

PAYROLL MANAGEMENT SYSTEM

A Project Report Submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

Of

Bachelor of Technology

in

Computer and Communication Engineering

by

Kurmilla C S S Kashyap, Sanjana Mishra, S. Lohith
210953058, 210953160, 210953134

Under the guidance of

Mr. Akshay K C (Lab Faculty 1)
Designation of Faculty
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

Mrs. Pooja (Lab Faculty 2)
Designation of Faculty
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India



MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

A Constituent Unit of MAHE, Manipal

May 2023

ABSTRACT

Payroll Management System is a thorough database management system project that automates payroll processing and makes it more efficient. Through this project we aim to enhance the security, efficiency and accuracy of the payroll process and by extension, reduce costs and increase employee satisfaction.

A centralised and secure relational database management system is used to store, manage and process payroll-related data. We incorporate various functionalities such as tax deductions and allowances, employee information management and tracking of attendance.

Through this, the system offers a hassle-free access to payslips and other important details to employees, a streamlined way to keep a track of employees and update specific details to managers and a simplified manner in which the admin can maintain a record of old and new employees in an organisation.

ACM TAXONOMY:

[Information Systems]: Data management systems, Information retrieval

SDG:

This project aims to achieve 2 sustainable development goals:

- Decent work and economic growth: because through a streamlined and automated there is a sense of uniformity in an organisation and the fact that the employees can view their detailed payslips brings about a sense of transparency. This encourages economic growth.
- Reduced inequalities: because the employees can view their detailed payslips and tax policies and always raise their voice if they face any inequality with strong evidence supporting them.

Table of Contents

- List of tables
- List of figures
- Chapter 1: Introduction
- Chapter 2: Literature Survey/Background
- Chapter 3: Objectives/Problem Statement
- Chapter 4: Data design (ER Diagram, Reduction, Schema Diagram, Normalization with suitable justifications)
- Chapter 5: Methodology (Implementation Details with block diagram to explain the project in detail. Do not put code here.)
- Chapter 6: Results
- Chapter 7: Conclusion and Future Work
- Chapter 8: References

List of Tables

- Salary(sal_id int,
state_tax float,
fed_tax float,
hour_pay float ,
gross_salary number(10,2),
primary key (sal_id));
- Status(position varchar(15),
primary key (position));
- Employee(emp_id int,
emp_name varchar(20),
address varchar(50),
email varchar(30),
hire_date date,
sal_id int ,
position varchar(15),
primary key (emp_id),
foreign key (sal_id) references salary on delete cascade,
foreign key (position) references status on delete cascade);
- Allowance(allowance_id varchar(5),
allowance_name varchar(15),

allo_description varchar(100),
primary key (allowance_id));

- Deduction(deduction_id varchar(5),
deduction_name varchar(15),
deduc_description varchar(100),
primary key (deduction_id));
- Allowance_details(allowance_id varchar(5),
emp_id int,
allow_amount float,
allow_eff_date date ,
primary key (allowance_id) ,
foreign key (allowance_id) references allowance on delete cascade,
foreign key (emp_id) references employee on delete cascade);
- Deduction_details(deduction_id varchar(5),
emp_id int,
deduction_amount float,
deduct_eff_date date ,
primary key (deduction_id) ,
foreign key (deduction_id) references deduction on delete cascade,
emp_id int,);
- Payroll_slip(emp_id int,
salary float,
no_of_leaves int ,
salary_per_day float,
deductions float,
allowances_per_leaves float ,
net_salary float,
primary key (emp_id),
foreign key (emp_id) references employee on delete cascade);
- Employee_cred(emp_id int,
emp_password varchar(10),
primary key (emp_id),
foreign key (emp_id) references employee on delete cascade);

List of Figures

- ER diagram
- Design process of Admin module (block diagram)
- Design process of manager module (block diagram)
- Design process of employee module (block diagram)
- Result image 1
- Result image 2
- Result image 3
- Result image 4
- Result image 5
- Result image 6
- Result image 7
- Result image 8
- Result image 9
- Result image 10
- Result image 11
- Result image 12
- Result image 13

Abbreviations:

No such abbreviations have been used in the report apart from a few attributes that have been defined later on, in the data design chapter (chapter 4).

CHAPTER 1

INTRODUCTION

The project report outlines the development of a payroll management system that utilises the database design and user interface. This system aims to make an organisation's task of managing employee salaries, deductions and allowances as well as maintaining a record of all the employees, easier.

Our database design includes tables for employee, salary, deductions and allowances amongst others. We created a user-friendly interface that allows access to relevant information along with updation and retrieval of data and makes these tasks hassle-free. In general, this project shows how, with the use of database design and ui creation together, can enhance payroll processing in an organisation.

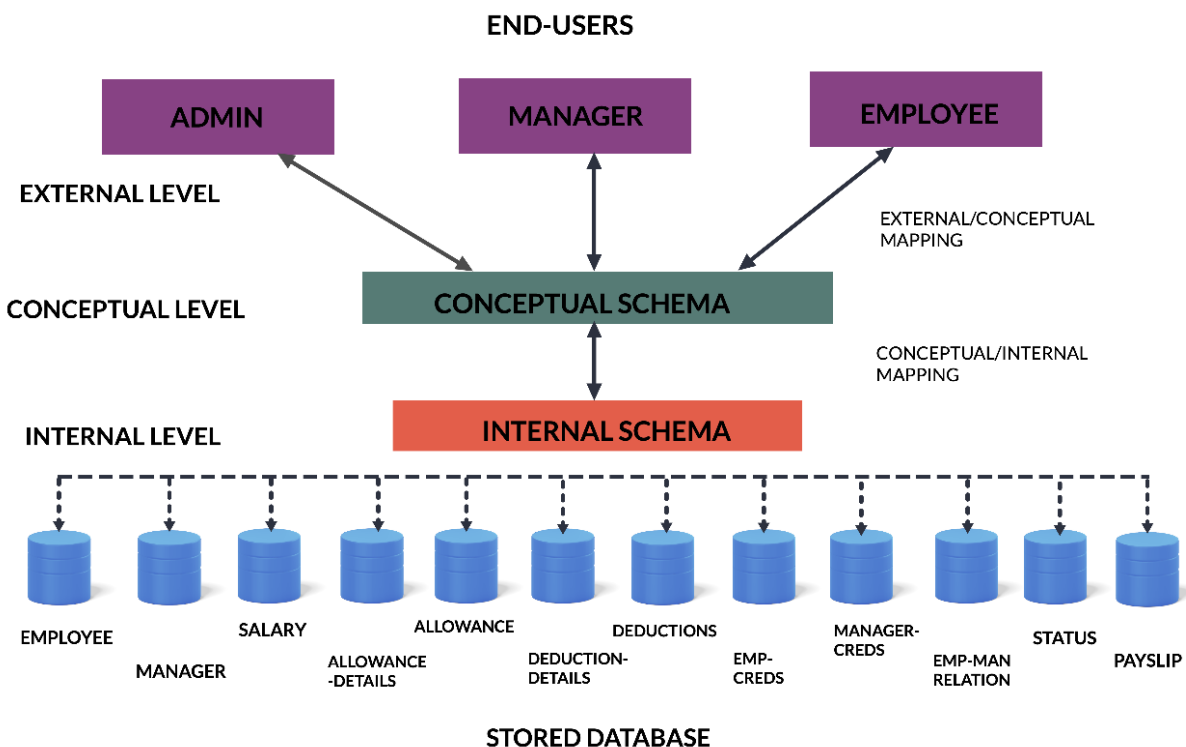


Figure 1.1: Three-tier architecture

We made use of SQLplus (Oracle 19c version) for our backend where we created and stored our database along with adding a few procedures and triggers.

