**Employees Management System**

***Dissertation Submitted in Partial fulfilment of the Requirement for the Award of the Diploma of***

**Computer Engineering**

**VIth Semester**

**Jan - May, 2023**



**Project Guide: Submitted By:** Nidhi ChawlaMa’amSuraj kumar gupta

**2013051062**

**DSEU Rajokri Campus**

**Department of Training and Technical Education, Delhi**

**2023DECLARATION**

I hereby declare that the project titled “**Employee Management System**” submitted by me for Diploma in Computer Engineering VIth semester to DSEU Rajokri Campus, Department of Training and Technical Education, Delhi, comprises my own work and due acknowledgement has been made in text to all other material used.

Signature of Student:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: **SURJ KUMAR GUPTA**

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CERTIFICATE FROM GUIDE**

It is to certify that the project entitled “**Employee Management System**”, submitted by Mr. **SURAJ KUMAR GUPTA**  to the DSEU Rajokri Campus, Department of Training and Technical Education, Delhi, has been completed under my supervision and the work is carried out and presented in a manner required for its acceptance to *Diploma in Computer Engineering VI* ***th*** *semester.*

**Project Guide**

**Signature:** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name: SURAJ KUMAR GUPTA**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**EXAMINATION APPROVAL CERTIFICATE**

This is to certify that this project report entitled **"EMPLOYEE MANAGEMENT SYSTEM”** submitted to DSEU Rajokri Campus, Delhi is a bonafide record of work done by **“SURAJ KUMAR GUPTA”** under my supervision from "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" to "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_".

**ACKNOWLEDGEMENT**

The success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The completion of any inter-disciplinary project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge.

We are eternally grateful to our teacher **Nidhi Chawla Ma’am** for her even willingness to give us valuable advice and direction under which we executed this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

**SURAJ KUMAR GUPTA**

**2013051062**

**ABSTRACT**

The EMS often incorporates a variety of features and capabilities, including performance management, payroll processing, benefits administration, and employee data management. The system is made to be simple to use, with a user-friendly interface and settings that can be adjusted to match the particular requirements of various businesses.

By automating many of the manual procedures connected with managing personnel data and workflows, an EMS can contribute to reducing administrative workload and increasing productivity. As a result, HR personnel may have more time to devote to strategic responsibilities like talent management and employee development.

There is a tonne of room for EMS to grow in the future, including the incorporation of artificial intelligence and machine learning tools to enhance data analysis and decision-making. To enhance communication and engagement between employees and management, the system could also be enhanced to include features like employee self-service portals, mobile applications, and social collaboration tools.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Topics** | **Page No.** |
| **1** | **Chapter-1: Introduction**   * Project Planning * Existing System * Proposed System * Aim * Objectives * Benefits * Future Scope | **9** |
| **2** | **Chapter-2: Project Planning**   * Project Schedule * Work Breakdown Structure * Gantt Chart and Timeline Chart * Hardware & Software Resources Requirements | **18** |
| **3** | **Chapter-3: Analysis**   * Business Needs * Functional & Non-functional Requirements * Constraints * DFD & Data Dictionary * Use case analysis * ER-Models | **21** |
| **4** | **Chapter-4: System Design**   * System Architecture * Physical Design using structured design or Object-oriented design (class and object diagram) * Database Schema & Relational design * User Interface Design | **33** |
| **5** | **Chapter-5: System Implementation**   * Details of modules and their abstraction * Various files and their path interconnection * Sources Codes * Snapshots in hierarchical order | **40** |
| **6** | **Chapter-6: Testing**   * Design of tests cases * Test reports for unit testing (Integration testing, System testing, Acceptance testing) * Test reports for specific testing (Load testing, Stress testing, Performance testing, Reliability testing, Recovery testing) * Object Oriented testing for object-oriented analysis design (Testing for inheritance, Polymorphism, Encapsulation) | **43** |
| **7** | **Chapter-7: Conclusion**   * Observation * Project Planning & Analysis * Performance * Limitations | **48** |
| **8** | **Chapter-8: Future Scope**   * References * Appendices | **49** |

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Topics** | **Page No.** |
| **2** | **Chapter-2**  **Gantt Chart** | **9** |
| **4** | **Chapter-4**   * **Database Schema and Relational Design: -**  1. **Employee Table** 2. **Department Table** 3. **Leave Table** | **37**  **38**  **38** |
| **6** | **Chapter-6**  **Testing Table** | **48** |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Caption of Figures** | **Page No.** |
| **1** | **Chater-1**  Database information systems - principle scheme | **12** |
| **3** | **Chapter-3**   1. DFD level 0 2. DFD level 1 3. DFD level 2 4. ER Model | **26**  **27**  **28**  **31-32** |
| **4** | **Chapter-4**   1. Class diagram 2. Object diagram | **36**  **37** |
| **5** | **Chapter-5**   1. Login frame 2. Main Dashboard 3. Add employee 4. Remove employee 5. View and Update employee | **39**  **40**  **40**  **41**  **41** |
|  | **Appendix:**   1. **MS-Access (2000)** 2. **Visual Studio – Microsoft** 3. **N et Framework-basic structure (Lecturers in C# - [4]:** 4. **Database relationships** 5. **Employees\_Details data table** 6. **Working\_History data table** 7. **Contact\_Person\_Detail data table** 8. **Time\_keeping\_History data table** 9. **Salary\_Information data table** | **49**  **50**  **50**  **51**  **51**  **52**  **52**  **53**  **53** |

**List of Figures**

**Chapter-1**

**INTRODUCTION**

In this era of constantly advancing technology, everything has been digitalized. The wide range of employment opportunities has increased the size of the human workforce. As a result, a system that can manage the data of a company's enormous population is needed. This project streamlines the record-keeping procedure thanks to its user-friendly design. The "EMPLOYEE MANAGEMENT SYSTEM" was developed in order to fix the problems with the earlier manual system. The faults with the current system are intended to be eliminated, and in some cases, reduced, via this programme.

The programme is maintained as straightforward as possible to reduce data entry errors. A warning of error is also displayed when wrong data is entered.

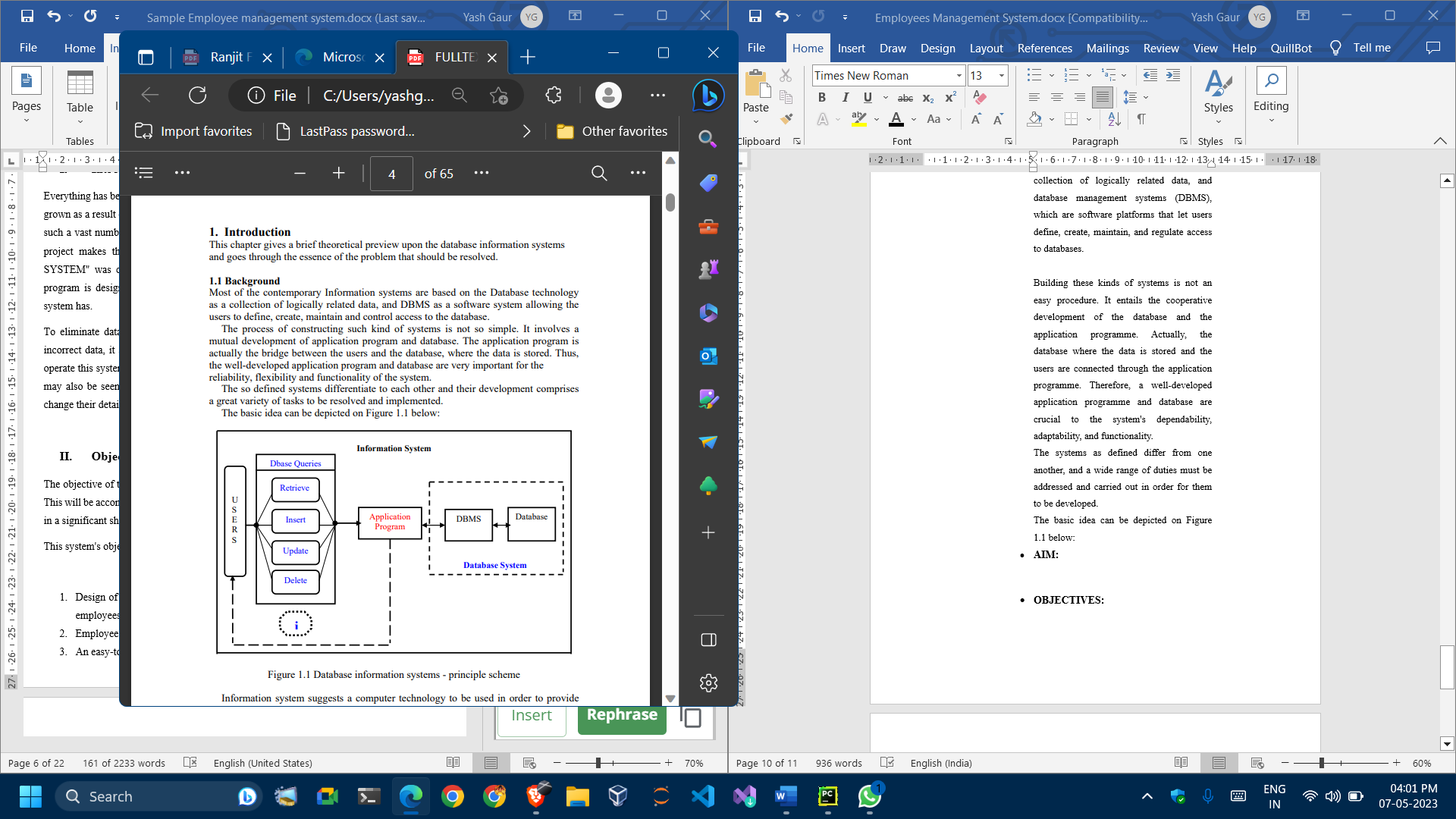
* **PROJECT BACKGROUND:**

The majority of modern information systems are built on database technology, which is a collection of logically related data, and database management systems (DBMS), which are software platforms that let users define, create, maintain, and regulate access to databases.

