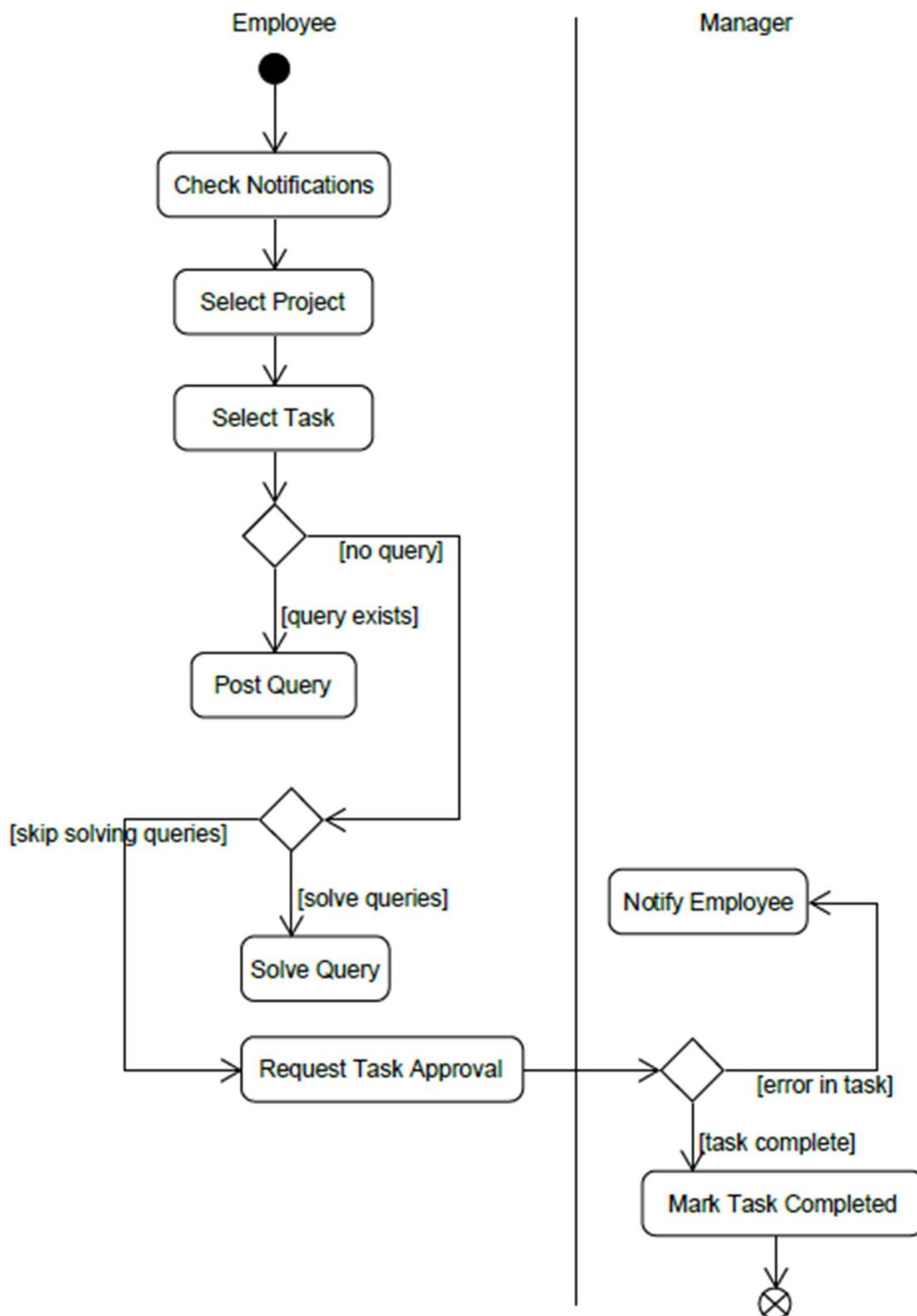
Sequence Diagram for Project Management



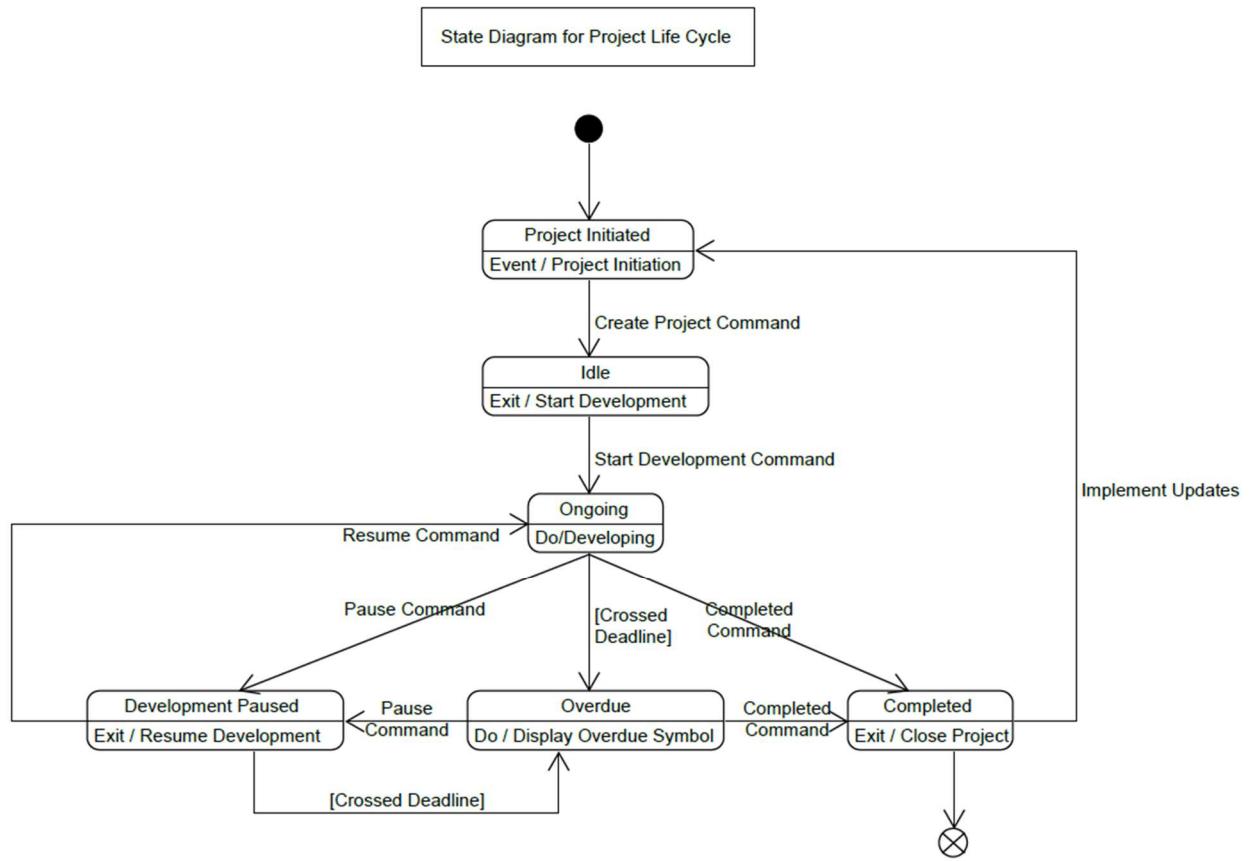
