EMPLOYEE MANAGEMENT SYSTEM

(EM-System)

# Project Report

**<Version 1.0>**

*Submitted in Partial Fulfillment of the Requirement For the Degree of*

**BACHELOR OF TECHNOLOGY (COMPUTER SCIENCES ENGINEERING)**

**November,2021**



|  |  |
| --- | --- |
| **PROJECT GUIDE:**  **Mr. Shambhu Bharadwaj** | **SUBMITTED BY:**  **Paras Jain (TCA2011006)**  **Shivang Sharma (TCA2011009)** |
|  |  |

**FACULTY OF ENGINEERING & COMPUTING SCIENCES TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD**

**DECLARATION**

We **Paras Jain and Shivang Sharma**, are student of **B.Tech(CSE) Vth Semester** , studying at Faculty of Engineering & Computer Science, Teerthanker Mahaveer University, Moradabad (UP), hereby declare that the Training Report on “**EMPLOYEE MANAGEMENT SYSTEM”** submitted in partial fulfillment of Bachelor of Technology(Computer Science Engineering) , is the original work conducted by us.

The information and data given in the report is authentic to the best of our knowledge.

This Training Report is not being submitted to any other University for award of any other Degree, Diploma and Fellowship.

## Date: 01/01/2022

**Paras Jain (TCA2011006)**

**Shivang Sharma (TCA2011009)**

**CERTIFICATE**

This is to certify that Project entitled “**EMPLOYEE MANAGEMENT SYSTEM**” has been submitted in partial fulfillment of the requirement for the degree of **Bachelor of Technology(C.S.E),** is carried out by student, name listed below under my supervision and guidance.

## Student Name (with Enrollment No.)

**Paras Jain (TCA2011006)**

**Shivang Sharma (TCA2011009)**

**Project Guide: Mr. Shambhu Bharadwaj**

**Department Name: CCSIT Designation: Research Scholar Date:** **01/01/2022**

**ACKNOWLEDGEMENT**

We are very happy to greatly acknowledge the numerous personalities involved in lending their

help to make our project “**EMPLOYEE MANAGEMENT SYSTEM**” a successful one.

We take this opportunity to express our deep sense of gratitude to our honorable Principal **“Dr. Rakesh Kumar Dwivedi”** for providing excellent academic climate in the college that made this endeavor possible.

We give our whole hearted admiration and deep sense of gratitude to “**Dr. Ashendra Kumar Saxena**”**,** HOD, “**CCSIT**”, TMU for his inspiration, valuable guidance, encouragement, suggestion and overall help throughout.

We express our sincere thanks to our guide **“Mr. Shambhu Bharadwaj”, “DESIGNATION”,** “**CCSIT**”, TMU, for his keen interest and invaluable help throughout the project.

We would like to express our sincere gratitude to our project coordinator **“Rajendra Prasad Pandey”, “Assistant Professor”,** “**CCSIT**”, TMU, for his kind support and encouragement throughout this course of work.

Finally, we express our gratitude to all the Teaching and Non-Teaching staff of “**CCSIT**”, TMU for their

timely support and suggestions.

|  |  |
| --- | --- |
| Date: 01/01/2022 | **Paras Jain (TCA2011006)**  **Shivang Sharma (TCA2011009)** |
|  |  |
|  |  |

**Brief About the Company**

**Internshala** is a dot com business with the heart of dot organization.

**Internshala** is **an internship and online training platform**, based in Gurgaon, India. Founded by **Sarvesh Agrawal**, an IIT Madras alumnus, in 2011, the website helps students find internships with organisations in India.

**Internshala** is on a mission to equip students with relevant skills & practical exposure to help them get the best possible start to their careers. Imagine a world full of freedom and possibilities. A world where you can discover your passion and turn it into your career. A world where you graduate fully assured, confident, and prepared to stake a claim on your place in the world.

How it all started,

**In 2010** - Sarvesh, our founder & CEO, started Internshala as a blog with a mission to bring a culture of meaningful internships in India. And for the first two years, he hired only virtual interns.

**In 2013** - After building a small team, we then launched our website with just one goal - to equip every student in India with their dream internship. And we did it all for free.

**In 2015** - The next big step could not have been anything other than launching our very own Android app, bringing Internshala in the ‘hands’ of the students.

**In 2016** - After many successful years as an internship platform, our motivation to upskill the students only increased, and that’s when we kickstarted a new journey with Internshala Trainings.

**In 2020** - With Fresher jobs, we embarked on a journey filled with newer challenges, which allowed us to provide bigger & better opportunities to graduates with 0-2 years of experience.

**In 2021** - With an insight that more than 90% of the graduates in India start their careers with a job that pays less than 3LPA, we came up with Jobs Oriented Specialization programs to help the students start their careers in their dream profiles.

**ABSTRACT**

The main objective of the EMPLOYEE MANAGEMENT SYSTEM is to manage the details of employee’s according to its salts. It manages all the information about employee’s. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing and choose the correct employee’s data.

Keywords: **JAVA**

**INDEX**

|  |  |  |
| --- | --- | --- |
| S.no | Topic | Page no |
| 1 | **INTRODUCTION** |  |
| 1.1 | Introduction |  |
| 1.2 | Aim |  |
| 1.3 | Feasibility Study |  |
| 1.4 | Organization of Report |  |
| 2 | **SOFTWARE REQUIREMENTS SPECIFICATION** |  |
| 2.1 | Hardware Requirement |  |
| 2.2 | Software Requirement |  |
| 3 | **DESIGN & Methodology** |  |
| 3.1 | Software Development Life Cycle Model |  |
| 3.2 | Methodology |  |
| 4 | **IMPLEMENTATION DETAILS** |  |
| 4.1 | FRONT END |  |
| 4.2 | BACK END |  |
| 5 | **TESTING** |  |
| 5.1 | UNIT TESTING |  |
| 5.2 | INTEGRATION TESTING |  |
| 5.3 | SOFTWARE VERIFICATION AND VALIDATION |  |
| 5.4 | Black-Box Testing |  |
| 5.5 | White-Box Testing |  |
| 5.6 | SYSTEM TESTING |  |
| 6 | **RESULTS** |  |
| 7 | **ADVANTAGES** |  |
| 8 | **CONCLUSION** |  |

# INTRODUCTION

The project EMPLOYEE MANAGEMENT SYSTEM is a software based application that allows the administrator to handle all the personal data of each and every employee’s quickly and safely. Using Interactive TUI anyone can quickly learn to use the complete system. EMPLOYEE MANAGEMENT SYSTEM.