Building these kinds of systems is not an easy procedure. It entails the cooperative development of the database and the application programme. Actually, the database where the data is stored and the users are connected through the application programme. Therefore, a well-developed application programme and database are crucial to the system's dependability, adaptability, and functionality.

The systems as defined differ from one another, and a wide range of duties must be addressed and carried out in order for them to be developed.

The basic idea can be depicted on Figure 1.1 below:



**Figure 1.1 Database information systems - principle scheme**

Information system denotes the application of computer technology to the provision of information to users inside an organisation, such as for the conversion of data into valuable information; Hardware and software for computers are created.

used, [2].

The creation of the human resources information system is one example. These kinds of systems are in charge of keeping track of employee information within a business and producing reports upon request.

Application Program D Base Queries

databases (DBMS)

Retrieve

Update / Insert / Delete

U U S E R S I - 5 -

Such a system could be integrated with other modules or information systems:

Accounting Information Systems (AIS) are created to turn financial data into information, whereas Management Information Systems (MIS) give managers information that will help them make decisions, and so on.

* **EXISTING SYSTEM:**

The current Employee Management System (EMS) was created in Java using the My SQL workbench with IntelliJ Idea Community version 2023 as the integrated development environment (IDE). The system utilises a Ryzen 3 64-bit processor and the Windows operating system. While IntelliJ Idea offers a complete set of tools and features for creating, testing, and deploying Java applications, the Java programming language offers a flexible and stable framework for creating strong and scalable software applications. The database for the EMS is managed by the My SQL workbench, ensuring that employee data is safe and simple to retrieve. The architecture of the system is created to be highly scalable and capable of handling massive volumes of employee data and workflows. In general, the current EMS system is a strong.

* **PROPOSED SYSTEM:**

Below are the important concepts on which the work has been done and with the support of these I was able to work on my project.

**NET BEANS:** -

A Java-based integrated development environment is called NetBeans (IDE). With the use of a collection of modular software components known as modules, NetBeans makes it possible to create applications. Windows, Mac OS X, Linux, and Solaris can all run NetBeans. It also enables the extension of other programming languages. Third-party developers can add new features to NetBeans-based products, including the NetBeans IDE, in addition to Java programming.

**JAVA:** -

Java is a high-level, object-oriented programming language that aids developers in effectively running their programs. When we discuss android applications, JAVA is the programming language that comes to mind. A programmer can simply create any form of Android application utilizing the JAVA programming language. Numerous libraries are also offered by Java, which is helpful in creating effective Android applications. A Java GUI widget toolkit is called Swing. It is a component of Oracle's Java Foundation Classes (JFC), which gives developers of Java programs access to an API for building graphical user interfaces.

**SWING: -**

Java GUI widget toolkit called Swing. It is a component of Oracle's Java Foundation Classes (JFC), which gives developers of Java programs access to an API for building graphical user interfaces. To improve upon the preceding Abstract Window Toolkit's collection of graphical user interface elements, Swing was developed (AWT). Swing provides a pluggable look and feel that enables applications to have a look and feel that is independent of the underlying platform and a look and feel that mimics the look and feel of various platforms.

**SQL:** -

The computer language SQL, also known as structured query language, is used to handle data in relational database management systems (RDBMS) and relational data stream management systems (RDSMS). Dealing with structured data, or data that has connections between entities and variables, is highly advantageous.

* **AIM:**

An Employee Management System (EMS) is designed to manage employee data and workflows in a thorough and effective manner. The system is made to simplify the numerous HR management procedures, including managing employee data, monitoring time and attendance, processing payroll, administering benefits, and managing performance. The ultimate objective is to increase accuracy, efficiency, and data security while reducing the administrative burden associated with handling employee data.

By acting as a central centre for employee data and workflows, an EMS also attempts to enhance communication and collaboration between staff and management. This may promote a more open and cooperative workplace where staff members feel appreciated and engaged. An EMS's ultimate goal is to increase HR management's overall efficiency and effectiveness, while also increasing the employee experience and promoting corporate expansion.

* **OBJECTIVES:**

Giving a comprehensive approach to personnel information management is the goal of this effort. The way employee data is managed will significantly change as a result of the creation and implementation of an HR management system. Among the goals of this system are the following:

1. Creating a human resources management system that can be used to add and remove employees, view and print employee information, and update employee information.

2. A well-designed database houses employee data.

3. A user-friendly user interface that enables system interaction.

* **BENEFITS:s**

An Employee Management System (EMS) has numerous advantages, such as:

1. **Enhanced efficiency:**

Data input, record-keeping, and reporting are just a few of the manual HR administration duties that an EMS automates. This lessens the workload for HR personnel and increases effectiveness.

1. **Enhanced accuracy:**

An EMS lowers the possibility of data entry and record-keeping errors, which can result in expensive errors and compliance problems. Employee data is kept current and correct thanks to the system.

1. **Savings on costs:**

An EMS lessens the need for human labour and paper-based systems, which can eventually result in cost savings. Additionally, the method may assist in lowering the possibility of non-compliance fees and other monetary penalties.

1. **Improved decision-making:**

An EMS gives HR managers access to real-time data and analytics that can be utilised to make decisions regarding employee development, talent management, and other strategic initiatives.

1. **Collaboration and Communication:**

Collaboration and communication between employees and management can be improved thanks to an EMS, which acts as a central hub for employee data and procedures. This may encourage a more open and cooperative work environment.

1. **Enhanced employee experience:**

Self-service portals, mobile apps, and other tools that facilitate employees' ability to manage their own data and workflows can all be made available by an EMS in order to enhance the experience of its users.

Overall, an EMS can help to increase the efficacy, efficiency, and accuracy of HR administration while also boosting the working environment and promoting corporate expansion.

**Chapter-2**

**PROJECT PLANNING**

Any project must have careful planning in order to be finished on schedule, within budget, and according to the specified requirements. Effective project planning involves taking into account a number of variables, such as the project schedule, work breakdown structure, Gantt chart, timeline chart, and needed hardware and software resources.