3.4 Activity Diagrams



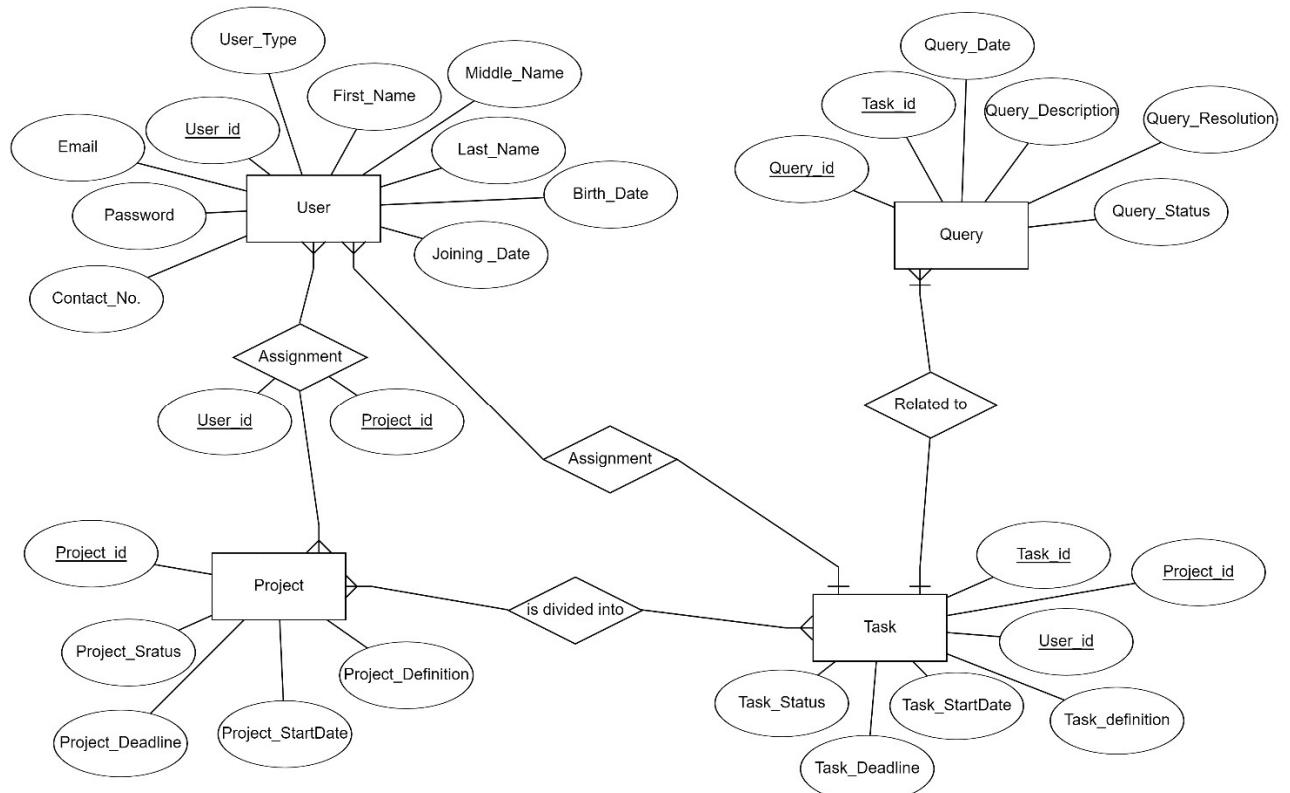
Activity Diagram for Employee Task Lifetime