Visual Studio 2022 (c#) was used for our frontend work and connectivity.

We have 3 models/user interfaces:

- Admin
- Manager
- Employee

Chapter 2

Literature Survey/Background

Literature Survey:

[1] P. A. Arunprasad and G. K. Padmavathi, Design and Development of Online Payroll Management System. International Journal of Computer Science and Information Technologies (IJCSIT), 2014

brings focus to the development of an online payroll processing system and its importance in today's world. The system's aim is to automate the processing of payrolls in organisations and how that would enhance the user experience. The paper provides an insight into the system's architecture and feature by outlining the use of database design and user interface creation. It describes the various technologies and programming languages used to design this model. The authors highlight how using an online website would increasing efficiency and security, reduce paperwork and human errors and by extension, reduce costs. They conducted a test and took feedback from the users which shows that the general audience would prefer an online payroll processing system over the manual method.

Background:

Capabilities of this system include:

- Logging in with a username and password for each of the 3 modules.
- Option to add or delete new employees whenever a change is made in the number of employees. This power is only held by the admin.
- The manager can view his employees and their details and change the salary and number of leaves.
- The employee can only view his details and also generate payslips. He can also generate a report for specified periods of time.

Chapter 3

Objectives/Problem Statement

Problem statement:

Organisations face a lot of problem in managing the details of a huge number of employees. For a long time, there was a dependency on manual systems which is low on security, time-consuming and prone to errors. This leads to incorrect salary amounts being received by some employees, irregular data of employees and increased costs.

To overcome these issues, a digital system is needed which would keep a timely update of employee data, process the payrolls and keep a track of the changes (if any) in the deductions and allowances. Through this project, we aim to create an online payroll system which would do all of these things with the use of database design and UI creation to enhance the efficiency of an organisation, resulting in employee satisfaction and reduced costs.

Objectives:

- Design a system that processes payroll with close to null errors, keep a track of any change in the salary policies and update any changes the count of employees.
- Create a hassle-free and user-friendly UI that would allow employees to view their details and access their payslips, managers to view employee details and make any necessary changes in their salary or leaves and admin update the employee database if any changes are made.
- Create a database that accurately stores and maintains employee information, salary and tax details among various other things.
- Improve the overall efficiency of an organisation by reducing errors and costs through a digitised system.