* **Project Schedule: -**

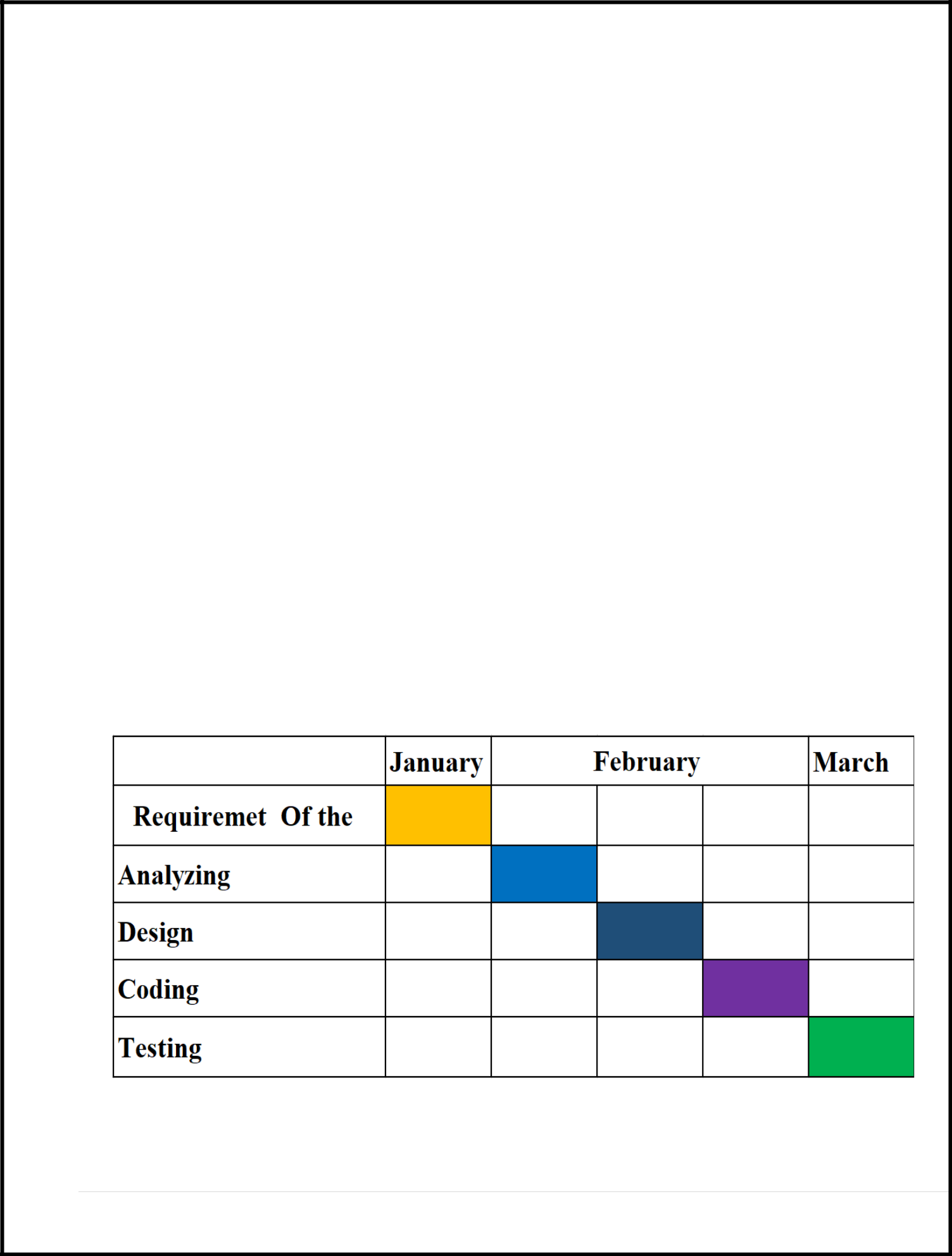
Making a project schedule is one of the initial tasks in project planning. The main project deliverables and milestones should be listed in this timeline, along with an estimate of how long it will take to complete each activity. The calendar should be thorough for a one-month project and include a summary of all activities and subtasks, along withclear deadlines and dependencies.

* **Work Breakdown Structure: -**

Another crucial element of project planning is the work breakdown structure (WBS). A hierarchical chart called the WBS divides the project into smaller, easier-to-manage parts. This makes it easier to make sure that every task is present and that they are all appropriately prioritised and completed in the right order.

* **Gantt Chart: -**

To visualise the project schedule and job breakdown structure, utilise the Gantt chart and timeline chart. The Gantt chart displays the project's chronology, with each task represented by a bar that spans the task's duration. Each task's length and interdependencies are displayed on the timeline chart.

* **Employee Management System Timeline Gantt Chart**

**Fig 2.1 Gantt Chart**

* **Hardware and software resources requirements: -**

The needs for hardware and software resources are also significant factors in project planning. It is crucial to make sure that the project team has access to the availability of all required hardware and software resources. This can entail having access to a certain model of computer or software programme as well as a particular set of data or information sources.

Overall, a good project relies heavily on its preparation. Project managers may make sure the project is finished on time, within budget, and to the required standards by developing a detailed project schedule, work breakdown structure, Gantt chart, timeline chart, and needs for hardware and software resources.

**Chapter-3**

**Analysis**

* **Business Needs: -**

A software programme called an employee management system (EMS) is made to manage employee data and automate HR procedures. Depending on the size and structure of the organisation, an EMS may have different business demands, but some standard needs include:

**Streamlining HR Processes: -**

An EMS can automate a number of common HR operations, including hiring new employees, monitoring time and attendance, managing performance, and administering benefits. This can shorten processing times and cut down on errors, allowing HR professionals to concentrate on more essential tasks.

**Compliance Management: -**

Businesses must abide by a number of labour laws, tax laws, and data privacy laws. By precisely collecting and timely reporting employee data, an EMS can aid in ensuring compliance.

**Talent management: -**

By monitoring employee performance, training and development, and career progression, an EMS can assist firms in identifying and developing their top talent.

**Cost savings: -**

An EMS can assist businesses in lowering the price of HR procedures like hiring, training, and benefits administration. Organizations are able to save time and money by automating these operations.

**Employee Engagement: -**

By giving employees access to their HR information and allowing them to take part in HR procedures like performance reviews and goal-setting, an EMS can help firms increase employee engagement.

An employee management system can help firms increase productivity, cut expenses, and manage their personnel more effectively, all of which will ultimately result in greater business performance.

* **Functional & Non-functional Requirements: -**

**Functional Requirements: -**

**Employee Data Management: -** The EMS must permit HR staff to produce and maintain employee records, which may include personal information, information about the employee's position, and performance information.

**Time and Attendance Tracking: -** The EMS must allow staff members to clock in and out of work, keep track of their sick and vacation days, and manage their schedules.

**Performance Management: -**Performance management: The EMS must permit HR employees to establish performance objectives, carry out performance evaluations, and monitor goal progress.

**Payroll Processing: -** To process employee pay and benefits, the EMS must link with the company's payroll system.

**Training and Development: -**

The EMS must permit HR professionals to oversee employee development plans and monitor success in relation to objectives.

**Non-Functional Requirements: -**

**Performance: -**

**Performance: -**

To guarantee that HR tasks are processed in a timely manner, the EMS must be able to manage a sizable volume of data and transactions swiftly and efficiently**.**

**Reliability: -**

The EMS must always be accessible and running in order for employees and HR personnel to use it as needed.

**Security: -**

The EMS must adhere to data privacy laws and safeguard employee data and HR operations against unwanted access or exposure.

**Usability: -**

The EMS needs to be simple to operate, with simple interfaces and instructions.

**Scalability: -**

The EMS must have the capacity to expand, adapt to shifting business needs, and interface with other systems as needed.

A crucial part in the software development process is analysing the functional and non-functional requirements of an EMS. This analysis offers a road map for designing, building, and testing the system and ensures that it satisfies the needs of the business and its users.

* **Constraints: -**

The design or execution of an employee management system is limited by constraints (EMS). Here are a few examples of restrictions that could be placed on an EMS:

1. **Budget: -**

For some organisations, especially smaller businesses with fewer resources, the expense of building, deploying, and maintaining an EMS may be a barrier.

1. **Technical requirements: -**

For an EMS to function properly, specialised hardware or software may be needed, such as a particular operating system or database. Having insufficient technical capabilities could be a limitation for an organisation.

1. **Data Privacy Laws: -**

When processing employee data, organisations are required to abide by data privacy laws. The kind of data that can be gathered, kept, and exchanged through an EMS may thus be constrained as a result.

1. **Organizational Culture: -**

The culture of the organisation may have an impact on the adoption of an EMS, particularly if there is a preference for manual processes or resistance to change.

1. **Legal Requirements: -**

The EMS must abide by all applicable laws, including those pertaining to labour, taxes, and discrimination, which may place limitations on how employee data is handled and processed.

1. **User Requirements: -**

The EMS must satisfy the requirements of its users, including HR personnel and employees, which may place restrictions on the design and performance of the system.

1. **Integration with Existing Systems: -**

An EMS might have to integrate with already-existing systems, like accounting or payroll software, which could place restrictions on the system's functionality and design.

1. **Time Restrictions: -**

Organizations may only have a short window of time to put an EMS into place, especially if there are pressing business needs or outside pressures to do so.

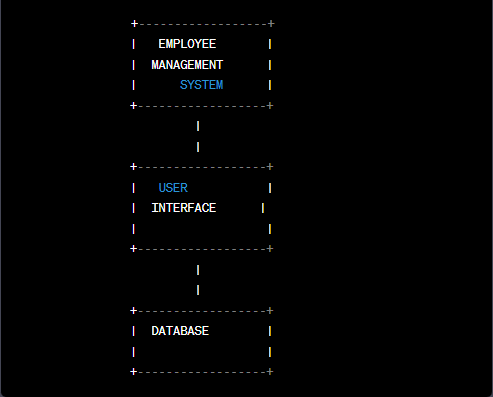
When creating and putting into place an EMS, constraints must be properly taken into account. Constraints may cause a system to be inefficient or expensive to operate if they are not addressed.

