“GUI based Employee Record Management System”

Major Project Report

Submitted in Partial Fulfillment of the
Requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Siddhant Patel
(18BEC107)

Department of Electronics and Communication Engineering,
Institute of Technology,
Nirma University,
Ahmedabad 382 481

May 2022

CERTIFICATE

This is to certify that the Major Project Report entitled **"GUI based Employee Record Management System"** submitted by **Mr. Siddhant Gaurav Patel (18BEC107)** towards the partial fulfillment of the requirements for the award of degree in Bachelor of Technology in the field of Electronics & Communication Engineering of Nirma University is the record of work carried out by him/her under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date: 21/05/2022

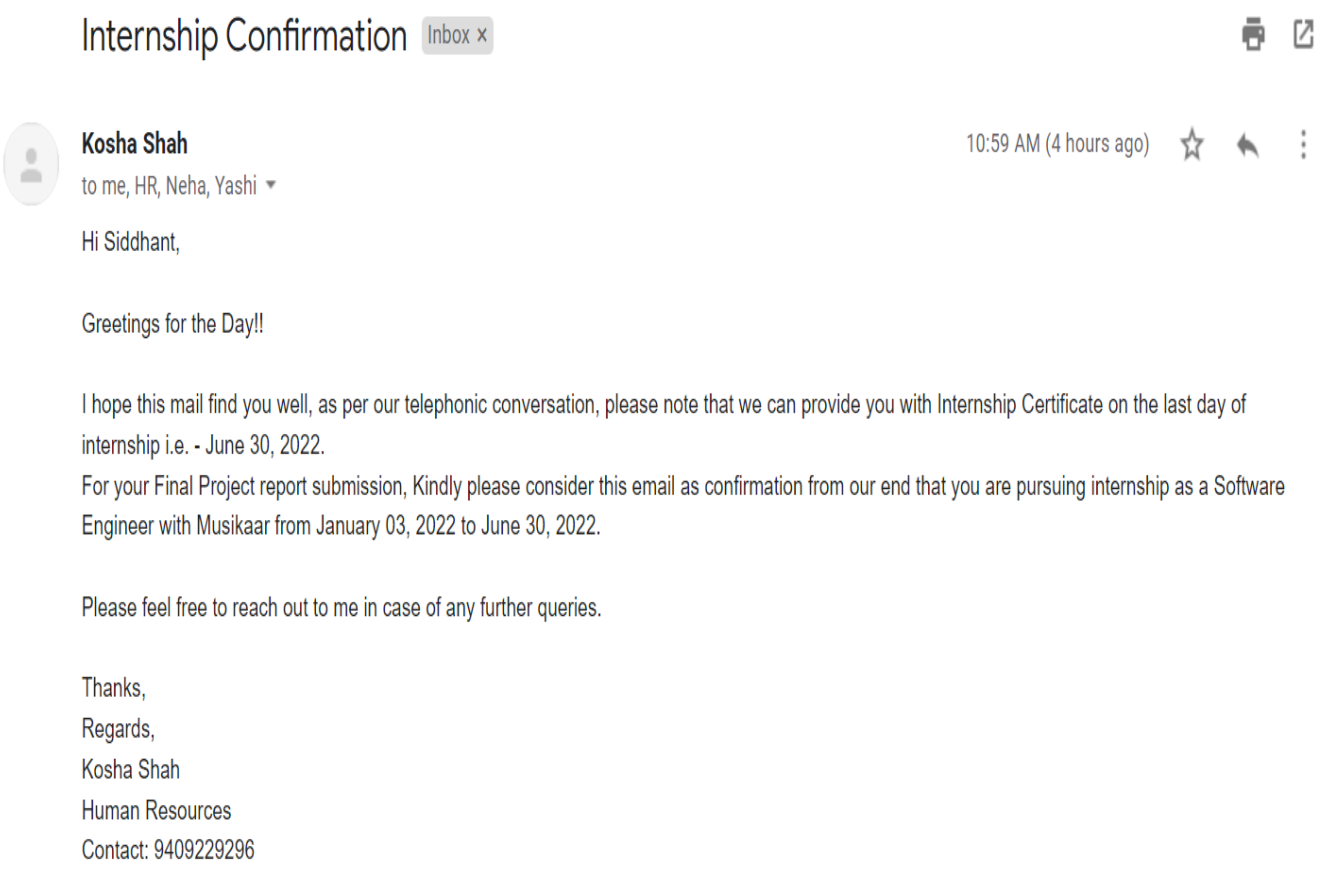
Industry – Guide
Alpesh Sorathiya
Tech Lead
Ahmedabad, Gujarat

Institute - Guide
Prof. Hardik Joshi
Assistant Professor
Ahmedabad, Gujarat

Head of Department
Department of Electronics &
Communication Engineering,
Institute of Technology,
Nirma University,
Ahmedabad 382 481

Director
Institute of Technology,
Nirma University,
Ahmedabad 382 481

CERTIFICATE



Internship Confirmation Inbox ×

Kosha Shah 10:59 AM (4 hours ago)

to me, HR, Neha, Yashi

Hi Siddhant,

Greetings for the Day!!