Chapter 4

Data Design

ER Diagram

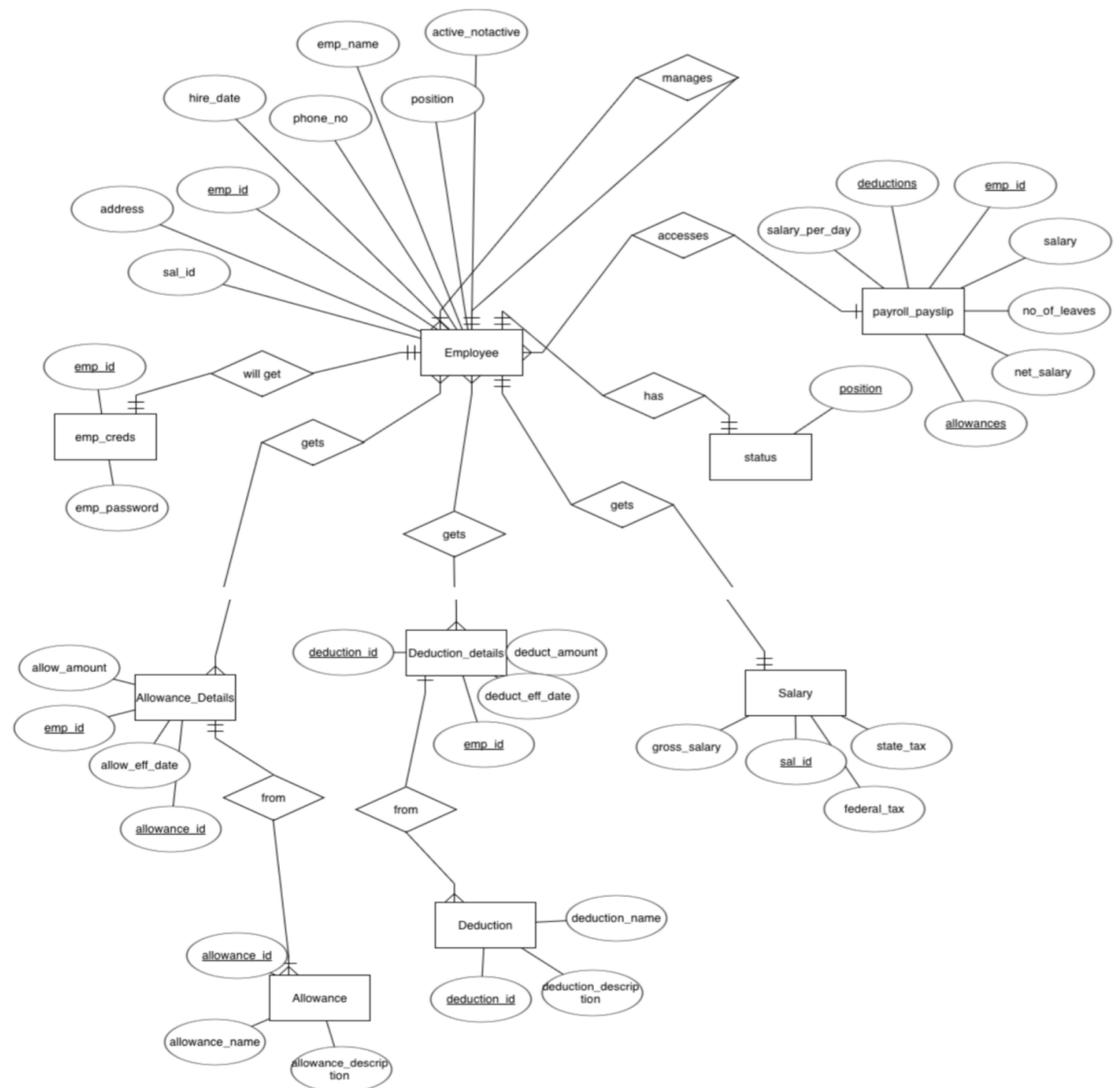


Figure 4.1: ER diagram

Relational Schema

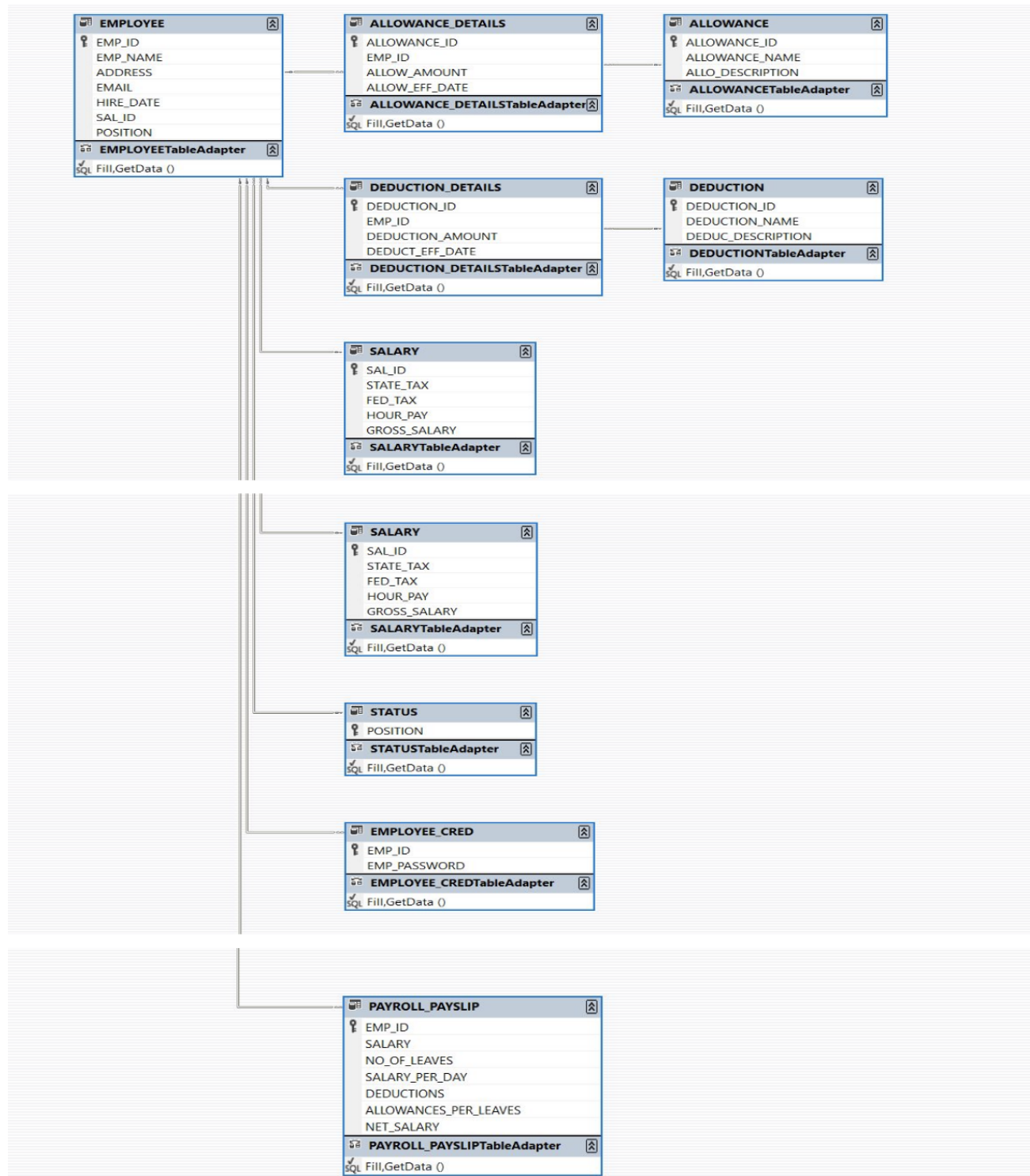


Figure 4.2: Relational schema

Normalisation:

The primary keys of all tables are as listed below:

- salary

Primary Key: sal_id

- status

Primary Key: position

- employee

Primary Key: emp_id

- allowance

Primary Key: allowance_id

- deduction

Primary Key: deduction_id

- allowance_details

Primary Key: allowance_id, emp_id

- deduction_details

Primary Key: deduction_id, emp_id

- payroll_payslip

Primary Key: emp_id

- emp_cred

Primary Key: emp_id

1NF:

To prove that the tables are in 1NF, we need to show that all the attributes in the table are atomic(indivisible). All the attributes in all of the tables above are atomic, proving that the database is in 1NF

2NF:

To prove that the tables are in 2NF there are 2 conditions that need to be satisfied:

1. The table must be in 1NF
2. All non-key attributes should be fully functionally dependent on the whole primary key.

salary

- The table has a primary key (sal_id)
- All non-key attributes are fully functionally dependent on the primary key.

status

- The table has a single attribute (position) as the primary key.

employee

- The primary key here is emp_id.
- Functional dependencies:
 - emp_id -> (emp_name, address, email, hire_date, sal_id, position)
 - sal_id -> (state_tax, fed_tax, hour_pay, gross_salary)
- The table has a transitive dependency where position depends on emp_id.
- To remove the transitive dependency and achieve 2NF, we can create a separate table

emp_position(emp_id, position)

the functional dependency for the following is:

emp_id->position

- The modified employee table with the new foreign key will be in 2NF.

allowance

- The table has a primary key (allowance_id)
- All non-key attributes are fully functionally dependent on the primary key.
- It is in 2NF because there are no partial dependencies.

deduction

- The table has a primary key (deduction_id)
- All non-key attributes are fully functionally dependent on the primary key.

allowance_details

- The table has a composite primary key (allowance_id, emp_id).
- All non-key attributes are fully functionally dependent on the composite key.

deduction_details

- The table has a composite primary key (deduction_id, emp_id).
- All non-key attributes are fully functionally dependent on the composite key.

payroll_payslip

- The table has a primary key (emp_id)
- All non-key attributes are fully functionally dependent on the primary key.

employee_cred

- The table has a primary key (emp_id)
- All non-key attributes are fully functionally dependent on the primary key.

From the above points, it's clear that the tables are in 2NF after any reduction necessary.

3NF:

To prove that the database is in 3NF, following conditions need to be satisfied:

1. The table is in 2NF
2. There is no transitive dependency for a non-prime attribute.

The tables satisfy these conditions and so, they are in 3NF.

Reduction:

Along with the 9 main tables, we will have 4 additional tables, after considering all the normal forms and reduction:

- manager(man_id, man_name, address, phone_no, hire_date, position, email)
- emp_man_relation(emp_id, man_id)
- man_creds(man_id, man_password)
- emp_position(emp_id, position)

Chapter 5

Methodology

We created our database on SQLplus. We made several tables and formed entity relationships. We have also made use of triggers and procedures to show some of the functionalities of payroll management system.

For our frontend, we made use of Visual Studio 2022 version. Various buttons, tabs and dropdown boxes are used in a clean format.