INTRODUCTION OF EMPLOYEE MANAGEMENT SYSTEM: –

Are you looking for Employee Management System for your summer industrial training project? We are here to help you. You can contact us The project Employee Management System is a software based application that allows the administrator to handle all the personal data of each and every employee’s quickly and safely. Using Interactive TUI anyone can quickly learn to use the complete system. Summer Industrial Training Project Help for Computer Science Using this, the administrator doesn’t have to sit and manage the entire activities on paper, and at the same time, the head will feel comfortable to keep check of the whole system. This system will give him flexibility to manage the entire system from a single online portal.

# AIM

Our proposed system is an Employee Management System that enables ease for the customers. It overcomes the disadvantages of the traditional paper system.

The objective of “Employee Management System” is designing a scheduling system for a work center. Employee Management System is a distributed application, developed to maintain the details of employees working in any organization. It maintains the information about the personal details of their employees.

This project consists of Java Program for Employee Management System. In this project I have created a TUI program for Employee Management System which consists of 4 different operation’s-

(1) Adding Employee

(2) Removing Employee

(3) Updating Employee

(4) Viewing details of Employee

This system improves the method of maintaining and managing employee’s personal details such as

1. Name
2. Father’s name
3. Employee's ID
4. Employee's Email ID
5. Employee's Position
6. Enter Employee contact Info
7. Enter Employee's Salary

# FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1. Operational Feasibility
2. Technical Feasibility
3. Economic Feasibility

# OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, predictability, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development require appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

# TECHNICAL FEASIBILITY

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

# ECONOMICAL FEASIBILITY

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

# ORGANISATION OF THE REPORT

* + 1. **INTRODUCTION**

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

# SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

# DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

# IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

# RESULTS AND DISCUSSION

This section has screenshots of all the implementation i.e. user interface and their description.

# 1.4.6 SUMMARY AND CONCLUSION

This section has screenshots of all the implementation i.e. user interface and their description

**CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION**

* 1. **Hardware Requirements**
* **Processor :** Pentium IV or Above
* **RAM :** 1GB or above
* **Hard Disk :** 50GB or above
* **Input Devices :** Keyboard, Mouse
* **Output Devices :** Monitor

# Software Required

* **Operating System :** Linux, Ubuntu, Mac, Windows XP, 7, 8, 8.1, 10, 11
* **Frontend :** IntelliJ IDEA
* **Backend :** JAVA

# CHAPTER 3 : DESIGN & Methodology

In This Section describes the methodology applied during the development of alternative employee’s system. A methodology is a model, which project managers employ for the design, planning, implementation and achievement of their project objectives. Effective project management is essential in absolutely any organization, regardless of the nature of the business and the scale of the organization. From choosing a project to right through to the end, it is important that the project is carefully and closely managed. Based on the nature of my project solution, it was essential to use incremental Software development life

cycle (SDLC). The project typically has a number of Phases and the level of control required over each phase are primarily defined by the nature of the Project, the complexity of the same and the industry to which the Project has to cater to. An Incremental (SDLC) model consists of a number of dependent increments that are completed in a prescribed sequence. Each increment includes a Launching, Monitoring and Controlling, and Closing Process Group for the functions and features in that increment only. Each increment integrates additional parts of the solution until the final increment, where the remaining parts of the solution are integrated.

# Software Development Life Cycle Model

* + 1. **WATERFALL MODEL**

The meaning of Agile is swift or versatile."Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Following are the phases in the Agile model are as follows:

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback



## 3.2.1 Justification for the Methodology

This model can be used when the requirements of the complete system are clearly defined and understood, like the case of this project where;

* Major requirements were evidently defined; however, some details evolved

with time.

* There was a need to complete the project within a fixed time schedule.
* A new technology is being used or the resources with needed skill set are not available.
* The project had some high-risk features and goals.

**Planning the site Design-** The design of the website include:

* + Details about the information architecture.
  + The planned structure of the site.

This is where most of us start to have fun — with the design phase of the project. While you can jump right into your editor now, we recommend you still remain outside of it and do your  [design in a graphics program](https://www.lifewire.com/the-graphic-design-process-1697522) or even on paper first.

You will want to think about:

* + Layout of your designs.
  + Color including creating a color scheme for the site and how those colors work together.
  + You should also plan on a theme for your site including decorative images and icons, including a site favicon.
  + Gather or Create the Site Content. Content is what people come to your site for. This can include text, images, and multimedia. By getting at least some of the content ready ahead of time, you can more easily start building the site.

You should look for:

* + **Text**: this can be articles, blog posts, lists, reviews, or anything that you want to write about on your site.
  + **Graphics**: there are lots of places to  [find images for web pages](https://www.lifewire.com/web-resources-for-public-domain-images-3482714) including photos you’ve taken and free images. Be sure you’re using the right format for your images.

After planning and designing the site, it is then given structure by using JAVA

The file is created using a package of java.io (i.e. file) and is linked to our software through FileWriter function.

# 

# Block Diagram/ Machine Diagram

# 

**Working of Project**

* **Add Employee :**

The Users will use their exclusive information to register. After filling the all details, he/she have to submitting it. If the area is not correctly filled the user remains on the same page but if the requirements are met the data goes to the new file which will created by employers ID and saves the information of the employee.

EXIT Portal

Update Employee Details

ADD Employee

Check Employee

Details

Remove Employee

* **Getting information of any employee :**

With the help of this software organisations will see all the details of registered

employee, and if the employee is not registered then it gives a message that is

java.io.FileNotFoundException: file213123.txt (The system cannot find the file specified)

* **Updating or Removing employee :**

If any wrong information is present in the file then we can update it very easily and quickly, and from this section we will remove employee if he/she (the employee) leaves the organisations.

# CHAPTER 4 : IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project.

# : FRONT END

* + 1. **IntelliJ IDEA**

IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software. It is developed by JetBrains (formerly known as IntelliJ), and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. Both can be used for commercial development. The first version of IntelliJ IDEA was released in January 2001, and was one of the first available Java IDEs with advanced code navigation and code refactoring capabilities integrated.

In a 2010 InfoWorld report, IntelliJ received the highest test center score out of the four top Java programming tools: Eclipse, IntelliJ IDEA, NetBeans and JDeveloper.

In December 2014, Google announced version 1.0 of Android Studio, an open-source IDE for Android apps, based on the open source community edition of IntelliJ IDEA. Other development environments based on IntelliJ's framework include AppCode, CLion, DataGrip, GoLand, PhpStorm, PyCharm, Rider, RubyMine, WebStorm, and MPS.