3.5 State Diagram



3.6 ER Diagram



3.7 Data Dictionary

Employee							
Sr. No.	Field Name	Data Type	Width	Required	Unique	PK/FK	References
1	Employee_Id	Varchar2	5	Yes	Yes	PK	
2	Name	Varchar2	20	Yes	No		
3	Email	Varchar2	30	Yes	Yes		
4	Password	Varchar2	20	Yes	No		
5	Contact_No.	Number	10	Yes	Yes		
6	Joining_Date	date		Yes	No		

Manager							
Sr. No.	Field Name	Data Type	Width	Required	Unique	PK/FK	References
1	Manager_Id	Varchar2	5	Yes	Yes	PK	
2	Name	Varchar2	20	Yes	No		
3	Email	Varchar2	30	Yes	Yes		
4	Password	Varchar2	20	Yes	No		
5	Contact_No.	Number	10	Yes	Yes		
6	Joining_Date	date		Yes	No		

Task							
Sr. No.	Field Name	Data Type	Width	Required	Unique	PK/FK	References
1	Task_id	Varchar2	5	Yes	Yes	PK	References
2	Project_id	Number	4	Yes	No	FK	Project
3	AssignedTo_User_Id	Number	4	Yes	No	FK	Employee
4	AssignedBy_User_Id	Number	4	Yes	No	FK	Manager
5	Task_Definition	Varchar2	100	Yes	Yes		
6	Task_StartDate	date		Yes	No		
7	Task_Deadline	date		Yes	No		
8	Task_Status	Varchar2	20	Yes	No		

Query							
Sr. No.	Field Name	Data Type	Width	Required	Unique	PK/FK	References
1	Query_id	Varchar2	5	Yes	Yes	PK	
2	Task_id	Varchar2	5	Yes	No	FK	Task
3	Query_Date	date		Yes	No		
4	Query_Description	Varchar2	100	Yes	No		
5	Query_Resolution	Varchar2	100	No	No		
6	Query_Status	Varchar2	20	Yes	No		

CHAPTER 4

IMPLEMENTATION

DETAILS

4.1 Description of Modules

All the modules described here require appropriate credentials to login and use the mentioned features which leads to access of system by authorized and authenticated users only.

1. Admin

In this module, we provide all the administrator rights to the administrator of the system. The administrator can add new employees or managers, can change or edit their personal details including password if required.

2. Project Manager

In this module, the Project Managers can create new projects, update the status of ongoing projects, assign various tasks to employees, assign employees to projects and solve the queries faced by employees while trying to accomplish a particular task assigned to them.

3. Employee

In this module, the employees can see their tasks, resolution of their queries and can update the status of the tasks being performed by them for a particular project.

4.2 Function Prototypes

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        con = new SqlConnection("Data Source=LAPTOP-UK43CTN;Initial Catalog=WorkFlowDB;Integrated Security=True");
        string Employee_Id = textBox1.Text;
        string Password = textBox2.Text;

        CommonLoginDetails.empid = Int32.Parse(textBox1.Text);

        if (con.State == ConnectionState.Closed)
            con.Open();

        string qry = "SELECT * from Employee where Employee_Id='" + Employee_Id + "' and Password='" + Password + "'";
        SqlCommand cmd = new SqlCommand(qry, con);
        SqlDataReader sdr = cmd.ExecuteReader();
        if(sdr.Read())
        {
            EmployeeHome eh = new EmployeeHome();
            eh.Show();
            this.Hide();
        }
        else
        {
            MessageBox.Show("Invalid Credentials!");
        }
    }
    catch(Exception ex)
    {
        Console.WriteLine(ex.ToString());
    }
    finally
    {
        con.Close();
    }
}
```

Function Prototype 1: Login for Employee

```
public void deleteTask()
{
    try
    {
        if (con.State == ConnectionState.Closed)
            con.Open();
        string s = "DELETE Task WHERE Query_Id = @id";
        cmd = new SqlCommand(s, con);
        cmd.Parameters.AddWithValue("@id", Task_Id);
        cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message, "Error Message");
    }
    finally
    {
        con.Close();
    }
}
```

Function Prototype 2: Deletion of Task by Project Manager

```

public void addManager()
{
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        string s = "INSERT INTO Manager VALUES(@Name, @Joining_Date, @Contact_No, @Email, @Password)";
        cmd = new SqlCommand(s, con);

        cmd.Parameters.AddWithValue("@Name", Name);
        cmd.Parameters.AddWithValue("@Joining_Date", Joining_Date);
        cmd.Parameters.AddWithValue("@Contact_No", Contact_No);
        cmd.Parameters.AddWithValue("@Email", Email);
        cmd.Parameters.AddWithValue("@Password", Password);

        cmd.ExecuteNonQuery();
        MessageBox.Show("Project Manager Added Successfully!");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message, "Error");
    }
    finally
    {
        con.Close();
    }
}

```

Function Prototype 3: Addition of Project Manager by Administrator

```

public void addQuery()
{
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        string s = "INSERT INTO Query VALUES(@Task_Id, @Query_Date, @Query_Description, @Query_Resolution, @Employee_Id)";
        cmd = new SqlCommand(s, con);

        cmd.Parameters.AddWithValue("@Task_Id", Task_Id);
        cmd.Parameters.AddWithValue("@Query_Date", Query_Date);
        cmd.Parameters.AddWithValue("@Query_Description", Query_Description);
        cmd.Parameters.AddWithValue("@Query_Resolution", Query_Resolution);
        cmd.Parameters.AddWithValue("@Employee_Id", Employee_Id);