* **DFD and data dictionary: -**

An illustration of a DFD and data dictionary for a personnel management system is provided below:

* **Diagram of data flow: -**

**DFD Level 0:**



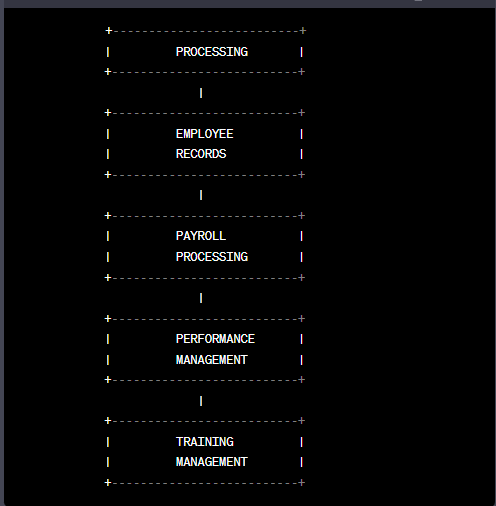
**Figure 3.1 DFD level 0**

**DFD Level 1:**



**Figure 3.2 DFD level 1**

**DFD Level 2:**



**Figure 3.3 DFD level 2**

* **Data Dictionary:**

1. **Employee Records:**
2. Employee ID:
3. Unique identifier for each employee.
4. Name: Employee's full name.
5. Address: Employee's home address.
6. Phone: Employee's phone number.
7. Email: Employee's email address.
8. Position: Employee's job title or position.
9. Hire Date: Date on which the employee was hired.
10. Salary: Employee's annual salary.
11. **Payroll Processing:**

Employee ID: Unique identifier for each employee

Pay Period: Date range for which payroll is being processed

Gross Pay: Employee's total earnings for the pay period

Deductions: Total amount deducted from employee's pay for taxes, benefits, etc.

Net Pay: Employee's take-home pay for the pay period.

1. **Performance Management:**

Employee ID: Unique identifier for each employee.

Performance Goals: Goals set for the employee's performance.

Performance Reviews: Records of the employee's performance reviews and ratings.

Performance Improvement Plans: Plans developed to help employees improve their performance

1. **Training Management:**

Employee ID: Unique identifier for each employee

Training Needs Analysis: Assessment of the employee's training needs

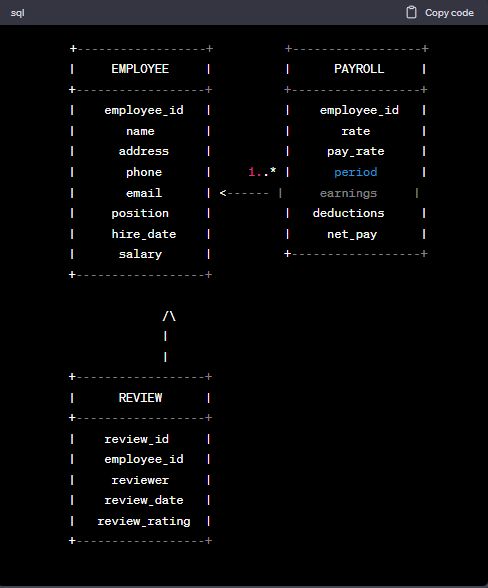
Training Plan: Plan developed to address the employee's training needs

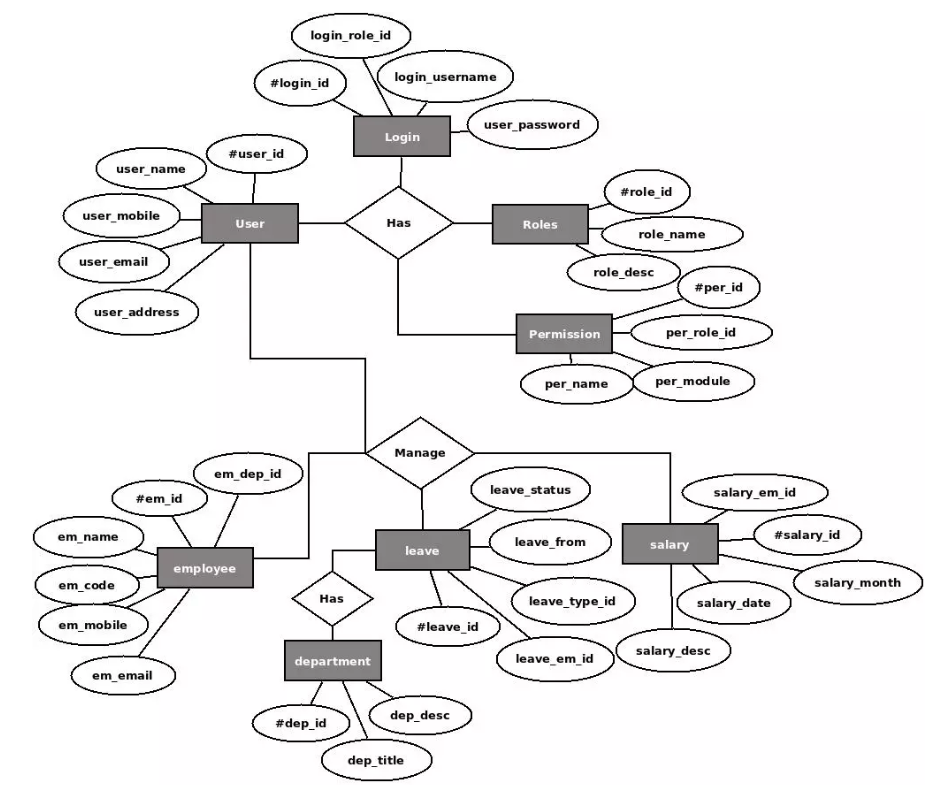
Training Records: Records of the employee's training activities and progress.

The DFD and data dictionary for a personnel management system shown here are merely examples. Depending on the organization's unique needs and limits, the design and implementation may differ.

* **ER Models**

A system's data entities and relationships are depicted graphically in an entity-relationship (ER) model. An illustration of an ER model for a personnel management system is given below:





**ER Model of Employee Management System**

**Chapter-4**

**System Design**

* **System Architecture: -**

Here is a general description of the system architecture of an employee management system, which may change depending on the particular requirements and limitations of the organisation:

**User Interface: -**

This layer offers the system's user interface, allowing users to interact with it via a command-line or graphical user interface (GUI) (CLI).

**Application Layer: -**

Business logic and application functionality are contained in the application layer of the system. It manages duties like managing personnel data, processing payroll, analysing performance, providing training and development, and other HR-related duties.

**Database Layer: -**

This layer houses the system's data. It consists of information about the employees, payroll, performance reviews, training and development, and other pertinent information.

**Security Layer: -**

By limiting access to the system and its data, the security layer ensures the system's security. To guarantee that only authorised users have access to the system and its data, it includes user authentication, authorization, and access control procedures.

**Integration Layer: -**

This layer connects the enterprise resource planning (ERP) system, accounting system, and other human resources (HR)-related systems to the employee management system.

Overall, an employee management system's system architecture is created to support the organization's numerous HR-related duties and operations while preserving the security and integrity of the system and its data.

* **Physical design using structured design or object-oriented design: -**

1. **Employee Class:**

In the system, this class stands in for an employee and includes the following characteristics:

1. **employee\_id:** unique identifier for the employee
2. **name:** name of the employee
3. **address:** address of the employee
4. **phone:** phone number of the employee
5. **email:** email address of the employee
6. **position:** job position of the employee
7. **hire\_date:** date when the employee was hired
8. **salary:** salary of the employee.
9. **Payroll Class:**

This class represents the payroll data for an employee and contains the following attributes:

1. **payroll\_id:** unique identifier for the payroll data..
2. **employee\_id:** unique identifier for the employee being paid
3. **rate:** hourly or monthly rate of pay for the employee.
4. **period:** pay period (weekly, bi-weekly, monthly)..
5. **earnings:** total earnings for the pay period
6. **deductions:** total deductions for the pay period.
7. **net\_pay:** net pay for the pay period.
8. **Review Class:**

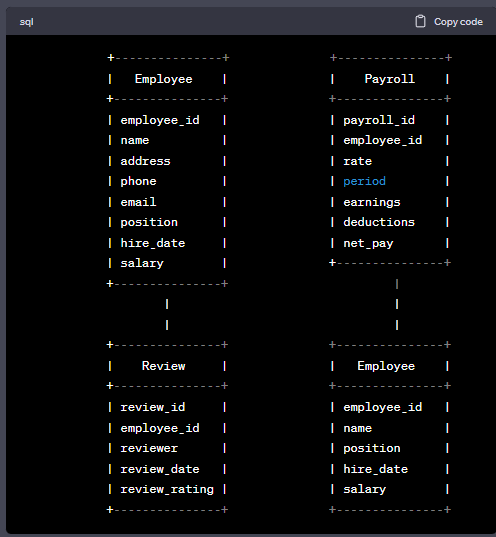
This class represents the performance review data for an employee and contains the following attributes:

1. **review\_id:** unique identifier for the review
2. **employee\_id:** unique identifier for the employee being reviewed
3. **reviewer:** name of the reviewer
4. **review\_date:** date when the review was conducted
5. **review\_rating:** rating given to the employee in the review
6. **EmployeeManagementSystem Class:**