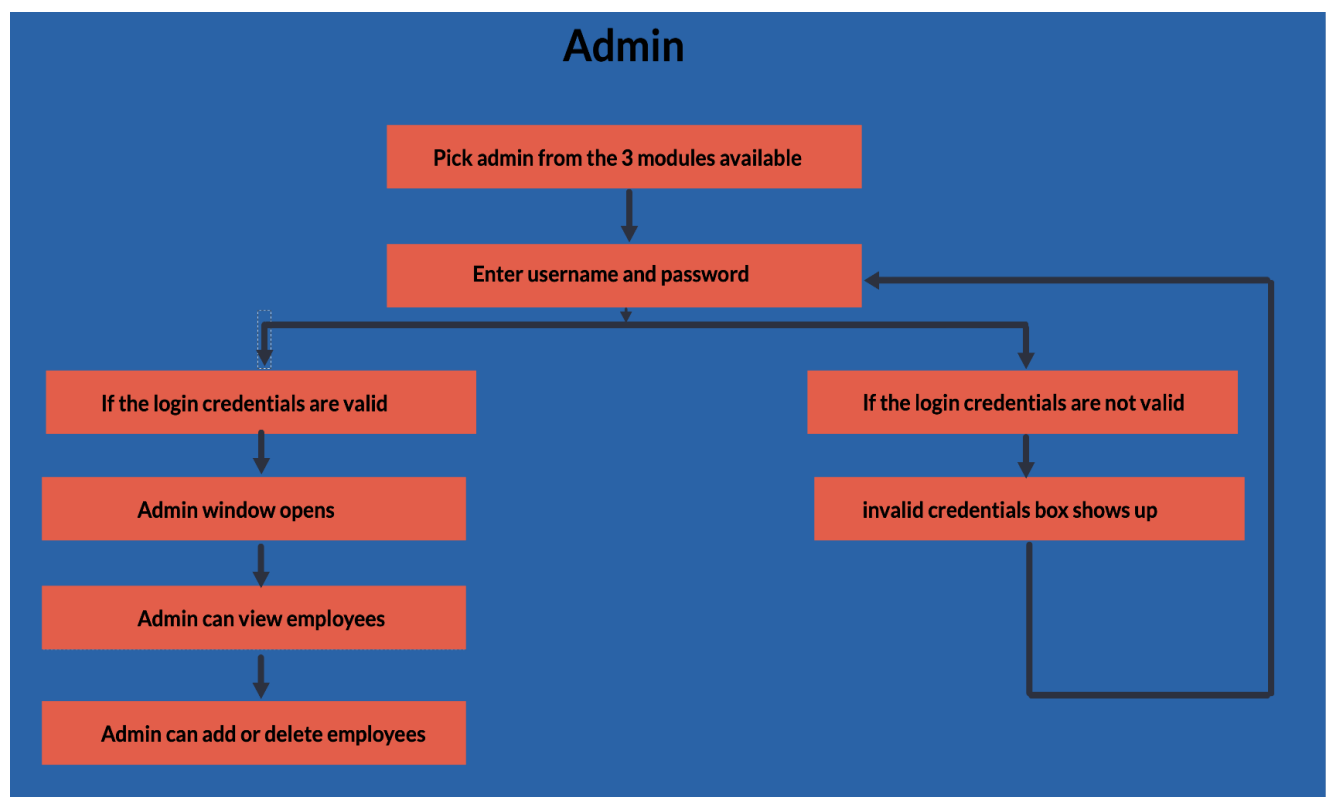


Figure 5.1: Design process of Admin module

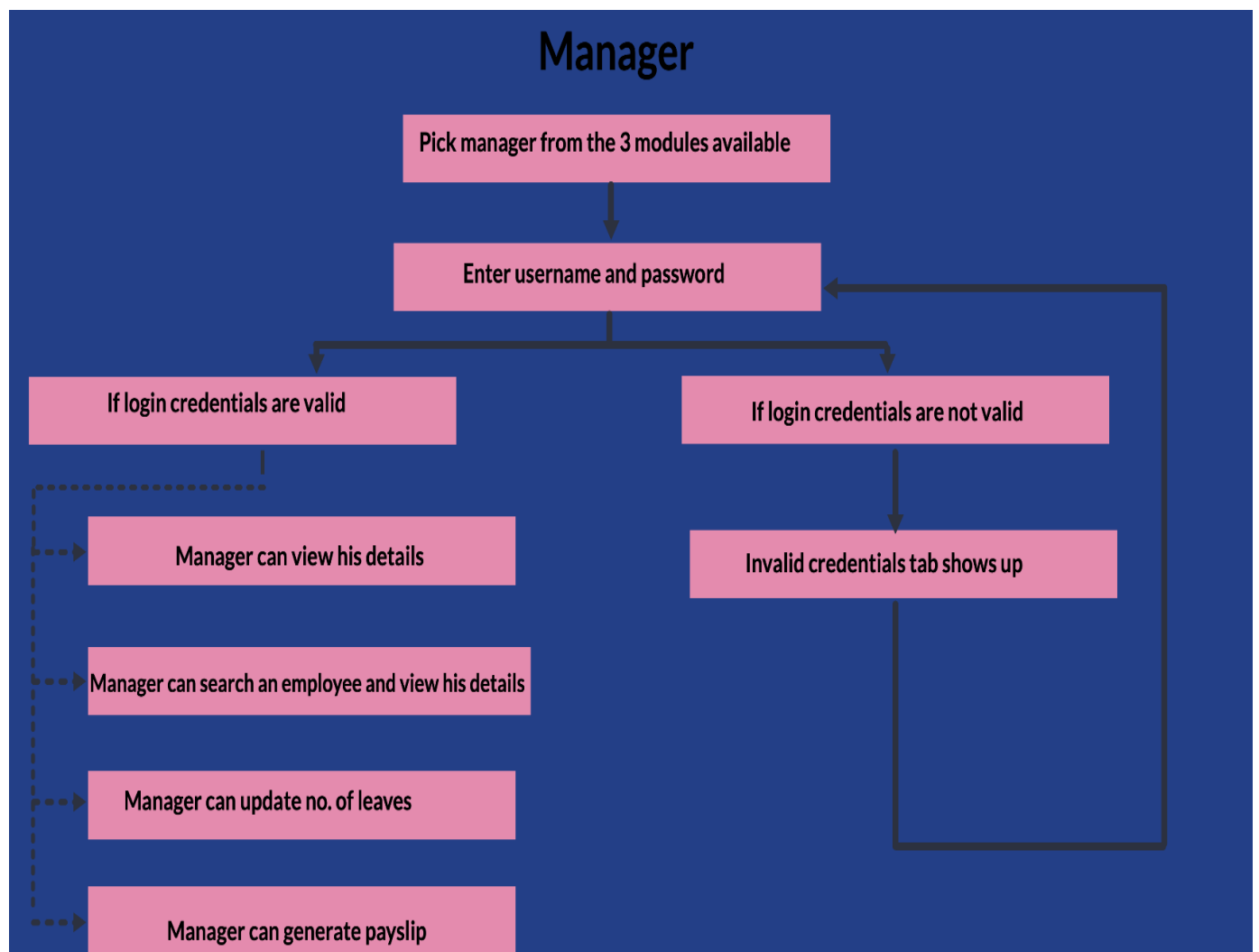


Figure 5.2: Design process of Manager