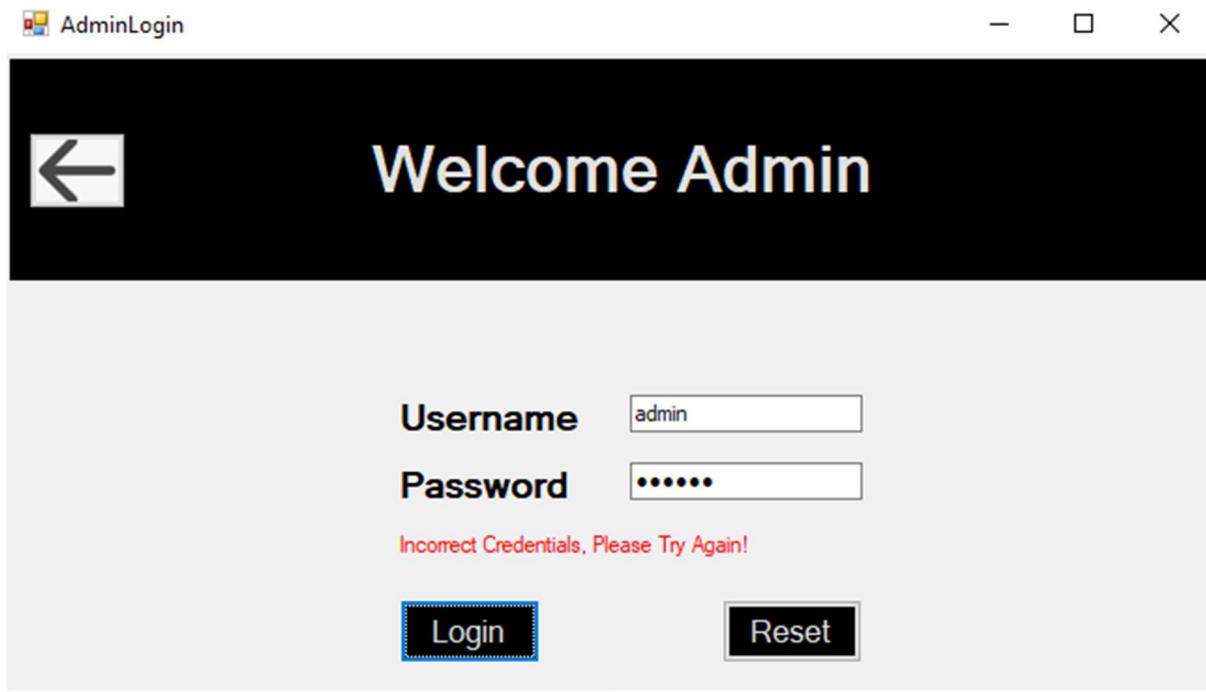
        cmd.ExecuteNonQuery();

        MessageBox.Show("Query Added! Project Manager will try to resolve soon!");
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message, "Error");
    }
    finally
    {
        con.Close();
    }
}