Though designed primarily for Java development, IntelliJ IDEA understands many other programming languages, including Groovy, Kotlin, Scala, JavaScript, TypeScript, and SQL, and it provides smart coding assistance for each of them.

Features of IntelliJ IDEA :

\* Coding assistance :

The IDE provides certain features like code completion by analyzing the context, code navigation which allows jumping to a class or declaration in the code directly, code refactoring, code debugging , linting and options to fix inconsistencies via suggestions.

\* Built in tools and integration :

The IDE provides integration with build/packaging tools like grunt, bower, gradle, and SBT. It supports version control systems like Git, Mercurial, Perforce, and SVN. Databases like Microsoft SQL Server, Oracle, PostgreSQL, SQLite and MySQL can be accessed directly from the IDE in the Ultimate edition, through an embedded version of DataGrip.

\* Plugin ecosystem :

IntelliJ supports plugins through which one can add additional functionality to the IDE. Plugins can be downloaded and installed either from IntelliJ's plugin repository website or through the IDE's inbuilt plugin search and install feature. Each edition has separate plugin repositories, with both the Community and Ultimate editions totaling over 3000 plugins each as of 2019.

# : BACK END

* + 1. **JAVA**

# Java is one of the programming language or technology used for developing web applications. Java language developed at SUN Micro Systems in the year 1995 under the guidance of James.

# 

# Gosling and there team. Originally SUN Micro Systems is one of the Academic university (Standford University Network)

# Whatever the software developed in the year 1990, SUN Micro Systems has released on the name of oak, which is original name of java (scientifically oak is one of the tree name). The OAK has taken 18 months to develop.The oak is unable to fulfill all requirements of the industry. So James Gosling again reviews this oak and released with the name of java in the year 1995. Scientifically java is one of the coffee seed name.

# Java divided into three categories, they are

# • J2SE (Java 2 Standard Edition)

# • J2EE (Java 2 Enterprise Edition)

# • J2ME (Java 2 Micro or Mobile Edition)

# Overview Of Java :

Java is a platform independent, more powerful, secure, high performance, multithreaded programming language. Here we discuss some points related to java.

The Java Runtime Environment (JRE) is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application.

JVM is set of programs developed by sun Micro System and supplied as a part of jdk for reading line by line of byte code and it converts into native understanding form of operating system. Java language is one of the compiled and interpreted programming language.

Garbage Collector is the system Java program which runs in the background along with regular Java program to collect un-Referenced (unused) memory space for improving the performance of our applications.

An API (Application Programming Interface) is a collection of packages, a package is the collection of classes, interfaces and sub-packages. A sub-package is a collection of classes interfaces and sub sub packages etc.

Java programming is containing user friendly syntax so that we can develop effective applications. In other words if any language is providing user friendly syntax, we can develop error free applications.

JIT is the set of programs developed by SUN Micro System and added as a part of JVM, to speed up the interpretation phase.

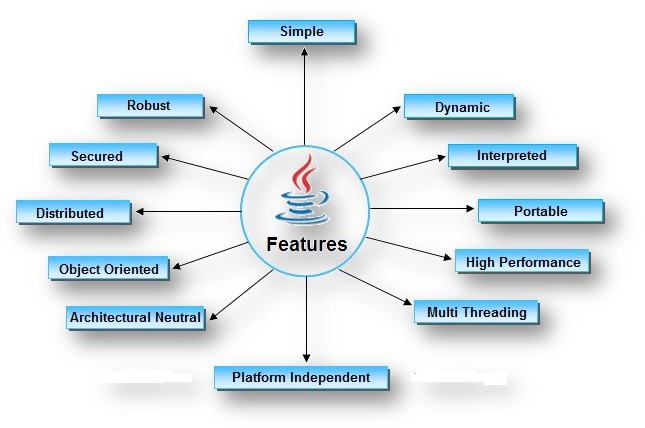
**JVM :** JVM (Java Virtual Machine) is a software. It is a specification that provides runtime environment in which java bytecode can be executed. It not physically exists.JVMs are not same for all hardware and software, for example for window os JVM is different and for Linux VJM is different. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.

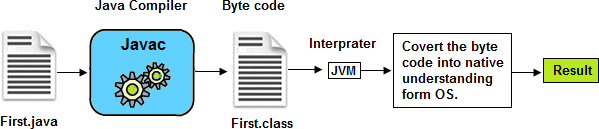
**JRE :** The Java Runtime Environment (JRE) is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application. It physically exists. It contains set of libraries + other files that JVM uses at runtime.

**JDK** : The Java Development Kit (JDK) is primary components. It physically exists. It is collection of programming tools and JRE, JVM

**Features Of JAVA :**

Features of a language are nothing but the set of services or facilities provided by the language vendors to the industry programmers. Some important features are;



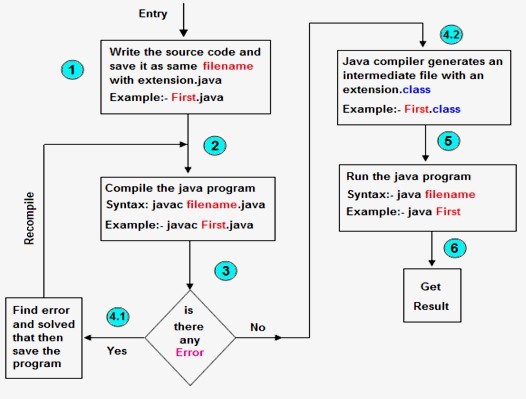


Path variable is set for providing path for all java tools like java, javac, javap, javah, jar, applet viewer which are use in java programming. These all tools are available in bin folders so we set path up to bin folders.

Classpath variable is set for providing path for predefined java classes which is used in our application. All classes are available in lib/rt.jar so we set classpath up to lib/rt.jar.

Steps For Compiling and Executing Programs:

The following sequence of steps represented in the diagram use compiling the java program and executing the java programs.



# CHAPTER 5 : TESTING

* 1. **: UNIT TESTING**

# Introduction

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

# Benefits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

**1) Find problems early** : Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests emselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

1. **) Facilitates Change** : Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.
2. **) Simplifies Integration** : Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.
3. **) Documentation** : Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit.

# : INTEGRATION TESTING

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

# Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter- process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages. Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The correspondences’ for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatic complexity of the software and software architecture, reusability of modules and life-cycle and versioning management. Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client- server integration, distributed services integration and high-frequency integration.

# Big Bang

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.

A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like

environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment.

# Top-down And Bottom-up

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage. Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module. Sandwich testing is an approach to combine top down testing with bottom up testing.