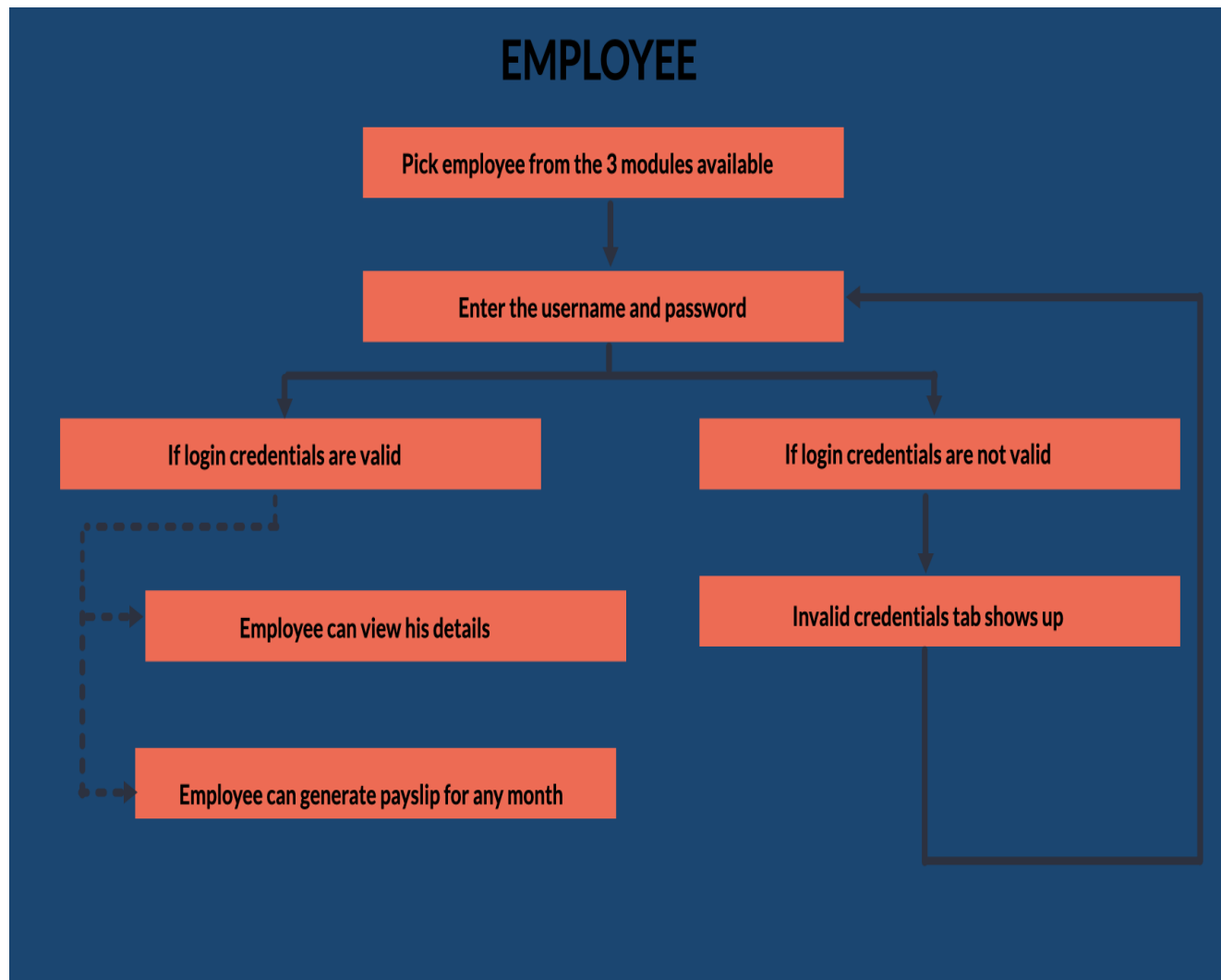
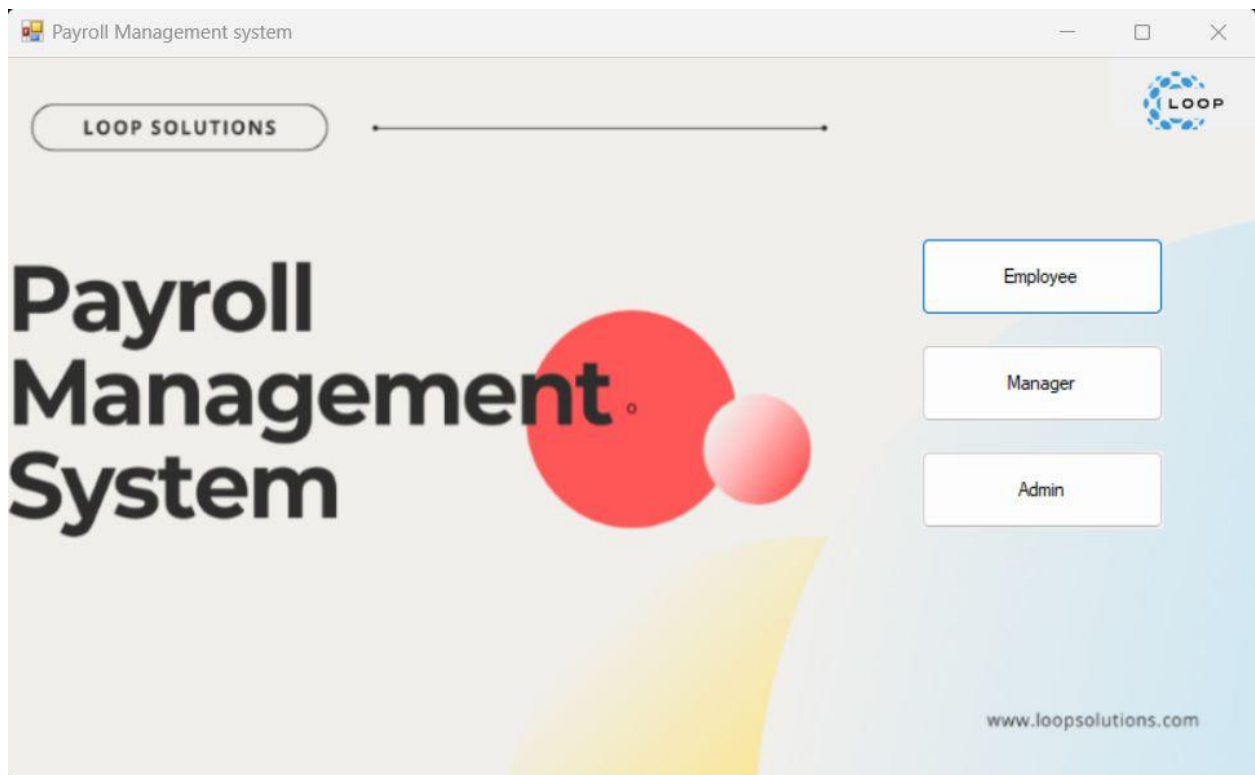


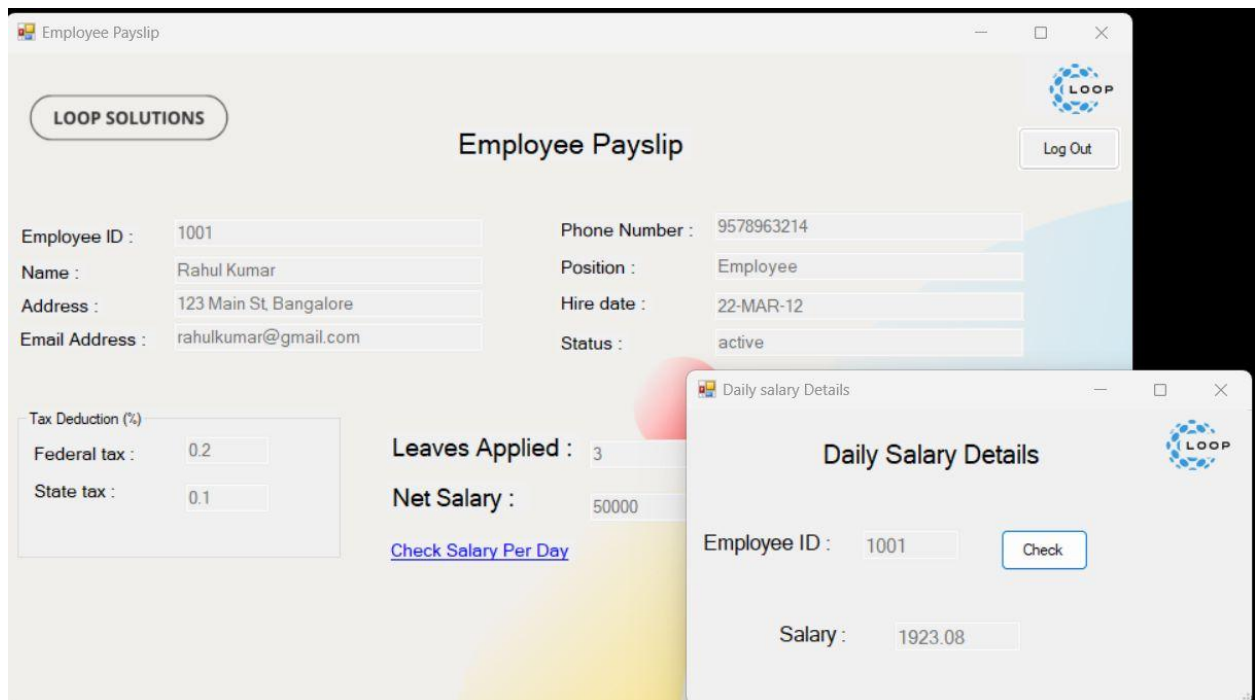
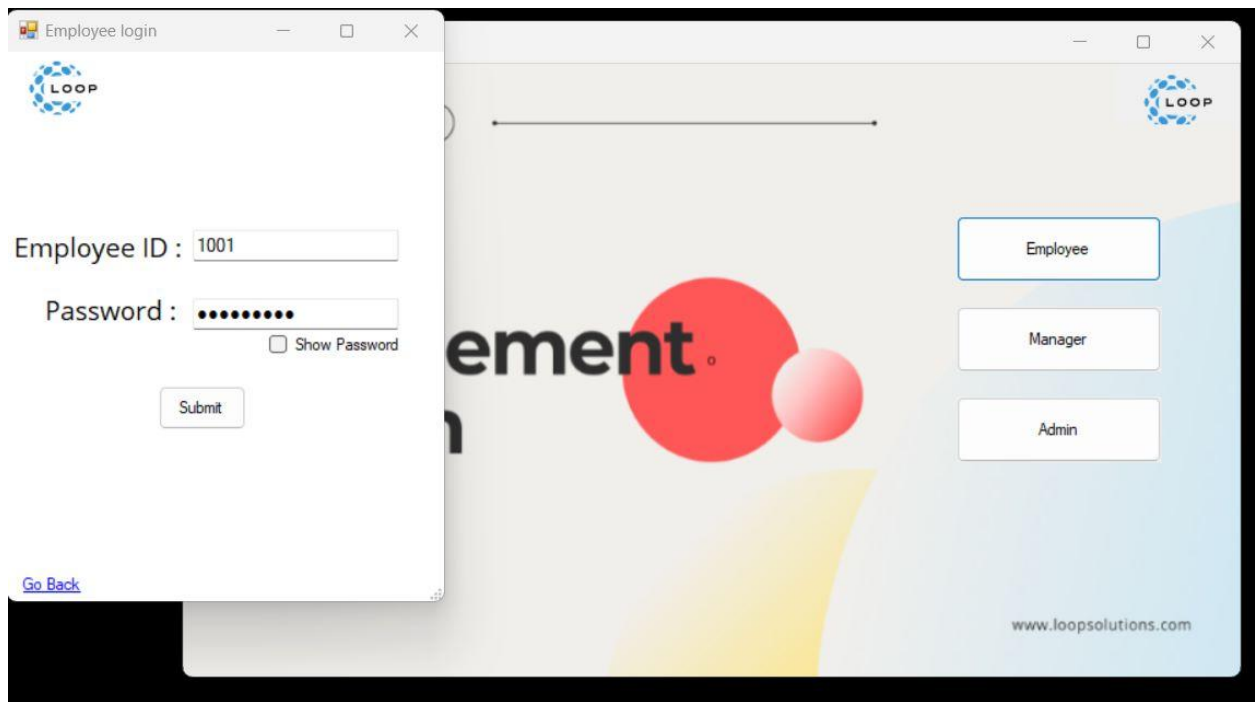
Figure 5.3: Design process of Employee