I hope this mail find you well, as per our telephonic conversation, please note that we can provide you with Internship Certificate on the last day of internship i.e. - June 30, 2022.
For your Final Project report submission, Kindly please consider this email as confirmation from our end that you are pursuing internship as a Software Engineer with Musikaar from January 03, 2022 to June 30, 2022.

Please feel free to reach out to me in case of any further queries.

Thanks,
Regards,
Kosha Shah
Human Resources
Contact: 9409229296

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Siddhant Gaurav Patel (18BEC107)**, a student of BTech in Electronics and Communication Engineering from the Institute of Technology, Nirma University worked in **Musikaar** as a project trainee from **3rd January 2022 to 30th June 2022.** During this period he was found regular and had done his project on **"GUI based Employee Record Management System",** under my supervision.

He has worked with utmost dedication and a high level of engineering and analytical competence.

We wish him all the best in his future endeavors.

Date: 21/05/2022

Alpesh Sorathiya
Musikaar

Undertaking for Originality of the Work

I, **Siddhant Patel, Roll No. 18BEC107**, give undertaking that the Major Project entitled **"GUI based Employee Record Management System"** submitted by me, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Electronics and Communication of Nirma University, Ahmedabad 382 481, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any other published work or any project report elsewhere; it will result in severe disciplinary action.

Signature of the Student

Date: 21/05/2022

Place: Ahmedabad

Endorsed by:

(Signature of External Guide)

(Signature of Internal Guide)

Acknowledgment

This project is the accumulation of all the experiences and teaching whereby I have been accompanied and supported by many people. It is a pleasant aspect that I now have the opportunity to express my gratitude for all of them.

With immense pleasure, I express my sincere gratitude, regards, and thanks to my supervisor Prof. Hardik Joshi for his excellent guidance, invaluable suggestions and continuous encouragement at all the stages of my research work. His interest and confidence in me were the reason for all the success I have made. I have been fortunate to have him as my guide as he has been a great influence on me, both as a person and as a professional.

It was a pleasure to be associated with Musikaar, and I would like to thank my industry guide, Mr. Alpesh Sorathiya. To acknowledge help taken from friends is always a joy. I take this opportunity to convey sincere thanks to my class fellows for their smiles and friendship making the life at Nirma University enjoyable and memorable.

The chain of my gratitude would be definitely incomplete if I would forget to thank the first cause of this chain, using Aristotle's words, The Prime Mover for showering His blessings on me always.

Siddhant Patel

Abstract

The main objective of every invention has always been to make human existence easier and more efficient. One such invention that has gained immense popularity over the years is GUI. GUI stands for Graphical User Interface, which is a visual representation of communication offered to the user for easy machine interaction. It is a typical user interface that incorporates graphical representations like buttons and icons, and communication can be accomplished by interacting with these icons rather than by using text or commands. GUIs can be created for any and every application through the power of programming. Clearly, this holds immense significance when it comes to everyday operations and their time-critical nature since they replace the need of executing long commands with click-and-go buttons. This report includes a detailed description of one such GUI that was developed for a similar purpose; to make the user experience richer for the employees of a company. Currently, the company's nontechnical staff uses a manual method to manage the records for its employees. This is done using a MySQL database and its corresponding queries to specifically perform CRUD operations every time. This is far from efficient and takes a lot of time to do. It is also tedious to query the database each time the user has to do the smallest of operations. The GUI application presents a simple solution that allows the user to perform the task of database management without the need of long commands. The methodologies, results and future scope of the project will be included in the rest of this document.

INDEX

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Abbreviations

GUI	Graphical User Interface
DBMS	Database Management System
API	Application Programming Interface

Chapter 1

Introduction

1.1 Background

Musikaar is an established IT firm providing offshore services in Quality Assurance, Software Development and Managed IT services to hi-tech companies in North America. It is based in Ahmedabad, Gujarat and has been in the industry since its establishment in 2009. Since then it has worked with lots of big companies like Tenable, McAfee, etc., providing a variety of client-personalized services. Some case studies being Mobile Device Management, Security Applications, Desktop Virtualization Product, SaaS Applications. I had completed my Summer Training in 2021 at Musikaar and I was fortunate enough to be offered the opportunity to undertake my semester 8 internship there. My guide during the time was Alpesh Sorathiya, who is one of the tech leads at Musikaar specializing in Mobile Apps and he was assigned to continue his guidance for the duration of the Winter Internship.

One of the traits that I appreciate about the company is the fact that they always try to work with projects that would serve some function during its life cycle and not just some gimmick that holde no utility. The aim of the company has always been to provide solutions to problems that their clients face and that was one of the reasons why I chose to return to the company for my Winter Internship.