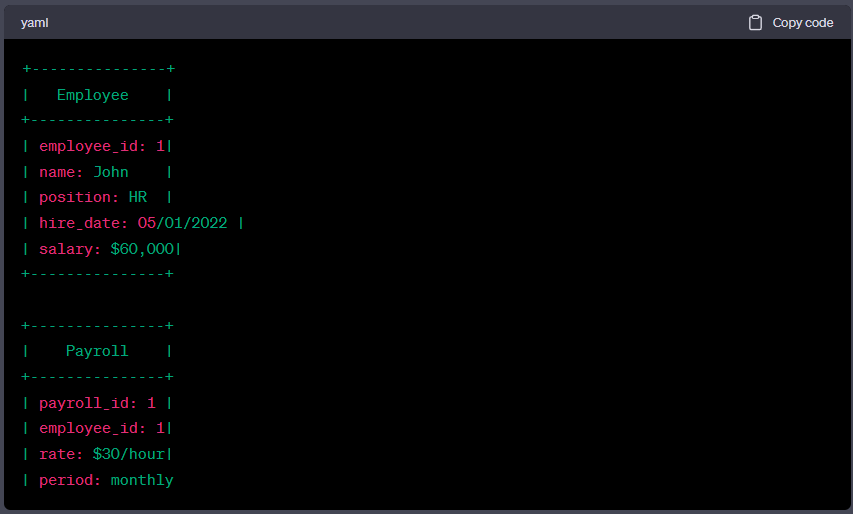
This class represents the main class of the system and contains the following methods:

1. **addEmployee():** method for adding a new employee to the system
2. **removeEmployee():** method for removing an employee from the system
3. **updateEmployee():** method for updating the information of an existing employee in the system
4. **addPayroll():** method for adding payroll data for an employee
5. **updatePayroll():** method for updating payroll data for an employee
6. **addReview():** method for adding a performance review for an employee
7. **updateReview():** method for updating a performance review for an employee
8. **generatePayrollReport():** method for generating a payroll report for a given pay period
9. **generatePerformanceReport():** method for generating a performance report for a given period.

* **Class Diagram:**



* **Object Diagram:**



* **Database Schema and Relational Design: -**

1. **Employee Table: -**



1. **Department Table: -**



1. **Leave Table: -**



* **The relationship between the tables is:**

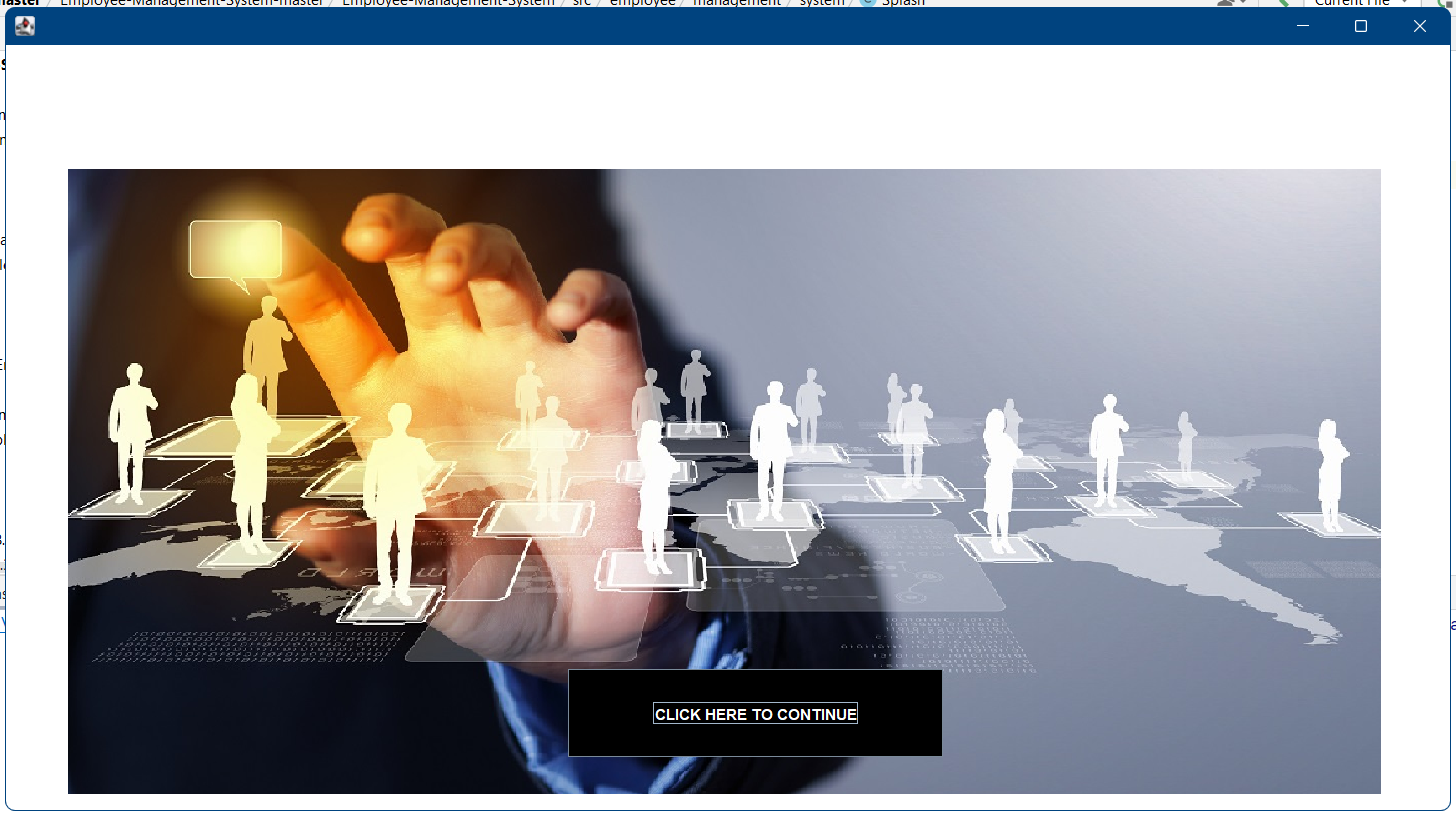
1. One department can have many employees (one-to-many relationship).
2. One employee can have many attendance records (one-to-many relationship).
3. One employee can have many leave records (one-to-many relationship).
4. One employee can have one manager (many-to-one relationship).

**Chapter-5**

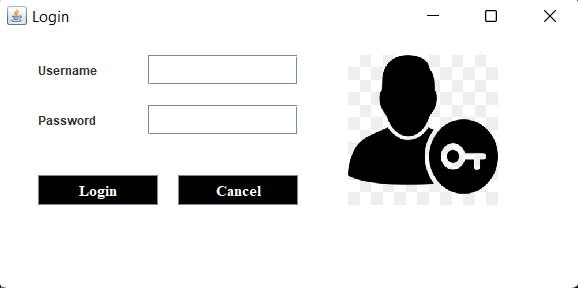
**System Implementation**

* **Implementation and Result: -**

Following are the screens of the Employee Management System where you can see all the features of this system in use and you can also see the GUI of the system.



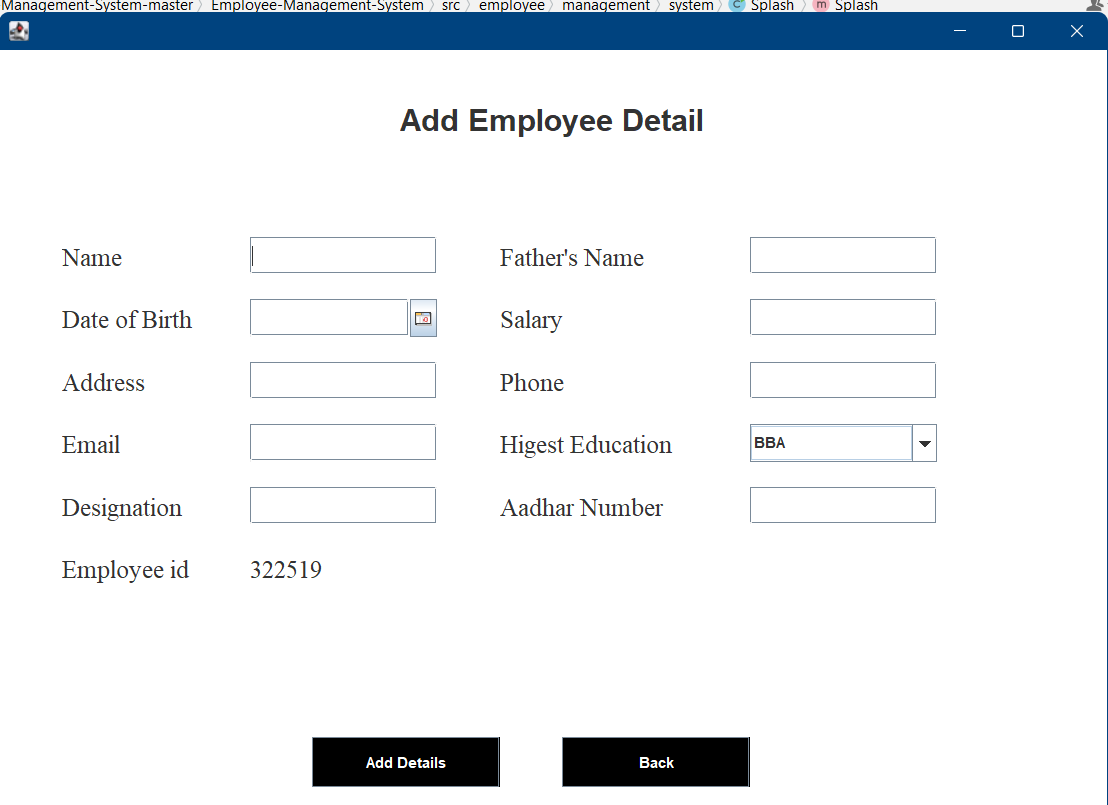
1. **Login frame –** This is the login frame of this system where user have to enter the required credentials to have access for the main dashboard.