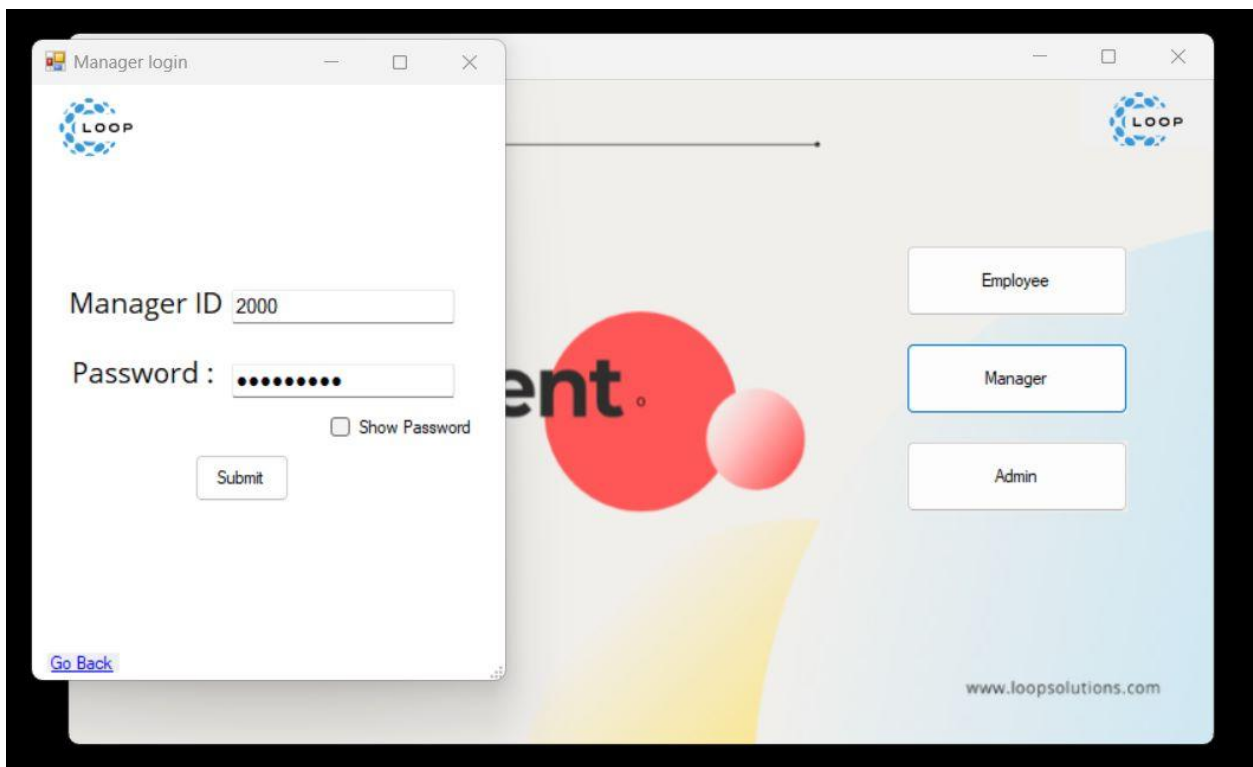
Chapter 6

Results

Result images:







Manager Personal Information

LOOP SOLUTIONS

Manager Payslip

[Exit Payslip](#)