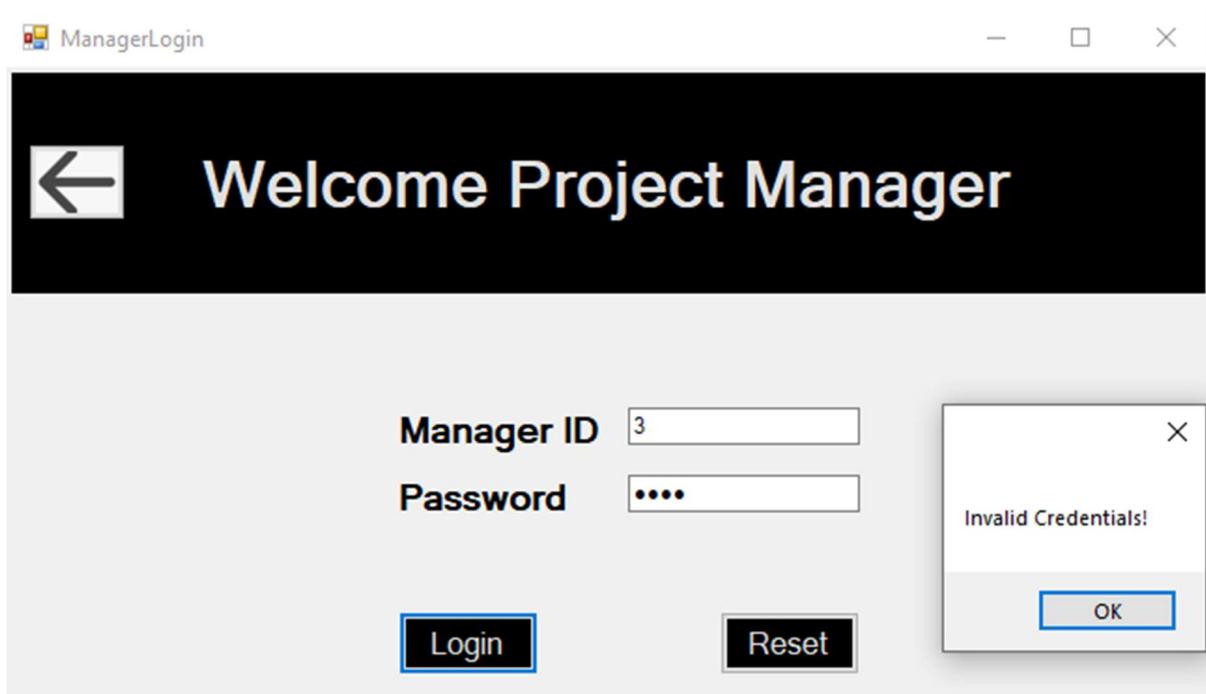
```

Function Prototype 4: Addition of Query by Employee

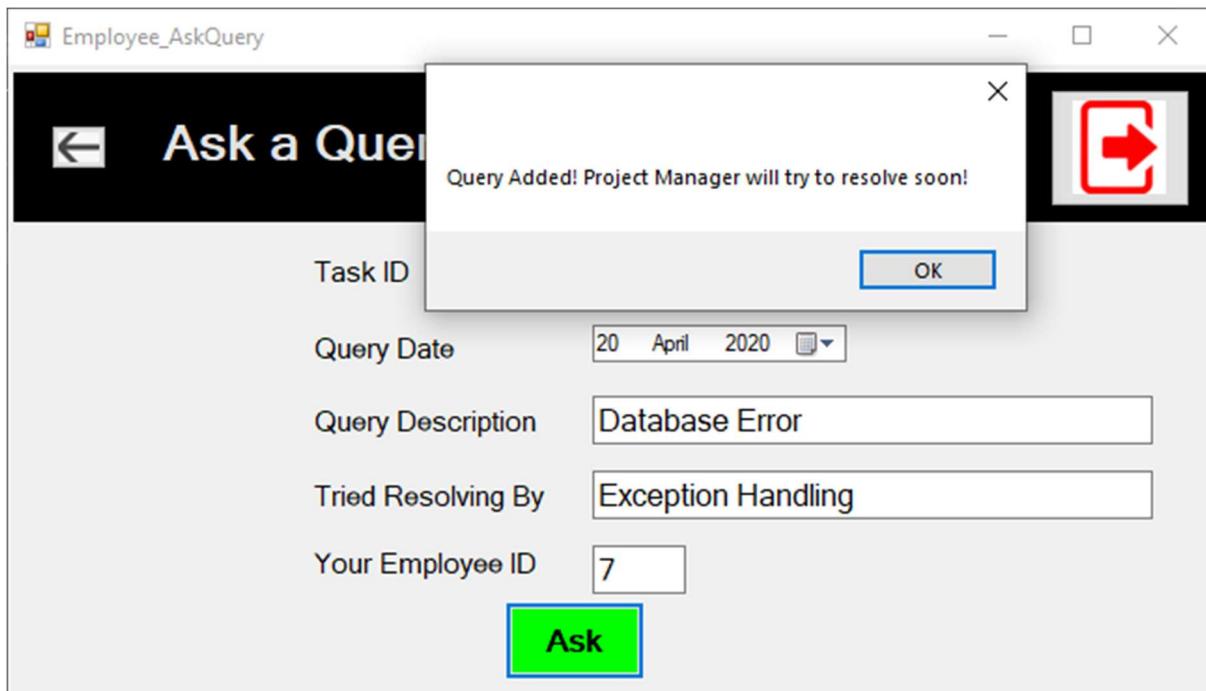
CHAPTER 5 TESTING



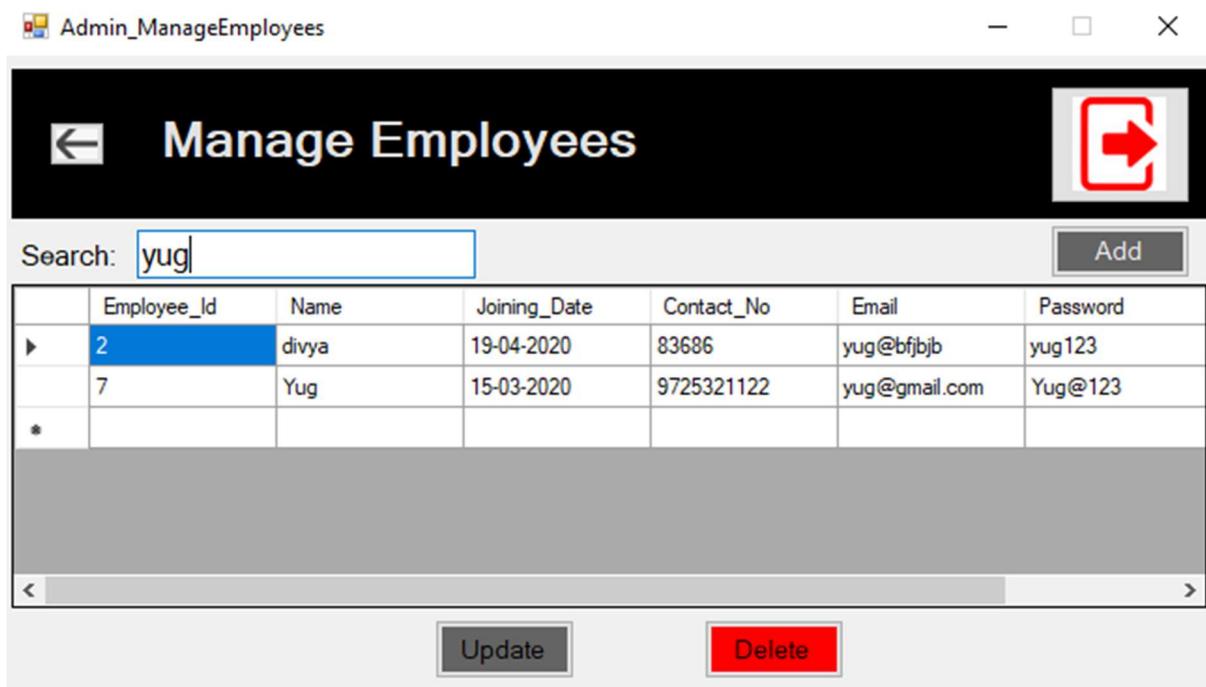
Testing 1: Error Label shown when Admin enters incorrect password



Testing 2: Message Box with appropriate message indicating false login attempt



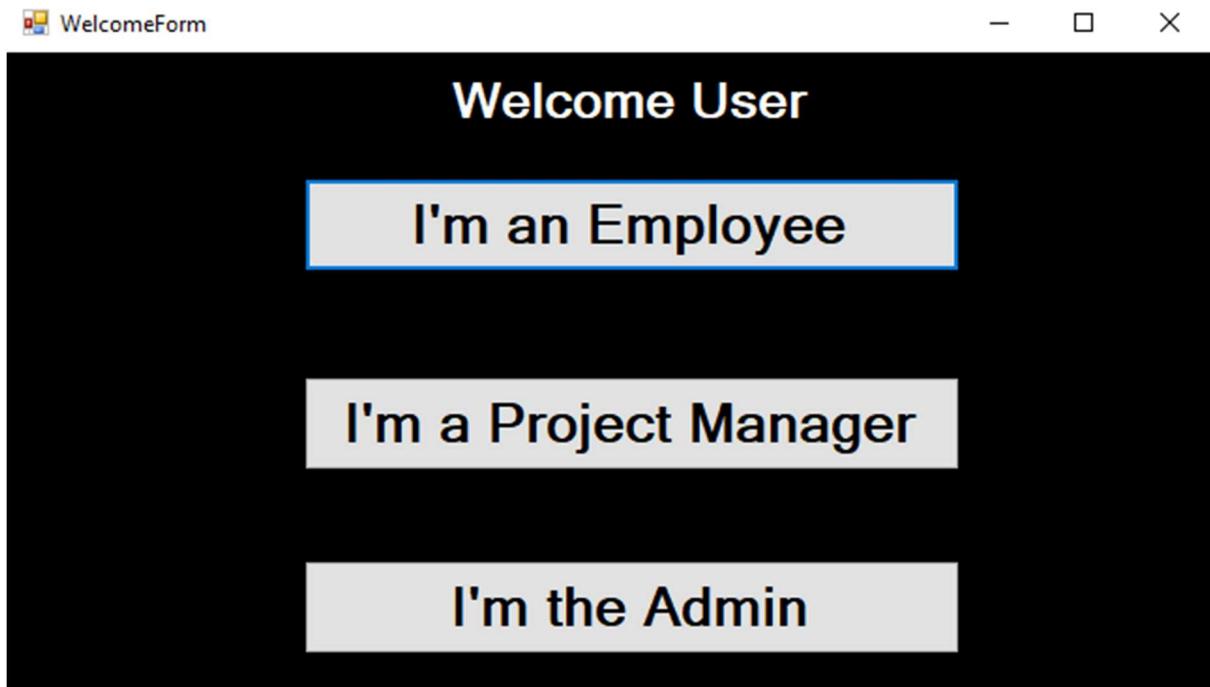
Testing 3: Successful addition of a Query by an Employee



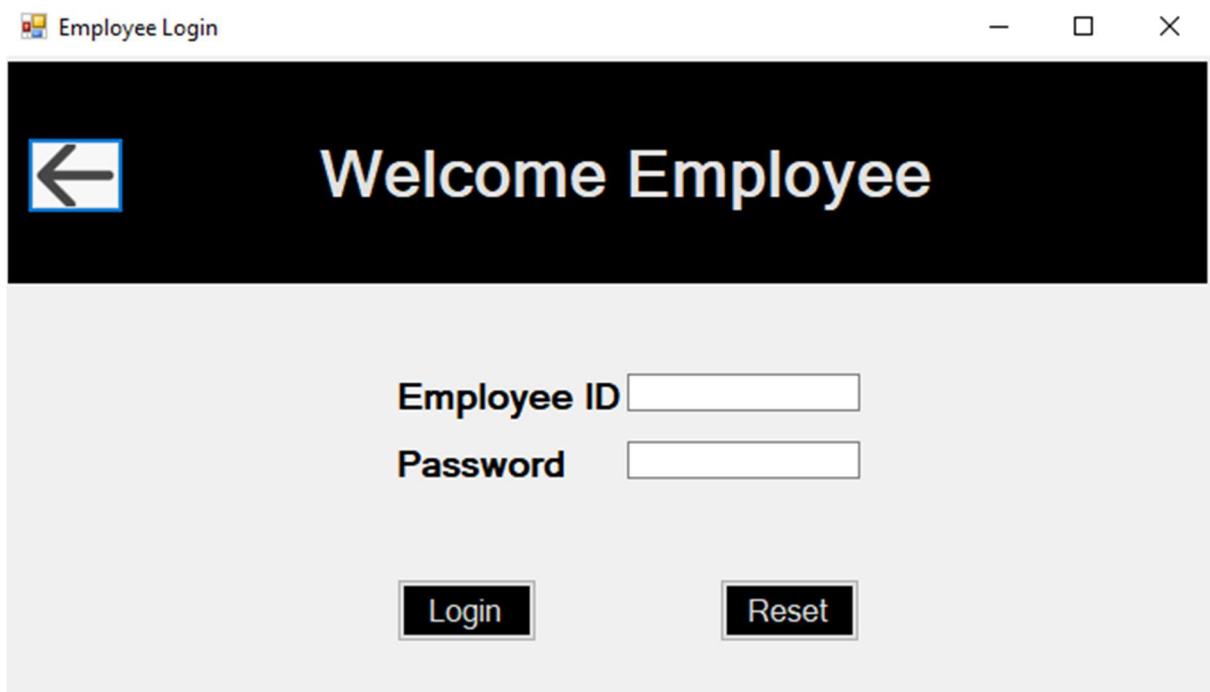
Testing 4: Live search using keyword matching with various fields (Name and Email here)

CHAPTER 6

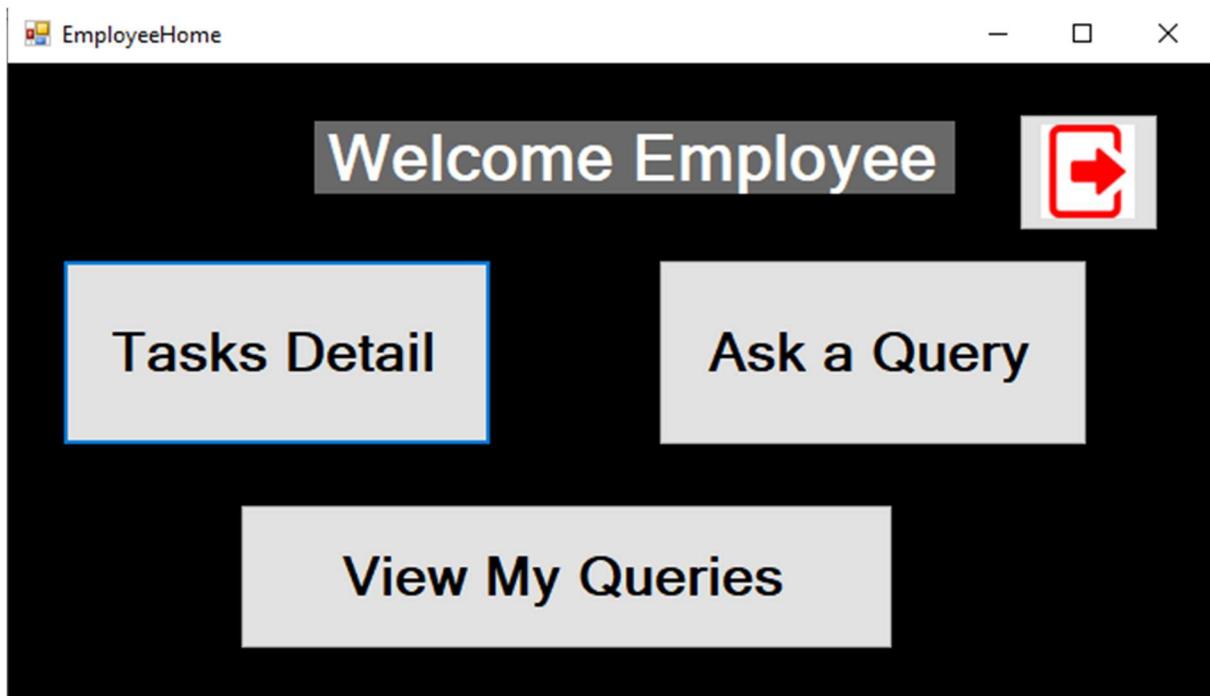
SCREENSHOTS



Screenshot 1: Welcome Page of the Application



Screenshot 2: Login Page for Employee Module

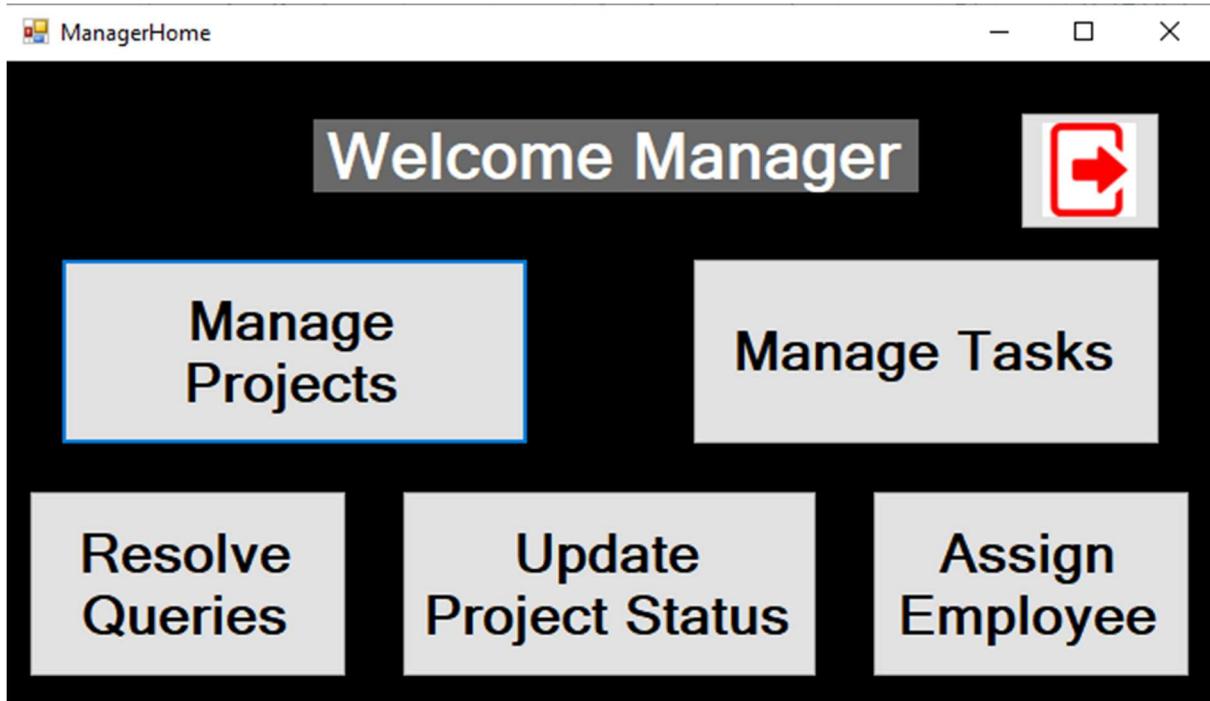