****

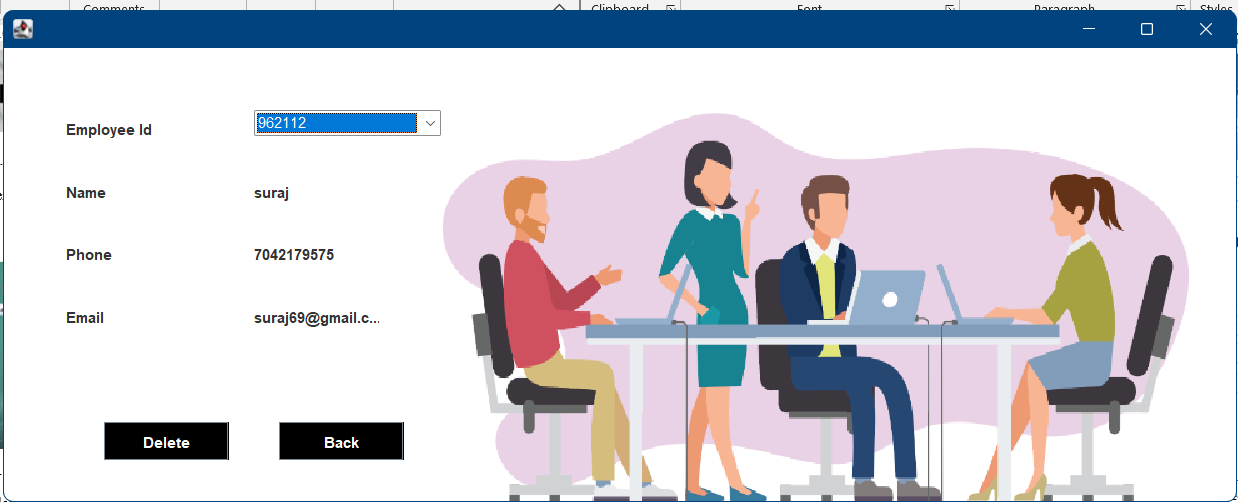
1. **Main Dashboard –** After login in, user is directed to the main dashboard of this system where user can perform various operations like adding an employee, deleting an employee.

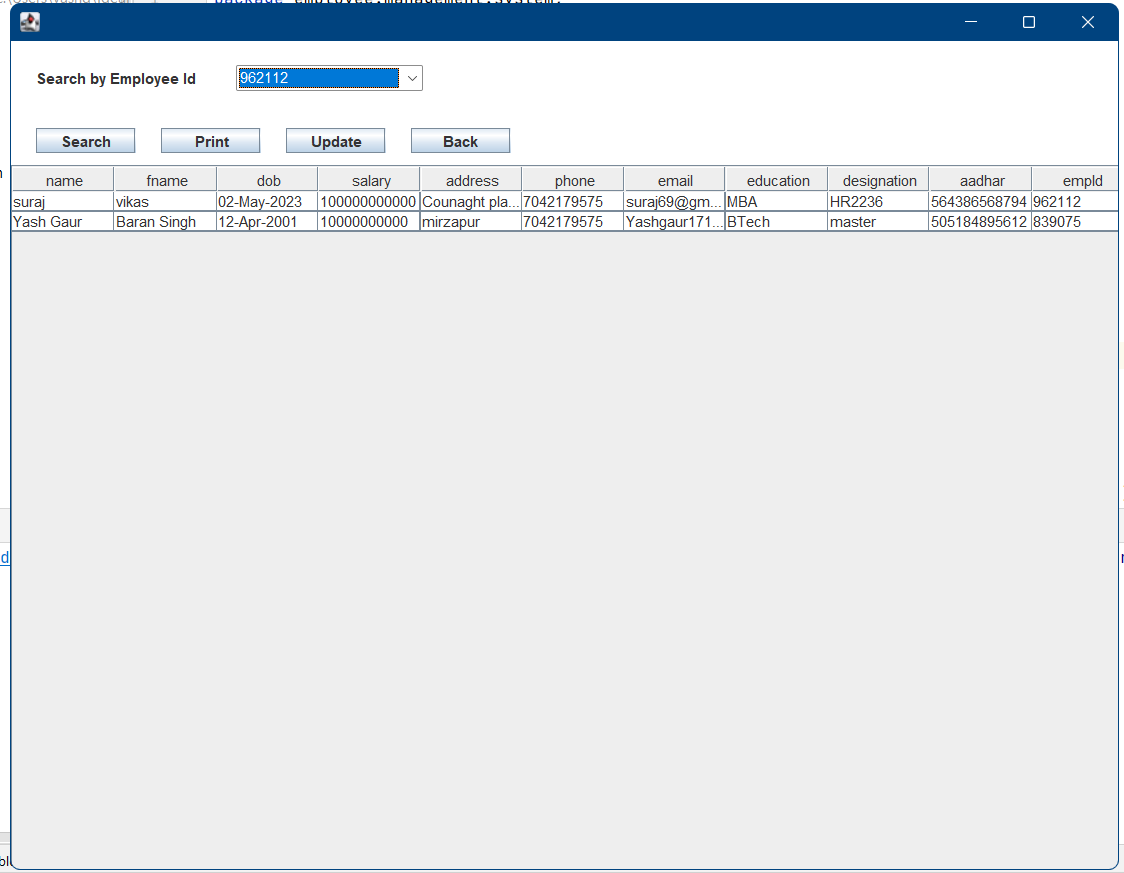


1. **Add employee –** Here user have to enter all the required credentials to add a new employee to the system.



1. **Remove employee –** User has to enter the employee id in order to delete his information from the system.



1. **View and update employee –**  In order to view and update employee information, the user have to enter employee ID. 

**Chapter-6**

**Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST ID** | **TEST PURPOSE** | **TEST**  **CONDITION** | **EXPECTED**  **OUTPUT** | **OUTPUT** | **REMARK** |
| **TC1** | Check username & password | If user details are not correct, display error message. | Grant access to main dashboard. | Access granted to main dashboard | Test successful |
| **TC2** | To add new user to the system | If user already exists, error message should be displayed. | New user should be added. | New user added successfully | Test successful |
| **TC3** | To view existing employee  Information | If employee exists, then information should be displayed, else error message should be displayed. | Employee information should be displayed. | Employee information displayed. | Test successful |
| **TC4** | To remove an employee | If employee exists, then employee should be removed else error message should be displayed. | Employee should be removed. | Employee removed successfully. | Test successful |
| **TC5** | Update employee information | If employee exists, then information should be updated. | Employee information should be updated. | Employee information updated successfully. | Test successful |

* **System Testing: -**

The process of confirming and validating the system's operation and behaviour in relation to the requirements and specifications is known as system testing in the context of the employee management system. This entails testing the system as a whole to make sure that it satisfies the organisational needs in terms of functionality and business needs.

System testing's objective is to find and address any flaws or problems that might prevent the system from performing as intended or from satisfying the needs of its users. All facets of the system, including the user interface, data entry and retrieval, business logic, security, and performance, should be tested.

In an employee management system, some crucial system testing should be carried out, such as:

1. **Unit Testing: -**

Unit testing performed on each module or block of code during development. Unit testing is normally done by the programmer who writes the code.

1. **Integration Testing: -**

Testing that involves integration ensures that the personnel management system can, if necessary, interact with other systems like payroll or timekeeping ones.

1. **Acceptance Testing: -**

Acceptance testing entails testing the system with end users to make sure it satisfies their requirements and expectations.

1. **Load Testing: -**

Testing the system's performance and behaviour under both average and peak loads is known as load testing. This would entail assessing the system's capacity to manage numerous concurrent users performing different tasks, such as adding and updating employee records or producing reports, in the case of an employee management system. Load testing helps to locate system bottlenecks and makes sure the system is capable of handling the anticipated load without stuttering or crashing.

1. **Stress Testing: -**

Stress testing: Stress testing entails putting a system's performance and behaviour to the test in circumstances that are beyond its range of regular operation. This would entail assessing the system's capacity to handle unexpected and excessive loads, such as an abnormally high volume of employee data or requests, in the case of an employee management system. Stress testing ensures that the system can recover gracefully from failures and helps to discover its breaking point.