Manager ID : 2000

Name : Aarav Gupta

Address : 123 Main St, Bangalore

Email Address : aaravgupta@gmail.com

Phone Number : 8977445566

Position : Manager

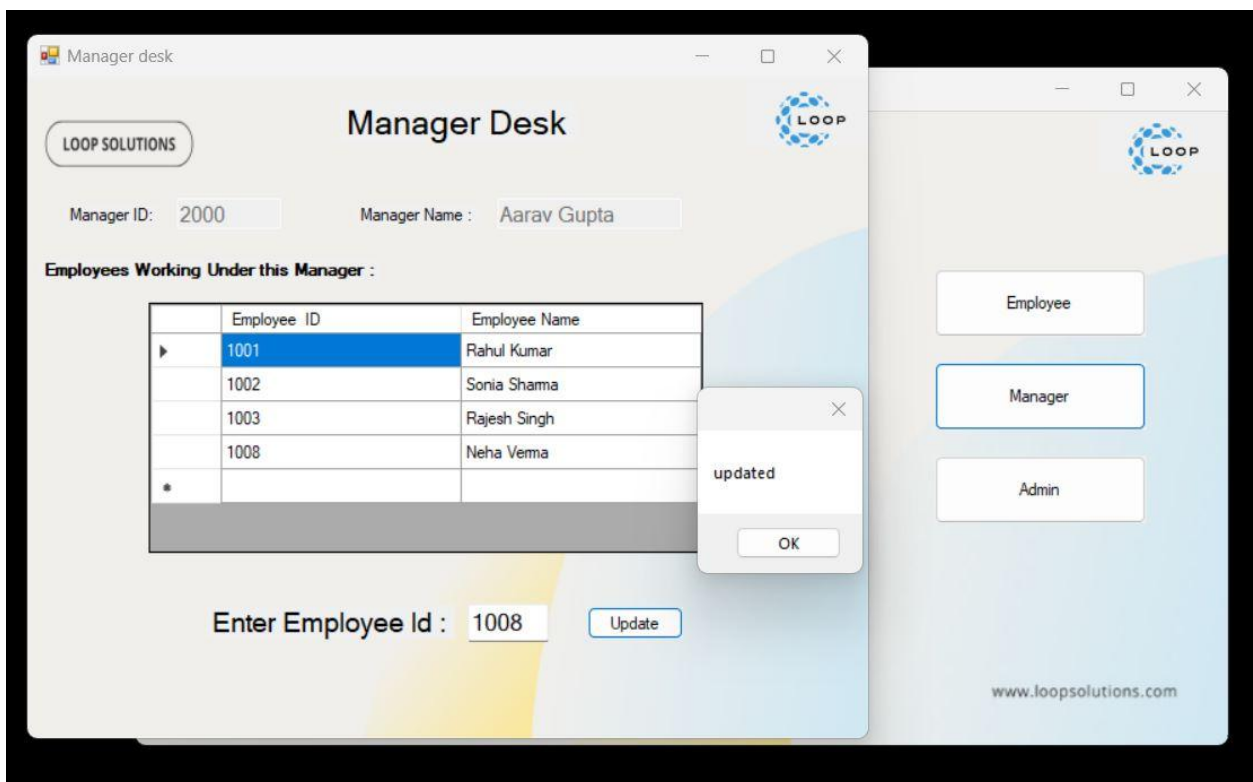
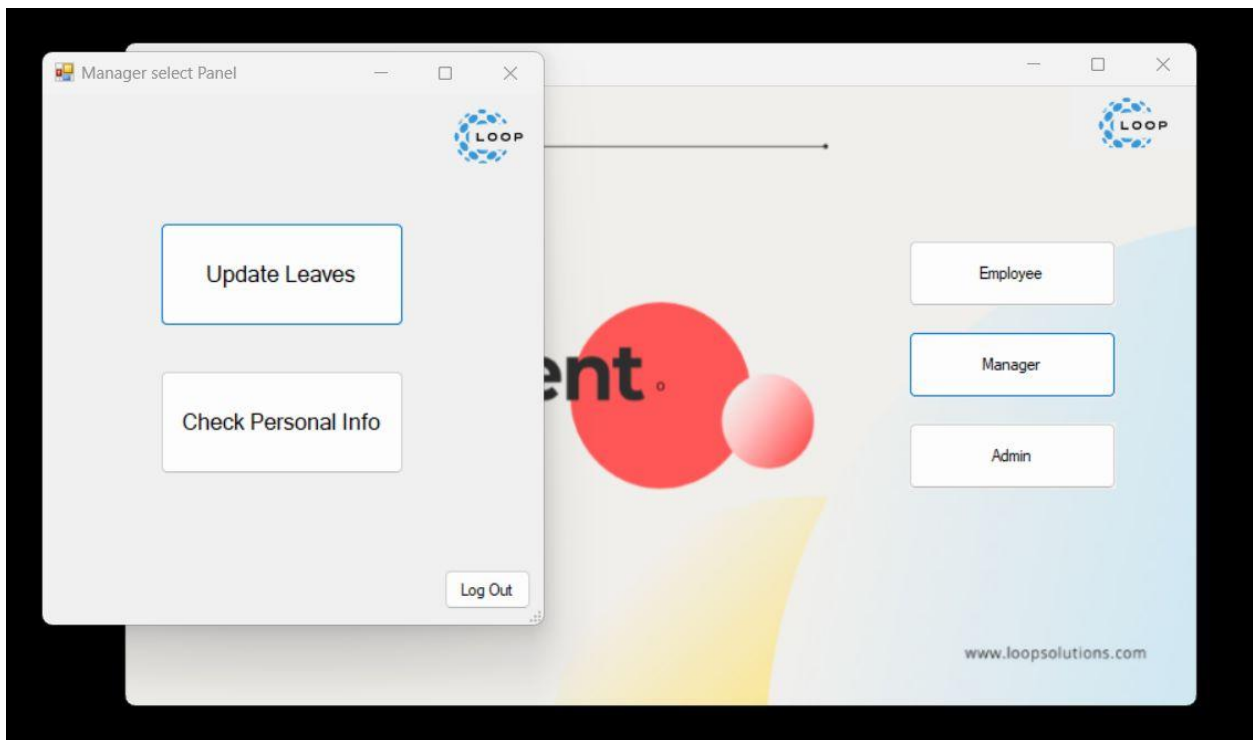
Hire date : 22-MAR-12

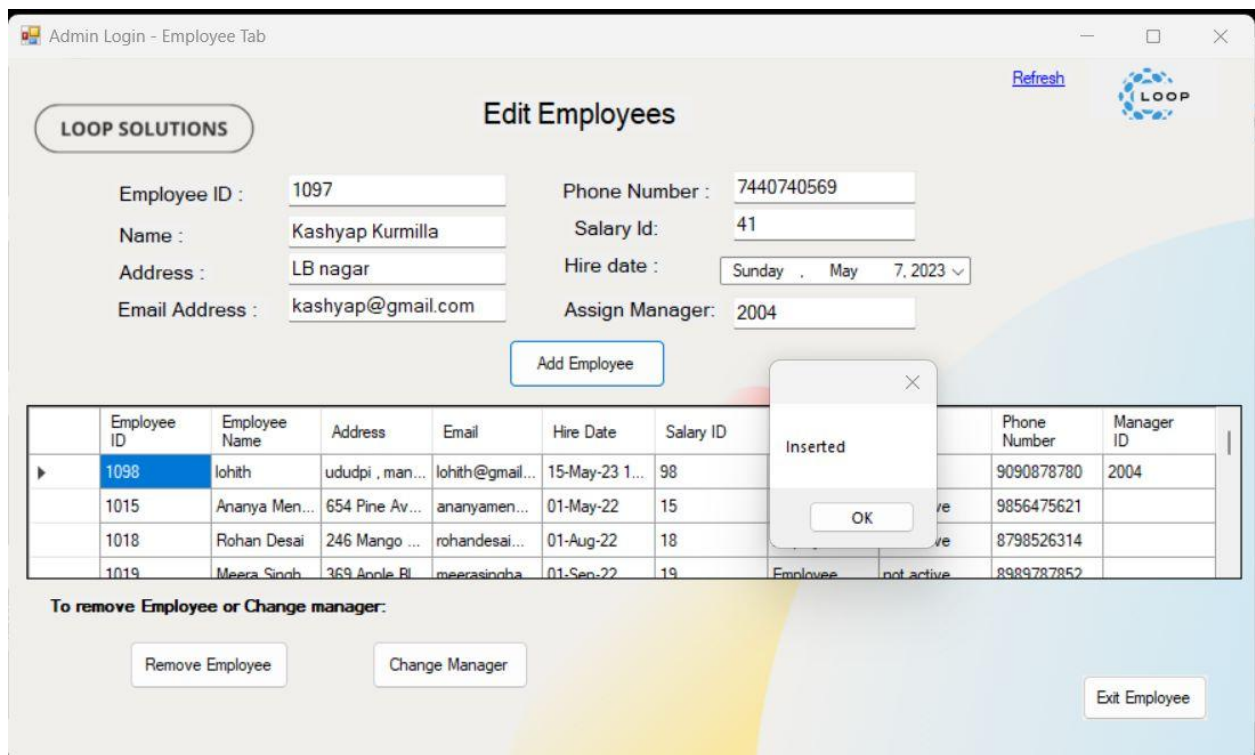
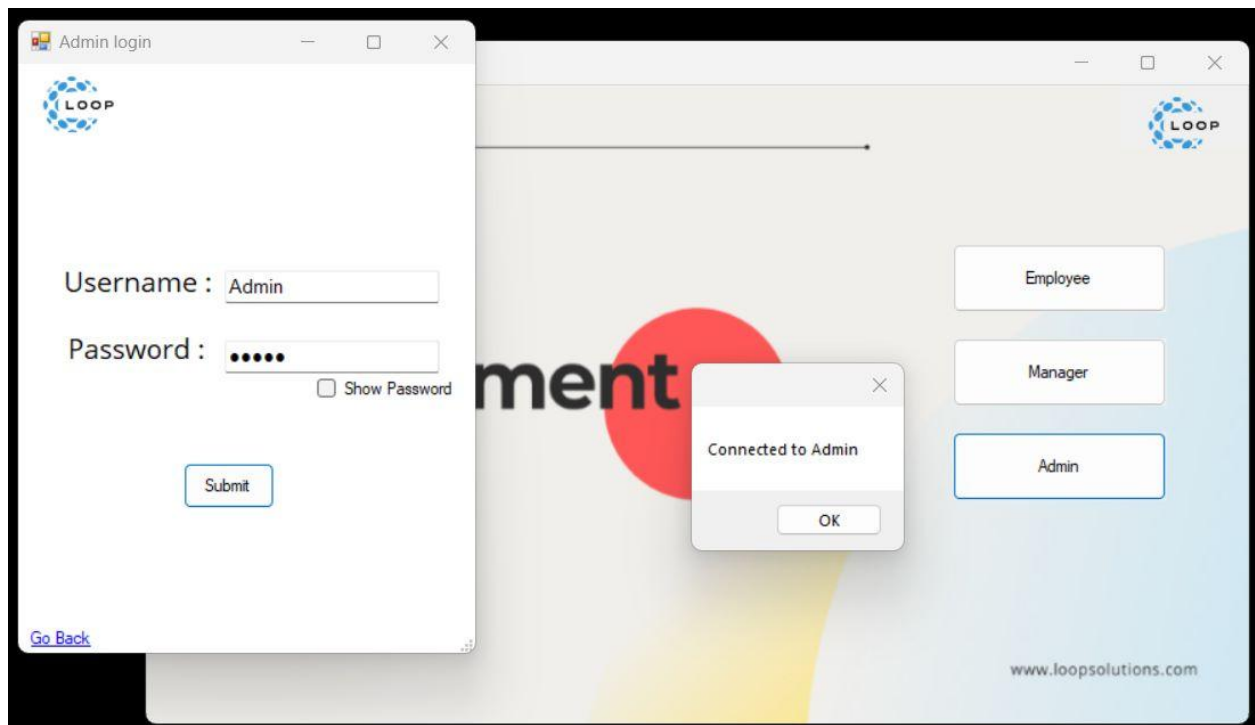
Tax Deduction (%)