# : SOFTWARE VERIFICATION AND VALIDATION

* + 1. **Introduction**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be

referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review. Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between Validation: Are we building the right product Verification : Are we building the product right? According to the Capability Maturity Model (CMMI-SW v1.1) Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use. From Testing Perspective Fault – wrong or missing function in the code. Failure – the manifestation of a fault during execution. Malfunction – according to its specification the system does not meet its specified functionality Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required. Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar: M&S Verification is the process of determining that a ⦁ computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications. M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use. Classification of Methods In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-mission critical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.

# Test Cases

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

# : Black-Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

# Test Procedures

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

# Test Cases

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external

descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non- functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

# : White-Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White- box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white- box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements

.

# Levels

1. **) Unit testing**: White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.
2. **) Integration testing:** White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behavior in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.
3. **) Regression testing**: White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

# Procedures

White-box testing's basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created.

These are the three basic steps that white-box testing takes in order to create test cases: Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white box testing to layout all of the basic information. Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly. Output involves preparing final report that encompasses all of the above preparations and results.

# : SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software

components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

# CHAPTER 6: RESULTS

All the main pages of the “Employee Management System” are build using

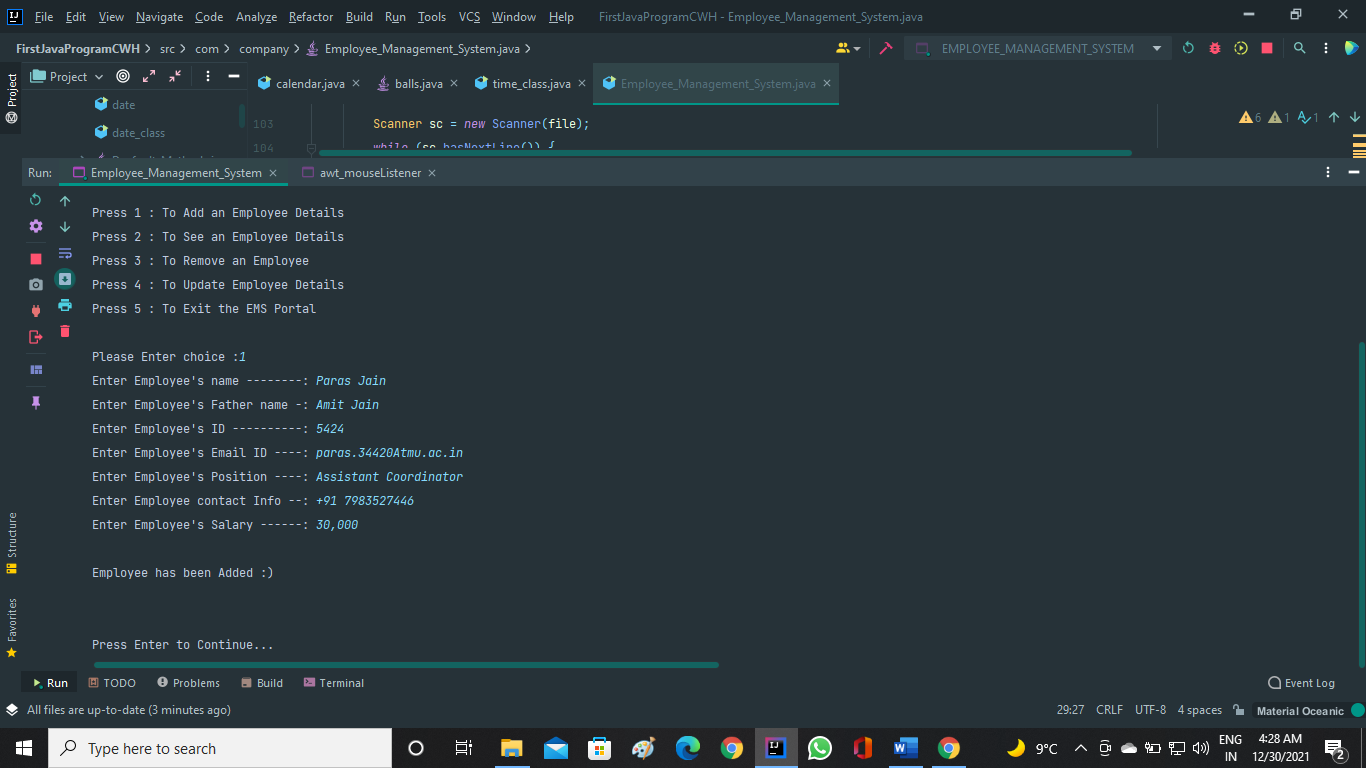
JAVA technology.

Here is the list of pages that we have created:

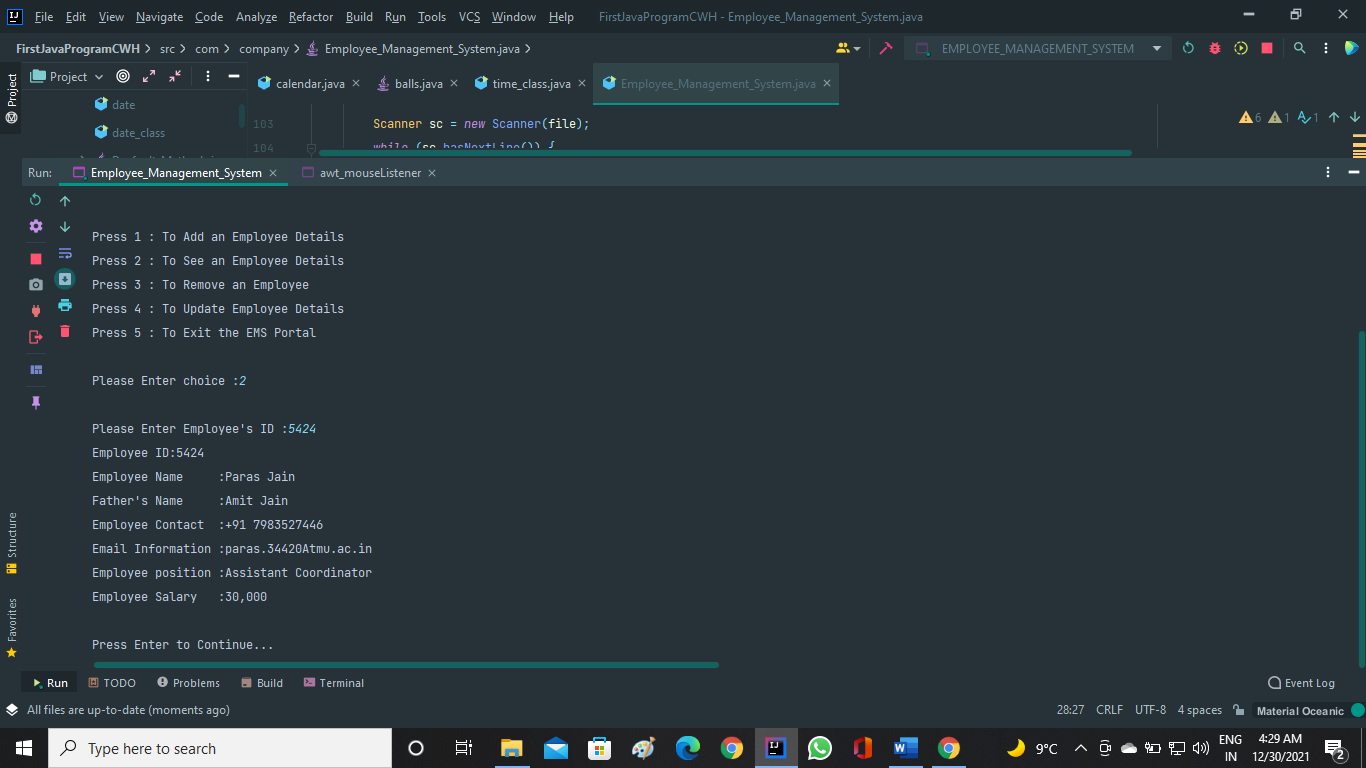
1. Page after running
2. ADD EMPLOYEE page
3. To SEE EMPLOYEE details page
4. To UPDATE EMPLOYEE details page
5. To REMOVE an EMPLOYEE
6. To EXIT the EM-System portal
7. FILE pages where all details are saved
8. **Page after running**