1.2 Motivation

I have always tried to work outside my comfort zone and the project I was assigned was the best way to gain experience in something I was not familiar with. Not only would I be able to expand my skill set since it involved working in a completely new programming language and environment, but I would also be able to help the company out by presenting a product that would prove beneficial to it. What motivated me further was my passion for logic building and problem solving. To provide the best results, I could research alternatives to softwares, algorithms, etc and incorporate them into the build to facilitate the best possible solution which would be accessible and efficient at the same time.

1.3 Objective

The objective of the internship and henceforth the project was to come up with an efficient alternative to the employee record management system that was currently in place at the company. The final build of the project was expected to be easily accessible by non-technical staff as well as efficient in terms of how fast the user is able to understand the project, its GUI and functionalities. I was instructed to create the project keeping in mind all the requirements of a data management system and depending on how the final build takes shape, it would be taken up for potentially replacing the current system in place.

1.4 Problem Statement

The company's non technical staff uses a manual method to manage the records for its employees. This is done using a MySQL database and its corresponding queries to specifically perform CRUD operations everytime. This is far from efficient and takes a lot of time to do. It is also tedious to query the database each time the user has to do the smallest of operations. The task assigned to me was to come up with a solution to this problem that would not only make the job of the user easier, but much richer in terms of user experience by providing functionalities that facilitate the same. I was to undergo training in Java and MySQL for the same after which I was tasked to research alternative softwares and solutions and implement the final project.

1.5 Approach

As part of my training, I learned everything from the basics of Java, OOPS and JAVA8 concepts, to Swing and AWT for GUI and events. I learned about the different queries related to MySQL's operations for database management. Finally, I incorporated everything I learned during the training into planning out and implementing the Employee Record Management System. My approach involved initiating a connection between the Java API and the MySQL server, followed by extracting login credentials from the front end, supplying them to the backend for successful login after which the user is presented with the full functionalities of the DBMS through a very easy to access GUI. I planned the final build of this project to consist of only 1 folder, containing all the dependencies and a single click and go file to boot the project. The detailed description of each of the above mentioned steps will be provided in the following sections of the report.

1.6 Scope of the project

The project includes a folder containing the dependencies and a click and go file that runs a Java API in the backend connecting to a MySQL server that holds the employee record database. Once the application is up and running, the user is allowed to then login and access the database to make any changes that they want. Along with the CRUD operations, the user is also able to make user of the added functionalities like checking out a particular record or the entire database records depending on the situation demand, export database records to different file formats like .pdf or .csv. It even provides the opportunity to upload the file to a shared Google Drive by utilizing the Google Drive API.

1.7 Organization of Rest of Report

The preceding sections included a brief introduction to the company as well as my motivations for choosing the company. It also included a short description regarding the objective and scope of the same as well as my motivations behind the project. The following sections will discuss in detail the literature review, the software requirements of the project, its functionalities, the concluding remarks and future prospects. Also included within those sections will be screenshots demonstrating the project and flowcharts to facilitate the better understanding of the project and the utility it holds.

Chapter 2

Literature Review

Sr. No	Author	Year	Paper	Scope
1	Rishabh Bajpayi	2020	Employee Management System- International Journal for Modern Trends in Science and Technology, 6(12): 225-234, 2020	This study covers developing a mechanism for them to solve problems at a lower cost. This system will track each employee's attendance and calculate their pay at the end of the month. It also calculates each employee's overtime and total working hours.
2	Prety Diawati	2019	Challenges of Implementing an Employee Management System for Improving Workplace Management Effectiveness, Journal of Environmental 234 International Journal for Modern Trends in Science and Technology Treatment Techniques , Special Issue on Environment, Management and Economy, Pages: 1200-1203,2019.	This research looks at a staff management approach that saves money, time, and energy while also benefiting the team. A system of employee management is established in an organization to simplify the process of maintaining records of maintenance. It aids HR departments in managing employee information. An employee management system is used to improve the efficiency of workplace management. The whole performance and various elements of an employee in an organization are managed by this employee management system.
3	Poonam Walimbe	2018	Review on Java Database Connectivity, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03, Mar-2018	This paper discusses the JDBC connectivity tool, a Java API (application programming interface) that allows Java programmers to connect to databases. The JDBC API is made up of several classes and interfaces defined in the Java programming language that provide a variety of techniques for updating and querying database data. It's a database-oriented relational driver. It allows the Java application to reuse an existing database connection rather than generating a new one.
4	Wei Wei	2019	Design and Implementation of Hotel	By assessing market demand, this platform applies software engineering principles to the design and execution

			Room Management System	of hotel room management systems. The advantages of the Java platform and layered architecture technologies are used to create an advanced, applicable, and dependable system that meets the system's long-term development goal. It realizes the informatization of hotel room management and contributes to the improvement of hotel room management information efficiency.