Federal tax :	0.18
State tax :	0.05

[Check Deduction details](#)

Net Salary : 45000





Enter Employee Number

Employee ID :

Delete

Refresh

LOOP

Edit Employees

Phone Number :

Salary Id:

Hire date : , 7, 2023

Assign Manager:

Add Employee

Record deleted successfully

OK

	Employee ID	Employee Name	Address	Email	Hire Date	Salary ID	Phone Number	Manager ID
	1058	abhiram	banglore	abhiram@g...	04-May-23 1...	32	7383939393	2006
	1097	Kashyap Kur...	LB nagar	kashyap@g...	07-May-23 1...	41	7440740569	2004
*								

To remove Employee or Change manager:

Remove Employee

Change Manager

Exit Employee

Change Manager

Employee ID :

Manager ID :

Update

Refresh

LOOP

Edit Employees

Phone Number :

Salary Id:

Hire date : , 7, 2023

Assign Manager:

Add Employee

Record updated successfully

OK

	Employee ID	Employee Name	Address	Email	Hire Date	Salary ID	Phone Number	Manager ID
	1058	abhiram	banglore	abhiram@g...	04-May-23 1...	32	7383939393	2006
	1097	Kashyap Kur...	LB nagar	kashyap@g...	07-May-23 1...	41	7440740569	2004
*								

To remove Employee or Change manager:

Remove Employee

Change Manager

Exit Employee

Admin Login - Manager tab

Refresh

LOOP SOLUTIONS

Edit Managers

Manager ID : 2009
Name : Kurmilla kashyap
Address : LB Nagar
Email Address : kashyap@gmail.com

Phone Number : 7330740569
Salary Id: 43
Hire date : Thursday , May 18, 2021

Insert Manager

Manager ID	Manager Name	Address	E-Mail	Hire Date	Salary ID	Phone Number
2000	Aarav Gupta	123 Main St, Ba...	aaravgupta@gm...	22-Mar-12	101	8977445566
2001	Mehul Sharma	456 Park Ave, D...	mehulshama@g...	15-May-14	102	9898747456
2002	Kavya Singh	789 Broad St, M...	kavyasingh@gm...	03-Aug-15	103	9874563322
2003	Rohit Kumar	567 1st St, Hyde...	rohitkumar@gma...	22-Sep-17	104	8978987895
2004	Divya Patel	111 Oxford Rd, ...	divyapatel@gmai...	08-Nov-19	105	9856321415

To remove Manager or Change Employee:

Remove Manager

Exit Manager

×

Inserted

OK

Admin Login - Manager tab

Refresh

Remove Manager

Manager ID : 2009

Remove

Edit Managers

Phone Number : 7330740569
Salary Id: 43
Hire date : Thursday , May 18, 2021

Insert Manager

Manager ID	Name	Address	E-Mail	Hire Date	Salary ID	Phone
2000	Aarav Gupta	123 Main St, Ba...	aaravgupta@gm...	22-Mar-12	101	8977445566
2001	Mehul Sharma	456 Park Ave, D...	mehulshama@g...	15-May-14	102	9898747456
2002	Kavya Singh	789 Broad St, M...	kavyasingh@gm...	03-Aug-15	103	9874563322
2003	Rohit Kumar	567 1st St, Hyde...	rohitkumar@gma...	22-Sep-17	104	8978987895
2004	Divya Patel	111 Oxford Rd, ...	divyapatel@gmai...	08-Nov-19	105	9856321415

To remove Manager or Change Employee:

Remove Manager

Exit Manager

×

Record deleted successfully

OK

Chapter 7

Conclusion and Future Work

Conclusion:

In conclusion, this system provides a feasible way to ensure accurate salary calculation, keep up with any change in tax policies and maintain a track of all employees in an organisation. The tables in the schema store employee, salary, payslip and other data and are updated accordingly. Relationship between the tables are well-established with the use of primary key and foreign key constraints which ensures data integrity and consistency, 2 of the biggest advantages of using a database management system.

The payroll management system provides a strong foundation for organisations to work on while reducing errors and costs that would normally go into maintaining a manual system.

Future Work:

While the provided system is quite efficient in processing payrolls and managing employees, it is just the tip of the iceberg. Various programming languages and technological advancements can be used in this field. Biometrics can be used to keep an absence record of all the employees, mobile applications can be created, payment gateways can be integrated and different business management tools can be used for creating an in-depth analysis of the payroll. These are just a few things that can be integrated in a payroll management system in the near-future.

Chapter 8:

References:

- <https://dl.acm.org/ccs>
- <https://www.niti.gov.in/verticals/sustainable-dev-goals>
- <https://sdgs.un.org/goals>
- [1] P. A. Arunprasad and G. K. Padmavathi, Design and Development of Online Payroll Management System. International Journal of Computer Science and Information Technologies (IJCSIT), 2014
- [2] A. Silberchatz, Henry F. Korth and S. Sudarshan, Database System Concepts. 6th edition