# 

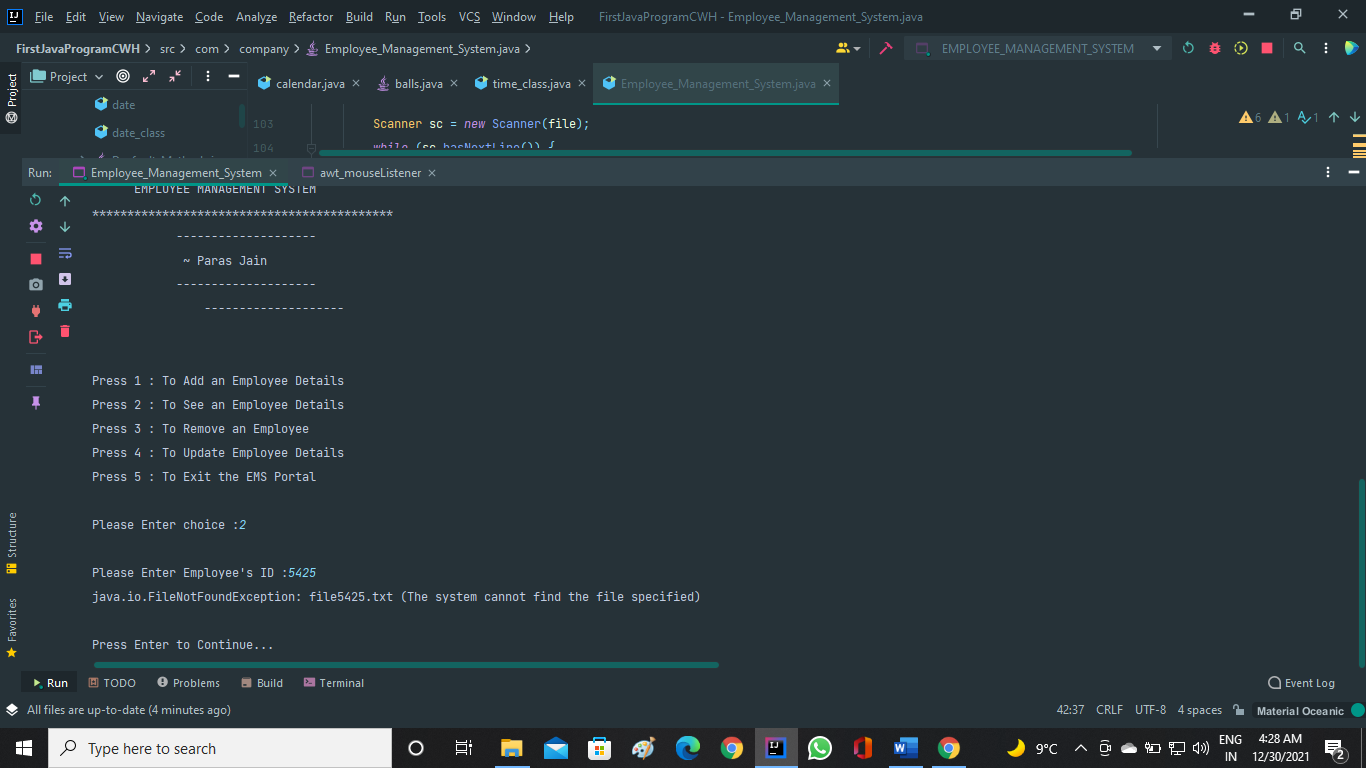
1. **ADD EMPLOYEE page:**



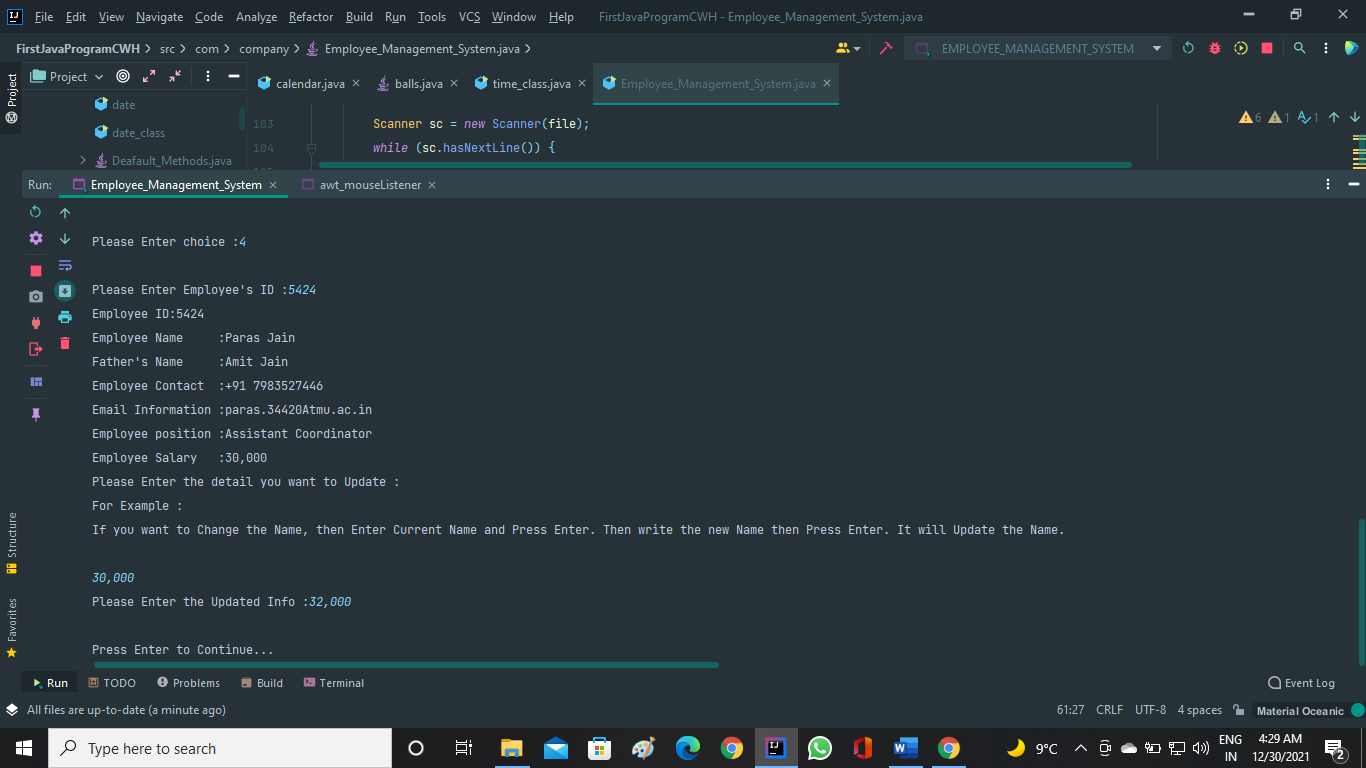