Table 2.1: List of referred research papers and articles

The research article [1] authored by Rishabh B., covers the development of a database management solution for mobile applications. The proposed solution is based on a smartphone application that requires internet access. To use this system, all that is required is a smartphone. The suggested solution is a mobile application with two apps for two different sorts of users: employers and employees. The data of a system is saved in Firebase's cloud storage. Firebase manages security and offers free email authentication support. Each user is also assigned an automatic unique identifier. It also offers services like email verification and password recovery. This firebase stores data on the cloud firestore, which saves data as collections of documents and fields.

Research article [2] authored by Prety D., covers the challenges encountered during the development of an employee management system. A system of employee management is established in an organization to simplify the process of maintaining records of maintenance. It assists the HR function in managing personnel information. It can be considered a component of a comprehensive Human Resource Management System, as well as a generic personnel management system. An employee management system is used to improve the efficiency of workplace management. The whole performance and various elements of an employee in an organization are managed by this employee management system.

Research article [3] authored by Poonam W., discusses the JDBC connectivity tool, a Java API (application programming interface) that allows Java programmers to connect to databases. The JDBC

API is made up of several classes and interfaces defined in the Java programming language that provide a variety of techniques for updating and querying database data. It's a database-oriented relational driver. It allows the Java application to reuse an existing database connection rather than generating a new one. This is especially useful in the cases of DBMS systems that are operated on the daily through use of APIs

Finally research article [4], authored by Wei W., discusses a platform that applies software engineering principles to the design and execution of hotel room management systems by assessing market demand. The advantages of the Java platform and layered architecture technologies are used to create an advanced, applicable, and dependable system that meets the system's long-term development goal. It realizes the informatization of hotel room management and contributes to the improvement of hotel room management information efficiency.

Chapter 3

Software Design

3.1 Software Requirements

The project is based on a Java Application with different classes containing multiple libraries each serving a different purpose. To make sure the project runs smoothly without any hiccups, there are a few requirements that must be met during the setup process.

Software:

- **Java:** Java is a high level, robust, object-oriented and secure programming language. It even acts as a platform with its own Java Runtime Environment and API. Even to this day Java is one of the most popular languages and with fair reason. Some of the major advantages java holds are:
    - It is easy to learn since unlike every other programming language which come with a learning curve, Java shares many similarities with C, C++ and JavaScript.
    - The Java API is extensive. The standard JDK comes with over 200 built-in packages containing Java APIs
    - It has vastly evolved over the years keeping in mind the older version. Even with incremental improvements over decades, Java does not cause compatibility issues with the previous versions allowing even more versatility in the code.

- **IntelliJ IDE:** Currently the build has been developed in Java v. 17.0.1 using the IntelliJ IDE. The future prospects for the project include the application being platform independent so that the dependencies list can be shortened, but for now the project requires an IntelliJ IDE to be installed on the localhost device. The advantages of IntelliJ were highlighted especially in the early stages of development where I was struggling to familiarize myself with Java. This is due to the incredibly user-friendly interface of the IDE which:-
    - provides easy start up and keyboard shortcuts for everything,
    - provides deep code insight, inspection and context actions, intelligent code completion,
    - easy debug support for libraries and dependencies, etc.

- **MySQL (Community Edition):** MySQL is currently the most popular open source database management system software used for managing the relational database. It is fast, scalable and easy to use. The core of the MySQL database is the MySQL server which we utilize in our project to connect to the Java GUI frontend through the application and inspect into the database tables and make required changes. The server program resides on the same physical or virtual system where the database files are stored, and it is responsible for all interactions with the databases.

3.2 Dependencies

- **Swing:** Java Swing tutorial is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Swing offers many features like:-
    - Lightweight components,
    - Expanded components features,
    - Excellent event handling,
    - Platform Independent,
    - Swing provides more powerful components such as tables, lists, scroll panes, colorchooser, tabbedpane etc.

- **AWT:** AWT provides excellent event handling functionality which is especially useful in applications that require a GUI. In my project, I was able to design a proper, sophisticated GUI which supported click and go functionality using AWT's "java.awt.event" package which allows allocation of events like button presses or clicks, text box entries, etc.

- **SQL:** Java provides a package called the javax.sql which provides the API for accessing and processing data stored in a database using the Java programming language. This also includes a framework whereby different drivers can be installed dynamically to access different data sources. In this project, the JDBC driver has been incorporated which allows many functionalities like initiating a connection to the MySQL database, passing SQL statements to the database as well as reading and writing data from and to the database with a tabular format. This can be seen in detail in the following where I explain the functionality of reading from MySQL database and writing to different file formats.

- **JDBC Connector:** In order to connect to the database, a JAR file containing the specified JDBC driver is required to be incorporated into the Java dependencies. This JDBC connector is one of the standard APIs for database connectivity, using which we can easily run our queries, statements, and also fetch data from the database.