Screenshot 3: Home Page of an Employee after Successful Login

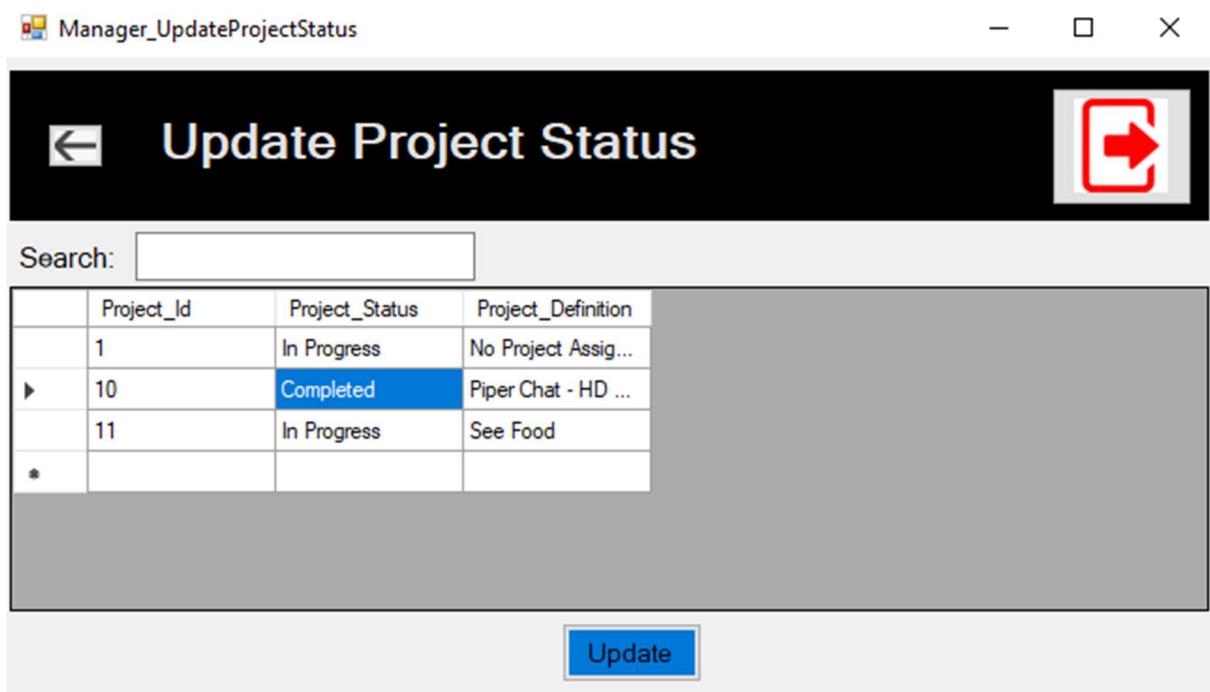
A screenshot of a Windows application window titled "Employee_TaskDetails". The window has a black header bar with a back arrow icon and a red square icon with a white arrow pointing right. Below the header is a text input field labeled "Your Employee ID:" followed by a text box. A table displays task details. The table has columns: Task_Id, Project_Id, AssignedTo_User_Id, AssignedBy_User_Id, Task_Definition, and Task_StartDate. Two rows are visible: one for Task_Id 2 assigned to User 1, and another for Task_Id 3 assigned to User 10. At the bottom is a blue "Update" button.

	Task_Id	Project_Id	AssignedTo_User_Id	AssignedBy_User_Id	Task_Definition	Task_StartDate