1. **To SEE EMPLOYEE details page:**



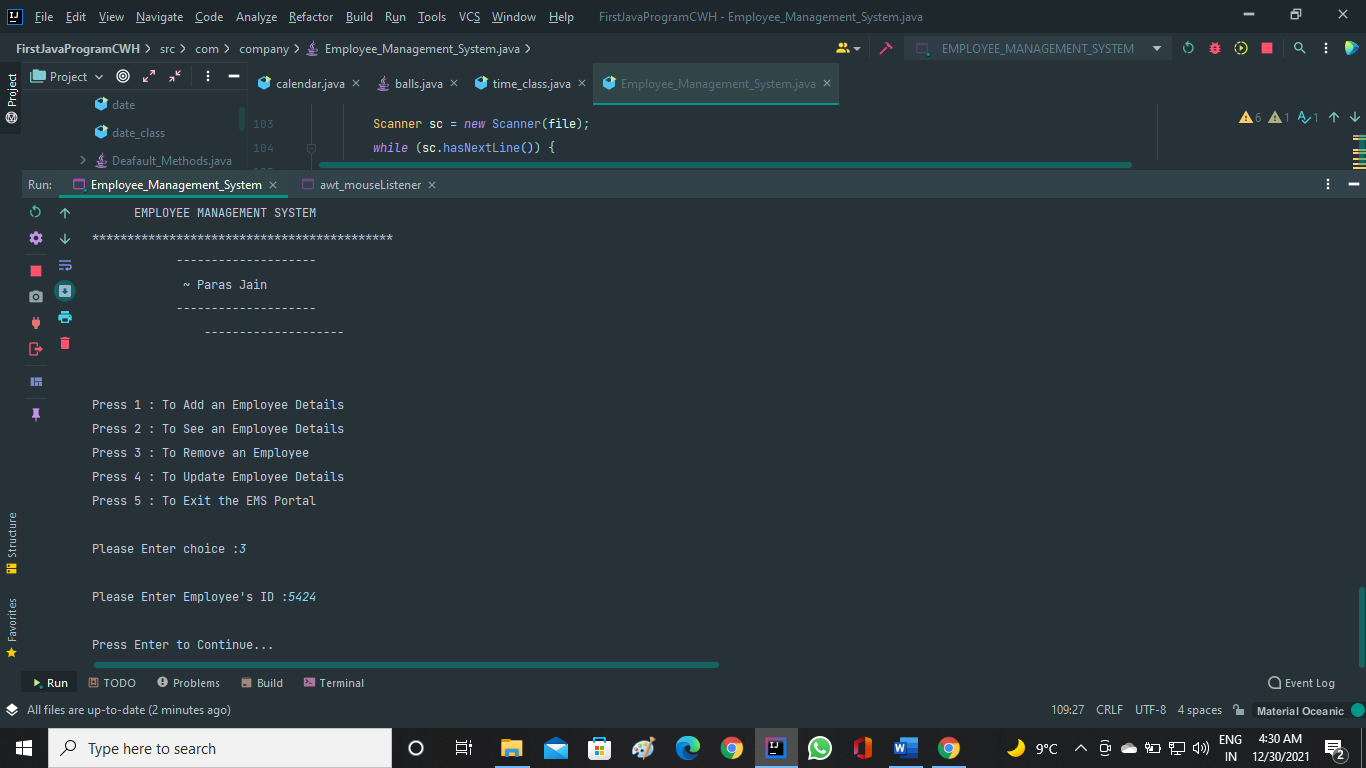
If you enter wrong employee id:



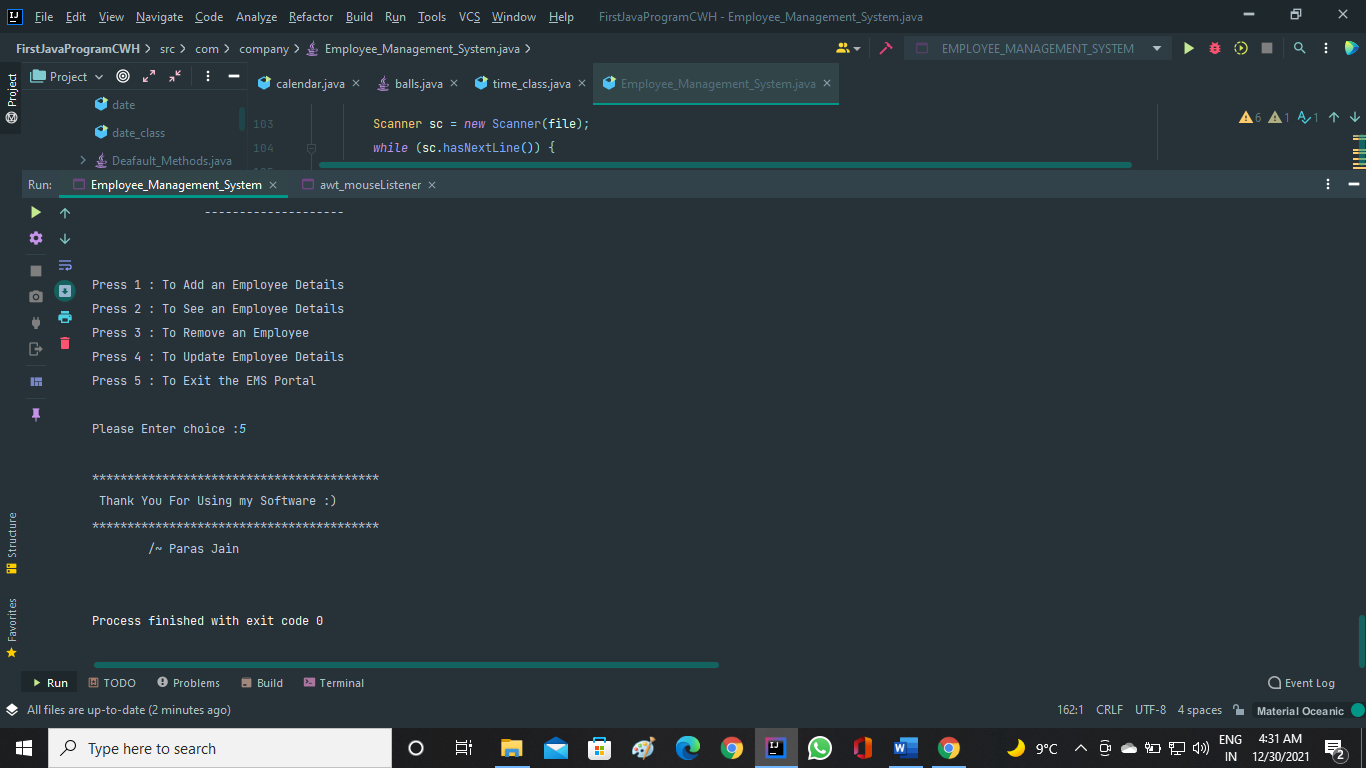
1. **To UPDATE EMPLOYEE details page:**



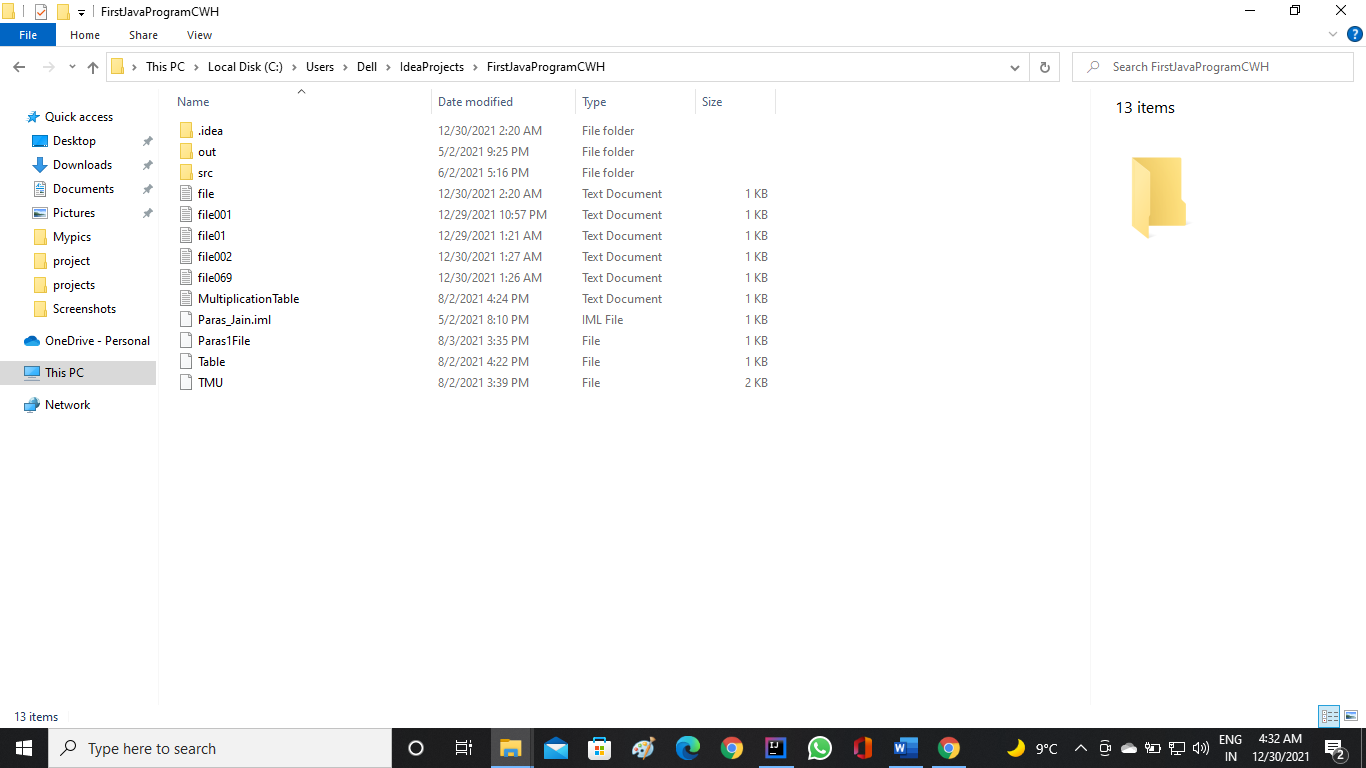
1. **To REMOVE an EMPLOYEE:**



1. **To EXIT the EM-System portal:**



1. **FILE pages where all details are saved:**



# CHAPTER 7 : ADVANTAGES

* Efficiency And Better Accuracy
* Fewer Compliance Risks
* Boosted Profitability
* Very Few Manual Errors
* Higher Productivity
* Higher Motivation
* Lower Costs
* Backup of data
* No paper work
* Reliable
* Convinience

**Source code of Employee Management System:**

package com.company;

import java.util.\*;

import java.io.\*;