- **Java IO:** It is an API that comes with Java which is targeted at reading and writing data. The purpose of this in my project is to store the output stream for facilitating functionalities for storing the MySQL data from the tables to a file format of required choice.

- **Itextpdf:** It is an external toolkit which can be added to the Java dependencies as a JAR file. It offers one of the best-documented and most versatile PDF engines in the world (written in Java and .NET), which allows you to not only integrate PDF functionalities into your workflow, but also in your applications, processes or products.

- **Gradle:** It is a build automation and management system designed specifically for building Java-

based projects. The main utility of the Gradle framework here is the Google Services plugin it offers. In order to provide certain functionalities in the project, I incorporated a Gradle build of the Drive API plugin with the GUI to turn a complex task into a simple button press operation.

3.3 Project Design:

In this section, I will be presenting a detailed description of the initial MySQL server setup, the build of the Java application, the implementation of all the function classes, their connections with each other as well as the GUI classes.

MySQL Server setup:

The main prerequisite for this project is that a MySQL server is running on the localhost that the application will be operating on. This is very important as if the server is not operational, the application will not be able to initiate a connection with the server and thereby the database.

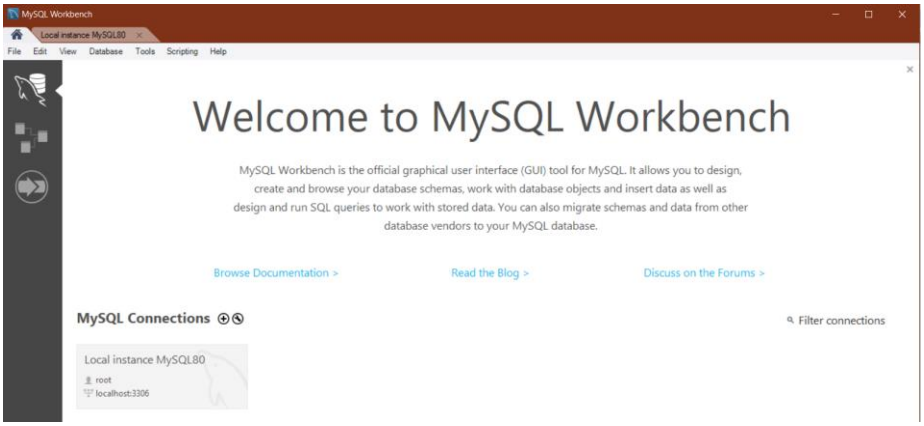


Fig. 3.1: MySQL Connection with Localhost

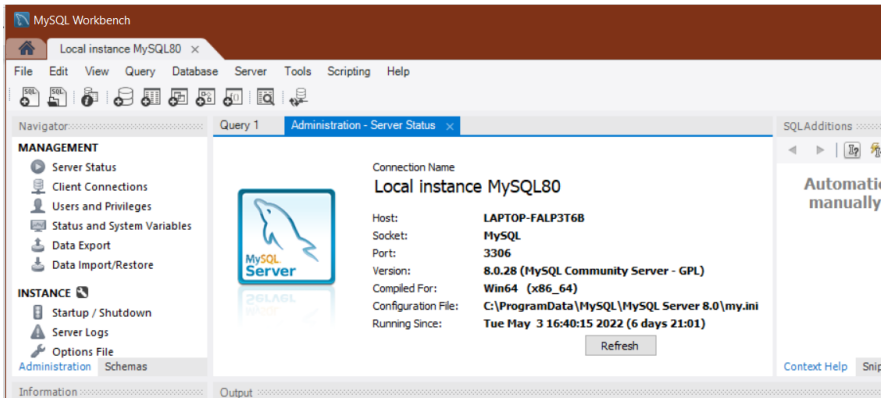


Fig. 3.2: MySQL Server running on localhost

Referring to figures 3.1 and 3.2, we can see that the MySQL server has been initialized on the localhost device and the service is running which means we are set to interface with the Java application for creating a DBMS.

Java Application Design:

This section involves description about the project structure, the function that each class in the structure performs and the GUI classes.