1. **Performance Testing: -**

This kind of testing determines whether the system is capable of handling the anticipated load and operating effectively under various circumstances, such as adding a significant number of employee records or producing intricate reports.

1. **Reliability Testing: -**

Testing for reliability involves determining if a system can function reliably and consistently over time. In the case of an employee management system, this would entail evaluating the system's capacity to run properly and error-free for a protracted length of time. Reliability testing aids in finding and resolving any problems that might eventually result in the system failing or malfunctioning.

1. **Recovery Testing: -**

Recovery testing is done to demonstrate a software salutation is reliable, trustworthy and can successfully recoup form possible crashes.

1. **Security Testing: -**

Security Testing is a variant of Software Testing which ensures, that system and applications in an organization, are free from any loopholes that may cause a big loss. Security testing of any system is about finding all possible loopholes and weaknesses of the system which might result into a loss of information at the hands of the employees or outsiders of the Organization.

**Chapter-7**

**Conclusion**

The initiative's objective is to give administrators access to computers and digitise employee databases in organisations. Software is used as an information system by employees and administrators. The user can utilise this location to keep their database indefinitely in a safe and secure manner. Adding.

Using the Employee Management System makes it straightforward and easy to delete, view, and change employee information.

* **Future Scope: -**

The system's GUI and newly additional features are the most fundamental ones. Future improvements to the Graphical User Interface and the addition of new functionality will be made to this system. If the graphical user interface is updated, the system will be easier to use, more features will greatly improve it, and HR will be able to carry out more tasks.

* **Limitations: -**

Java is a well-liked programming language used to create many different kinds of applications, including systems for managing employees. However, Java has its limitations when it comes to creating employee management systems, just like any other computer language. A few of the Java programming language's constraints on staff management systems are as follows:

1. **Performance:**

Although Java is renowned for its speed and performance, it might not always be the best option for applications that need real-time processing and high performance, such as extensive staff management systems. This is because a Java Virtual Machine (JVM), which is required to run Java applications, can affect performance and lengthen response times.

1. **Memory management:**

Java's automatic memory management system, which releases memory when objects are no longer needed, can occasionally cause performance problems and cause the application to lag. This is especially true for memory-intensive apps, including those that work with a lot of employee data.

1. **Complexity:**

The creation and maintenance of Java applications can be challenging, particularly when intricate business processes are involved. The employee management system's development time and cost may increase as a result.

1. **Security:**

Despite the fact that Java is well renowned for its security features like the Security Manager and Access Controller, there may still be security flaws that hackers might take advantage of. These flaws may provide unauthorised access to sensitive employee data, including personal and payroll information.

1. **Deployment:**

Integrating a Java-based staff management system with other systems, such as payroll and time tracking systems, can be a challenging and time-consuming procedure. Delays and higher expenses could come from this.

Despite these drawbacks, Java's scalability, robustness, and cross-platform portability make it a popular choice for creating employee management systems. These restrictions can be lessened with careful planning and design, and the Java programming language can be used to create a powerful staff management system.

**Chapter-8**

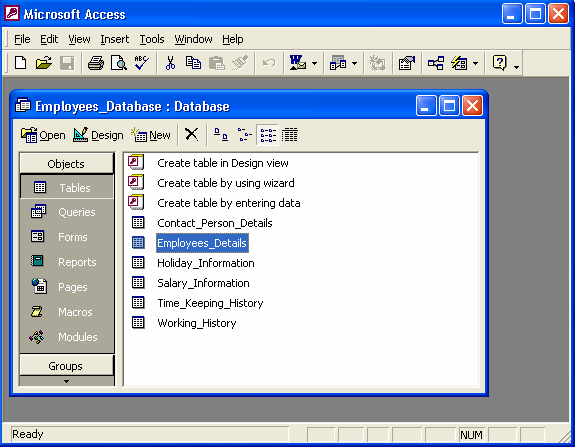
**References: -**

1. Renae Broderick, John W. Boudreau, “Human resource management, information technology, and the competitive edge”, Academy of Management Executive, 1992 Vol. 6 No. 2
2. Julie Bulmash, “Human Resource Management and Technology”, Chapter 3.
3. Ian Sommerville, “Software Engineering”, 9th Edition, Addison-Wesley, 2011.
4. Avison, D. and Fitzgerald, G. (2003). Information systems Development Methodologies, Techniques and Tools.3rd Edition. McGraw-Hill Education Limited Bershire
5. Juan Manuel Munoz Palacio, Information systems development methodologies for Data-driven Decision Support Systems, 2010.
6. Deitel, PJ & Deitel, HM, 2008, Internet & World Wide Web How To Program, Dorling Kindersley, India.

**Appendix**

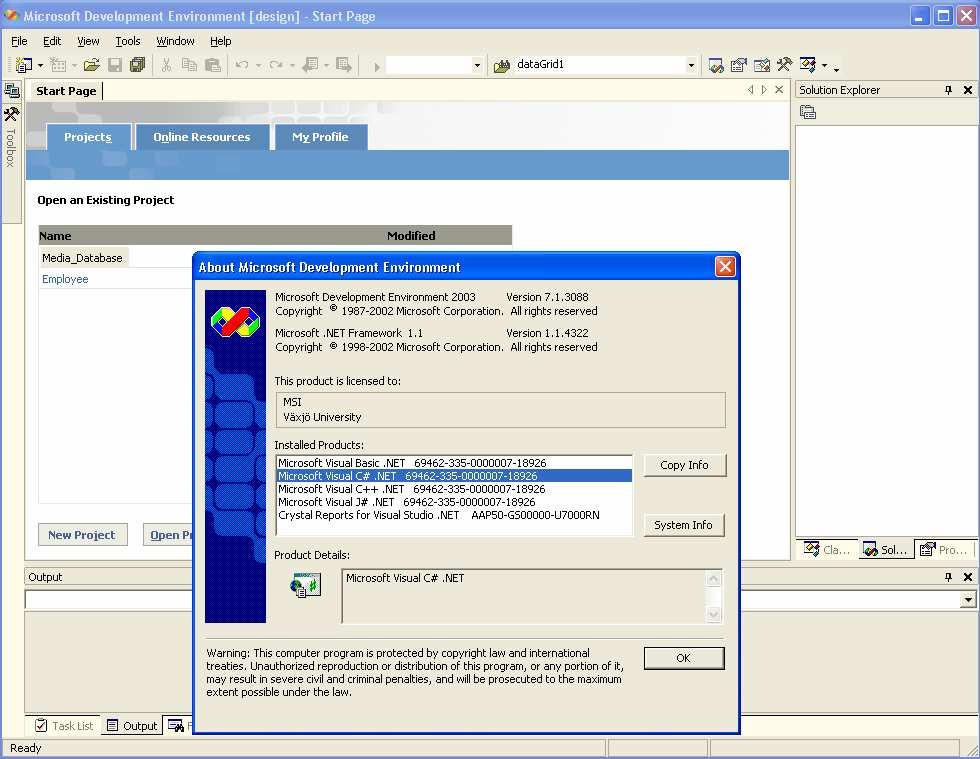
**Appendix A: Programming Environments and Database Content**

This section will give some visual details about the content and the structure of the database that has been designed and constructed for the purposes of the program.

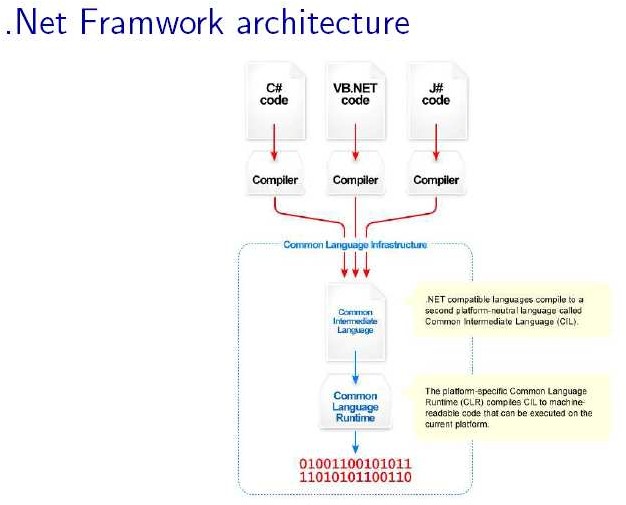
* **MS-Access (2000):**

**Figure A.1**

* **Visual Studio – Microsoft:**

 **Figure A.2**

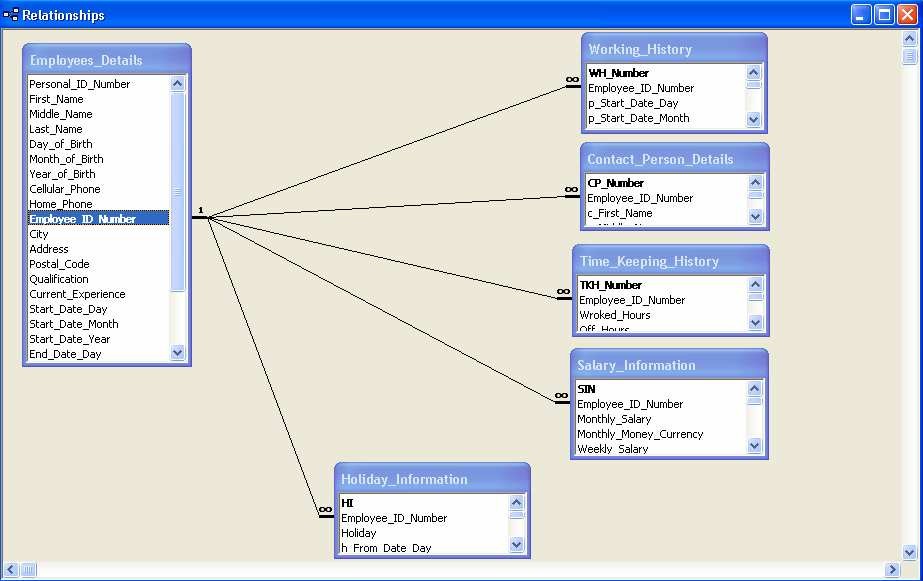
**Net Framework-basic structure (Lecturers in C# - [4]:**