// MENU OF EMPLOYEE MANAGEMENT SYSTEM

class MainMenu {

public void menu() {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\t EMPLOYEE MANAGEMENT SYSTEM");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\t\t\t--------------------");

System.out.println("\t\t\t ~ Paras Jain and Shivang Sharma ");

System.out.println("\t\t\t--------------------");

System.out.println("\t\t\t --------------------");

System.out.println("\n\nPress 1 : To Add an Employee Details");

System.out.println("Press 2 : To See an Employee Details ");

System.out.println("Press 3 : To Remove an Employee");

System.out.println("Press 4 : To Update Employee Details");

System.out.println("Press 5 : To Exit the EMS Portal");

}

}

// To add details of Employee

class Employee\_Add {

public void createFile() {

Scanner sc = new Scanner(System.in);

EmployeeDetail emp = new EmployeeDetail();

emp.getInfo();

try {

File f1 = new File("file" + emp.employee\_id + ".txt");

if (f1.createNewFile()) {

FileWriter myWriter = new FileWriter("file" + emp.employee\_id + ".txt");

myWriter.write("Employee ID:" + emp.employee\_id + "\n" + "Employee Name :" + emp.name + "\n" +

"Father's Name :" + emp.father\_name + "\n" + "Employee Contact :" + emp.employee\_contact + "\n" +

"Email Information :" + emp.email + "\n" + "Employee position :" + emp.position + "\n" +

"Employee Salary :" + emp.employee\_salary);

myWriter.close();

System.out.println("\nEmployee has been Added :)\n");

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

} else {

System.out.println("\nEmployee already exists :(");

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

}

} catch (Exception e) {

System.out.println(e);

}

}

}

// Taking Employee Details

class EmployeeDetail {

String name;

String father\_name;

String email;

String position;

String employee\_id;

String employee\_salary;

String employee\_contact;

public void getInfo() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter Employee's name --------: ");

name = sc.nextLine();

System.out.print("Enter Employee's Father name -: ");

father\_name = sc.nextLine();

System.out.print("Enter Employee's ID ----------: ");

employee\_id = sc.nextLine();

System.out.print("Enter Employee's Email ID ----: ");

email = sc.nextLine();

System.out.print("Enter Employee's Position ----: ");

position = sc.nextLine();

System.out.print("Enter Employee contact Info --: ");

employee\_contact = sc.nextLine();

System.out.print("Enter Employee's Salary ------: ");

employee\_salary = sc.nextLine();

}

}

// To Show details of Employee

class Employee\_Show {

public void viewFile(String s) throws Exception {

File file = new File("file" + s + ".txt");

Scanner sc = new Scanner(file);

while (sc.hasNextLine()) {

System.out.println(sc.nextLine());

}

}

}

// To Remove Employee

class Employee\_Remove {

public void removeFile(String ID) {

File file = new File("file" + ID + ".txt");

if (file.exists()) {

if (file.delete()) {

System.out.println("\nEmployee has been removed Successfully");

}

}

else {

System.out.println("\nEmployee does not exists :( " );

}

}

}

// To Update details of Employee

class Employee\_Update {

public void updateFile(String s,String o,String n) throws IOException {

File file = new File("file"+s+".txt");

Scanner sc = new Scanner(file);

String fileContext="";

while (sc.hasNextLine()) {

fileContext = fileContext + "\n"+sc.nextLine();

}

FileWriter myWriter = new FileWriter("file" + s + ".txt");

fileContext = fileContext.replaceAll(o,n);

myWriter.write(fileContext);

myWriter.close();

}

}

// To Exit from the EMS Portal

class CodeExit {

public void out() {

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" Thank You For Using our Software :) ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\t\t/~ Paras Jain and Shivang Sharma \n");

System.exit(0);

}

}

// Main Class

class Employee\_Management\_System {

public static void main(String args[]) {

// To clear the output Screen

System.out.print("");

Scanner sc=new Scanner(System.in);

Employee\_Show epv =new Employee\_Show();

int i = 0;

// Calling Mainmenu Class function

MainMenu obj1 = new MainMenu();

obj1.menu();

// Initialising loop for Menu Choices

while(i<6) {

System.out.print("\nPlease Enter choice :");

i = Integer.parseInt(sc.nextLine());

// Switch Case Statements

switch(i) {

case 1:

{

// Creating class's object and calling Function using that object

Employee\_Add ep =new Employee\_Add();

ep.createFile();

System.out.print("");

obj1.menu();

break;

}

case 2:

{

System.out.print("\nPlease Enter Employee's ID :");

String s=sc.nextLine();

try {

epv.viewFile(s);

}

catch(Exception e){

System.out.println(e);

}

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

System.out.print("");

obj1.menu();

break;

}

case 3: {

System.out.print("\nPlease Enter Employee's ID :");

String s=sc.nextLine();

Employee\_Remove epr =new Employee\_Remove();

epr.removeFile(s);

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

System.out.print("");

obj1.menu();

break;

}

case 4: {

System.out.print("\nPlease Enter Employee's ID :");

String I = sc.nextLine();

try {

epv.viewFile(I);

}

catch(Exception e) {

System.out.println(e);

}

Employee\_Update epu = new Employee\_Update();

System.out.print("Please Enter the detail you want to Update :");

System.out.print("\nFor Example :\n");

System.out.println("If you want to Change the Name, then Enter Current Name and Press Enter. Then write the new Name then Press Enter. It will Update the Name.\n");

String s=sc.nextLine();

System.out.print("Please Enter the Updated Info :");

String n=sc.nextLine();

try {

epu.updateFile(I,s,n);

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

System.out.print("");

obj1.menu();

break;

}

catch(IOException e) {

System.out.println(e);

}

}

case 5: {

CodeExit obj = new CodeExit();

obj.out();

}

}

}

}

}

# CHAPTER 8: CONCLUSION

Finally, in Employee management system, we have developed secure, user-friendly interface. This System can take care of each member whether it is an Administrator or Employee. This System will help them to properly manage the details, the employee data will help in growth without creating any hassle. This System is completely secure, there is no chance of any unauthorized access. So, using this system will help in reducing the labor and provide more facility for employee to like the services. And it also helps to chemist for enhancing their business.

# BIBLIOGRAPHY

**1.)** [**https://www.javatpoint.com/java-tutorial**](https://www.javatpoint.com/java-tutorial)

**2.)** [**https://www.youtube.com/watch?v=WRQGCtcuuhw**](https://www.youtube.com/watch?v=WRQGCtcuuhw)