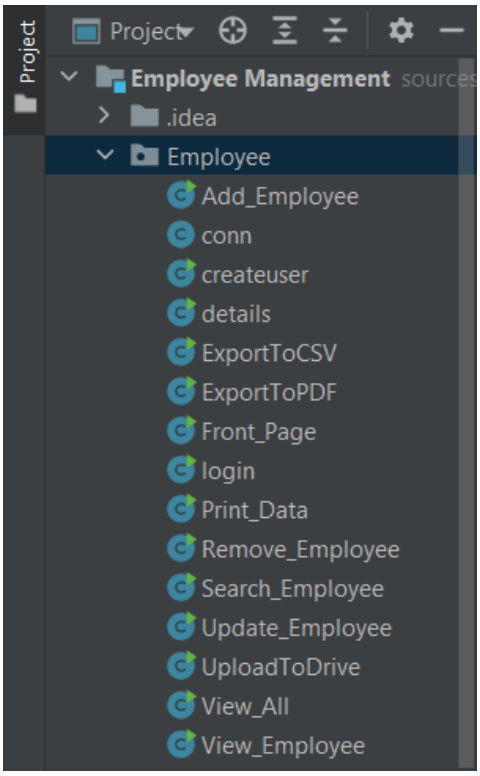


Fig. 3.3: Project Structure depicting Java classes

As depicted in figure 3.3, we can observe multiple java classes each serving a unique purpose in the overall system. The following section will lay out the characteristics of each of these classes.

1. Front_Page: This is the main class that is run first and is responsible for displaying the front page or the home screen of the GUI. The home screen displays the company logo, the title of Record Management System as well as a button which leads the user to the login screen as depicted in figure 3.4 below.

Fig. 3.4: Home screen of GUI

The GUI in this class as well as all other classes is planned and laid out using the Swing library which provides different classes elements like buttons, frames (i.e windows), panels, imageicons, etc. The AWT library is used to add an ActionListener to the button here (as well as other buttons and text fields in the other classes) and this ActionListener is tied to an actionPerformed function inside the same class which “listens” or detects when the button is clicked. Here, in this case, the button click makes the program call an instance of the login class.

2. login: The login class displays text boxes and prompts for entering the username and password of the MySQL server. These fields are read from the GUI using getText() and getPassword() methods and stored to concatenate as strings to a MySQL query. First, the connection class “conn” is instantiated with an object which is used to prepare and execute the query which compares the approved credentials in the MySQL database table with the ones entered by the user in the GUI. If the credentials match, the program calls the next class in the hierarchy which is the details class and the GUI successfully displays the next frame corresponding to the details page. If the credentials don't match, the GUI displays the pop-up message box stating that the user has entered Invalid credentials and the login box is re-displayed.

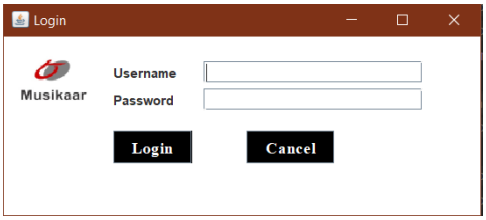


Fig. 3.5: Login Page

3. conn: This is the main class responsible for creating the connection between the Java application and the MySQL server. We utilize the "java.sql" library to access the JDBC driver. To make the connection, we utilize the following information:

- **Driver class:** The driver class for the mysql database is **com.mysql.jdbc.Driver**.
- **Connection URL:** The connection URL for the mysql database is **jdbc:mysql://localhost:3306/siddhantpatel** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running (replaceable with localhost IP address), 3306 is the port number and siddhantpatel is the database name which can be replaced with the database name of the users choice.
- **Username and Password** which were used to set up the MySQL server.

We use the above information to create a connection variable which can be then called by other classes to perform operations on the database.

4. details: This class acts as the main screen displaying all the functions that the application offers.

Fig. 3.6: Details page after successful login

As shown in the figure 3.6 below, there are different buttons that users can press and depending on the event, a new window will appear corresponding to the class that pertains to it. Just like the other classes, all the GUI buttons are laid out using Swing and their events are assigned using the AWT library.

5. Add_Employee: The add button in the details page shown in figure 5, creates an instance of the Add_Employee class on the press of that button. This hides the details page and a new window pops up, as shown in figure 3.7 below.

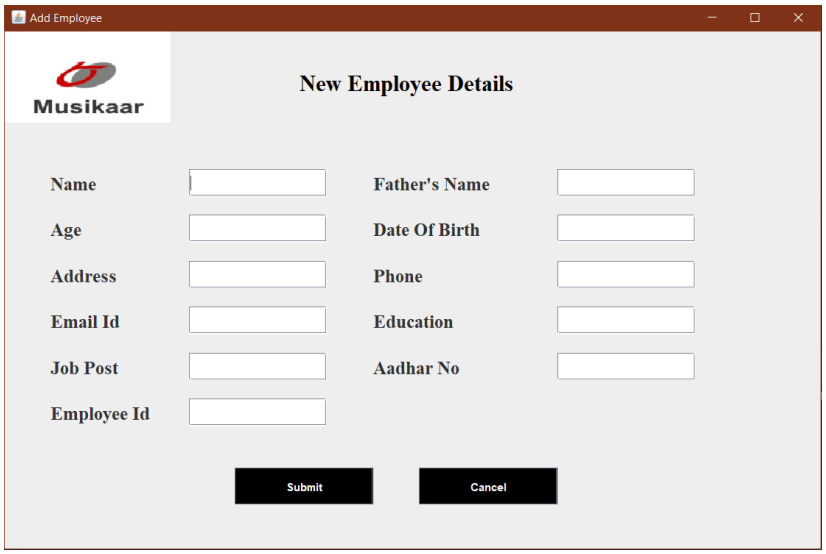


Fig. 3.7: Add new employee window

The GUI prompts the user to add the details of the new employee. Once that is done, the user can choose to submit the data or cancel. If the user decides to submit the data, the application gets the data from the text fields, stores it to concatenate to a query that performs the function of adding the data to the appropriate columns in the database table that holds the records for the employees. If the user decides to cancel this operation, the window is closed and the user is taken back to the details main screen page.