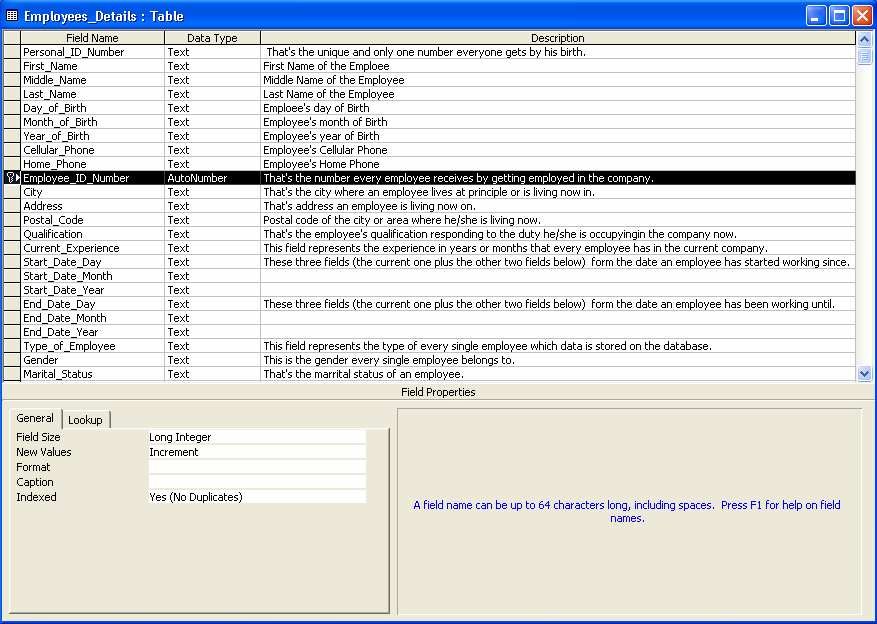


**Figure A.3**

* **Database relationships:**

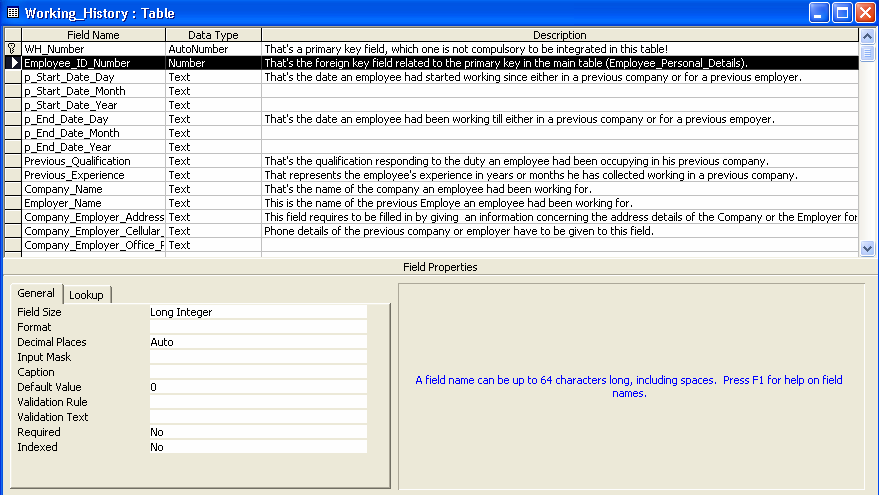
**Figure A.4**

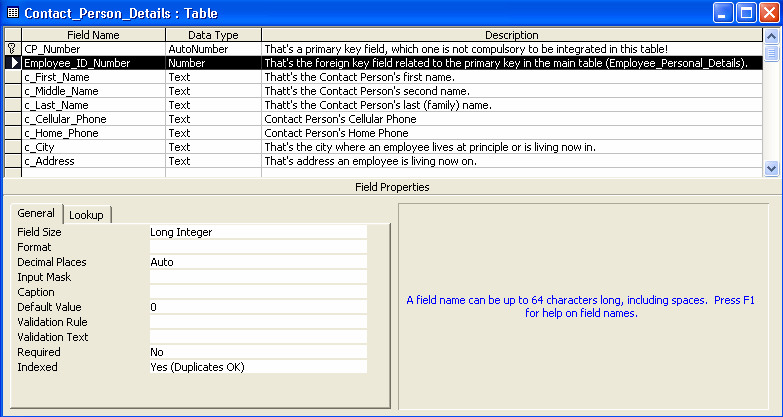


* **Employees\_Details data table:**

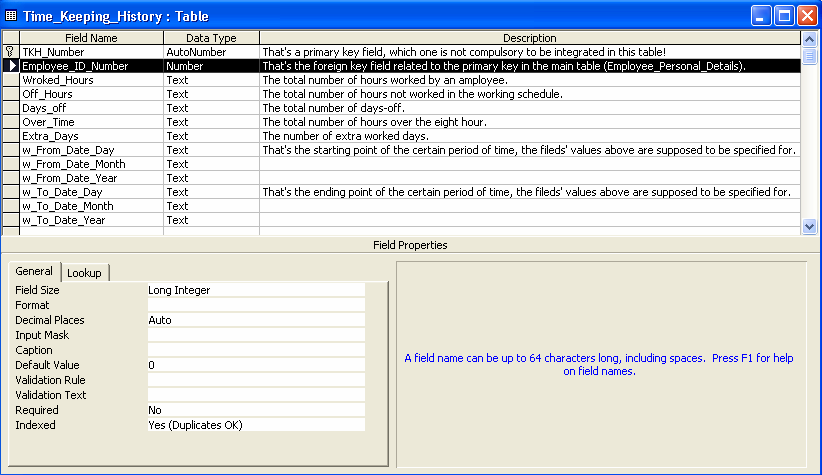
**Figure A.5**

* **Working\_History data table:**

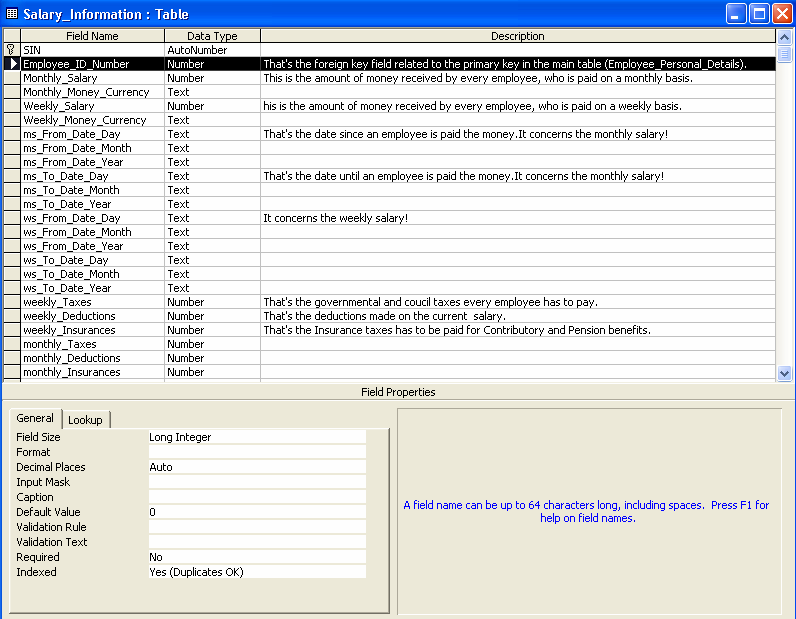
**Figure A.6**

* **Contact\_Person\_Detail data table:**

**Figure A.7**

* **Time\_keeping\_History data table:**

**Figure A.8**

* **Salary\_Information data table:**

**Figure A.9**