▶	2	1	7	1	Trial Task	19-04-2020
	3	10	7	3	Create ER Diagram	21-04-2020

Screenshot 4: Employee's Task Details Page



Screenshot 5: Welcome Page for Manager Describing All Available Features



Screenshot 6: Page for updating Project Status of an Ongoing Project

CHAPTER 7

CONCLUSION

The functionality implementation in the system was done after understanding all the system modules according to their particular requirements.

Functionalities that are successfully implemented are:

- Login
- Registration
- Manage Employees/Managers
- View Tasks
- Ask Query
- Resolve Query
- Update Project Status
- Assign Task
- Assign Employee to Project

After the successful implementation of the system along with all the modules and functionalities, comprehensive testing was performed to determine the loopholes and possible flaws/errors in the system.

CHAPTER 8

LIMITATIONS AND

FUTURE

ENHANCEMENTS

8.1 Limitations

- Admin is required to handle the app.
- Admin should have basic knowledge to handle the app.
- Tasks are not further divided into sub-tasks.
- It is Desktop Based Hardware Dependent.

8.2 Future Extension

- Mobile based notification alerts.
- Email and SMS based OTP verification.
- Chat facility for easy and fast communications of employees/managers.

CHAPTER 9

BIBLIOGRAPHY

- Stack Overflow:

<https://stackoverflow.com/>

- Microsoft Docs:

<https://docs.microsoft.com/en-us/dotnet/framework/winforms/>

- C Sharp Corner:

<https://www.c-sharpcorner.com/technologies/windows-forms>

- Wikipedia:

https://en.wikipedia.org/wiki/Windows_Forms