6. Remove_Employee: If the user wants to remove the records of an employee from the database, they can click the remove button on the details page and this hides the main screen and displays a new window that prompts the user to enter the unique ID of that employee. The entered ID is cross checked with the database table and if it matches with any records, the contact details of the employee are added so that the user can contact the employee regarding any confirmation about data removal. Along with

the contact details, the user is also prompted with the button the confirm the removal of the entered employee's details. This can be seen in figure 3.8 below.

Fig. 3.8: Remove employee window

7. Search_Employee: This is a special class made exclusively to interconnect the Add_Employee and the Update_Employee classes. This class displays a search function window when the Update button is pressed on the details page and the user is prompted to enter the ID of the employee whose records they want to update. If the user is found, the Search_Employee class retrieves the ID entered in the GUI and instantiates the Update_Employee class with the ID passed as argument. This is so because when an object of the Update class is instantiated with an argument, the constructor of the Update class calls the Add_Employee class with that argument (here, the argument being the employee ID). What this does is that the Update_Employee window will display the same format of window as the Add_Employee class but the text fields will be filled with the data corresponding to the ID which was passed as argument. This serves the purpose of easing the updating process as the user does not need to type in all the details and they may change only the details that are required to be changed before hitting submit.

8. Update_Employee: If the user wants to make changes to an already existing record, they can click the Update button in the details page. This will redirect the user to the Search Employee window which performs the actions explained in the description above, and the existing details of the employee are prefilled and displayed in the Update window so that it becomes easier for the user to make changes only where necessary instead of having to fill the entire record table.

The same can be seen in figure 3.9 below.

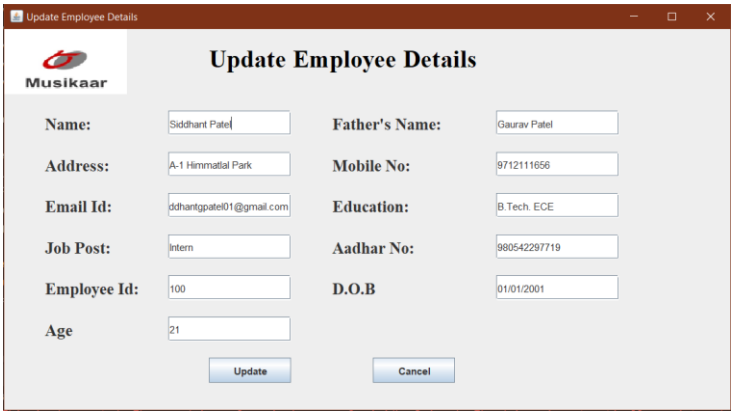


Fig. 3.9: Updating records window

9. View_Employee: This function serves the purpose of displaying the records of the employee whose ID is entered. When the user clicks on the View button on the details page, the View_Employee class displays the window which prompts the user to enter the ID of the employee they would like to view the records of. Once the user enters the ID and clicks on the search button, the Print_Data class is instantiated with the ID entered and it queries the database table to display the records of the employee whose ID matches with the passed argument. If found, the results of the query are stored and displayed in a new window, from which the user can choose to either Print, on the press of which a pdf file of the same is generated or they can go back to the details page on clicking the Cancel button.

Fig. 3.10: View records of particular employee

10. View_All: If the user wants to view all the records in the database table, the user can press the View All button in the details page. This displays a new window with the entire database table containing all the records in a tabular manner in a scrollable panel. This window is also opened without hiding the details page so if the user wants to keep the record table open to refer while using any other functionality of the GUI, they can do so with ease.

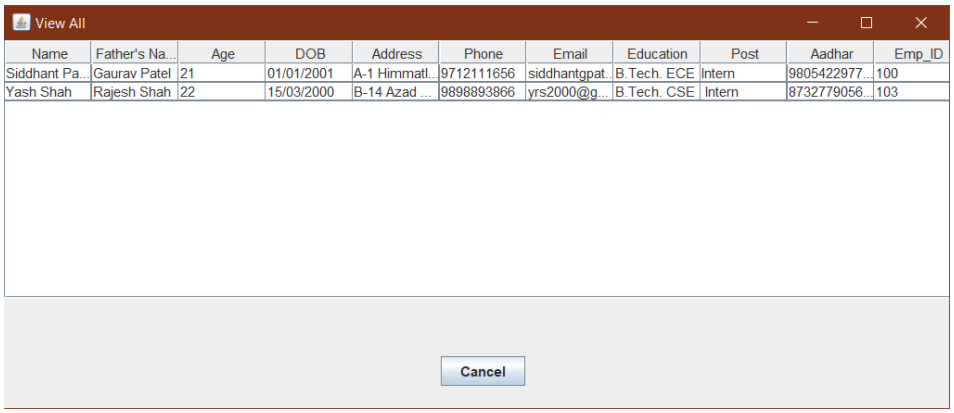


Fig. 3.11: View all employee records

11. ExportToCSV: This class is responsible for reading the tabular data from the database table and writing it to a comma separated file (or .csv file). This along with the other two utilities that will be described below are the ease of access utilities of this application that helps the user save the database table records to different file types as well as Drive storage. This class utilizes the Java IO Filewriter package to read the data from the result set of a query that stores all the records from the employee table, create a file with the stored data and append them cell by cell.

12. ExportToPDF: This class is responsible for reading the tabular data from the database table and writing it to a PDF document. It utilizes the itextpdf package which is one of the most popular packages when it comes to creating pdf files using Java. This class prepares and executes a query that selects all the data from the database table and stores it in a resultset similar to how all the other classes worked. Then using the features of the itext package, an instance of a PDF file is created and a table with the appropriate amount of columns is added. After that the result set is looped over till empty and each data entry is added to the cells sequentially.

13. UploadToDrive: One of the best accessibility features that also took the most time and research was the functionality of uploading database table data to online Google Drive storage. It involves a few steps

of setup to authorize the java application to connect to the Google Drive API. The only use of the UploadToDrive class is to instantiate an object of the ProcessBuilder package so that we can run a Gradle project that is built specifically for this operation. The main job is performed by the Gradle project which can be seen as a series of steps below.

- **Google Cloud Project creation:** The Google Cloud project is required to create, enable, and use all Google Cloud services, including managing APIs, managing permissions, etc. This is important if we are to host our java script as a Google Cloud project and that way we can access all Google Cloud services. For that, we go to the Google API Console and create a new project.
- **Enabling Drive API:** Since we require only the use of the Google Drive service, we go to the available API's section and enable the Google Drive API.
- **Authorizing our application:** All the Google APIs use the OAuth 2.0 protocol for authentication and authorization. In order to access the Drive API, we need to obtain OAuth 2.0 client credentials (which is in the form of a .json file) from the Google API Console. The client application, here our Gradle Project, then requests an access token from the Google Authorization Server, gets the token in response and uses it to access Google Drive API.
- **Creating the Gradle Project:** Now that the setup for the Google Project is done and the credentials for authorization are saved on the localhost, we can create the Gradle Project that runs the script for the uploading process.
    - **Creating basic Gradle project structure:** We initialize a basic Gradle project in a new directory and create separate folders to store the Gradle Java script and the credentials json file so that we can access them at the same place.
    - **Rewrite build.gradle file:** The gradle build file is overwritten with the piece of code that we require to access the Google Drive services. The main contents of this file include the plugin names 'Java' and 'Application' which notify the Gradle project to import any dependencies for running Java and the creating a JVM executable file respectively. It also includes the name of the Main java class which will be run during the execution of the Gradle project. And lastly, it contains the dependencies for the Google OAuth client, the Client API as well as the Drive API.
    - **Setting up the main Java class:** This file mainly contains the global variable instances of application name, the json factory variable which is required to read and write Java to Json

and vice versa, the directory path for saving authorized credentials, a list that stores the Drive access scope (in our case read and write) and the credentials path.

- **Functions of the main class:**
  - getCredentials(): Accepts a Network HTTP Transport type variable to access data from the Cloud via the web. The function retrieves the data from the credentials json file as an inputstream and delivers it ahead to be stored in a GoogleClientSecrets type variable after conversion from JSON to JAVA using the Json factory variable. Then using the information from the NetHTTPTransport, the ClientSecrets and the Scopes, an authorization request is built and the returned object is an Authorizer Credentials object.
  - main(): This function is responsible for building the main authorized API client service. A Google NetHTTPTransport variable is instantiated using the newTrustedTransport() method and used to build a new Drive service variable using the credentials object returned when called using the aforementioned HTTP Transport variable. This Drive service variable is now an authorized client ready to access the Drive services. Lastly, we create a new file to write into the Google Drive of the account linked with the Cloud Console and after running the gradle script we can see that the file is successfully uploaded to the Drive.

3.4 Flow of Project

The preceding section describes in detail the project structure as well as the working of the different classes and how they relate to each other. This section involved a graphical representation of the same to better understand the class hierarchy. Figure 11 represents all the Java classes and their hierarchy and figure 12 shows the basic flow of project from the user's point of view.

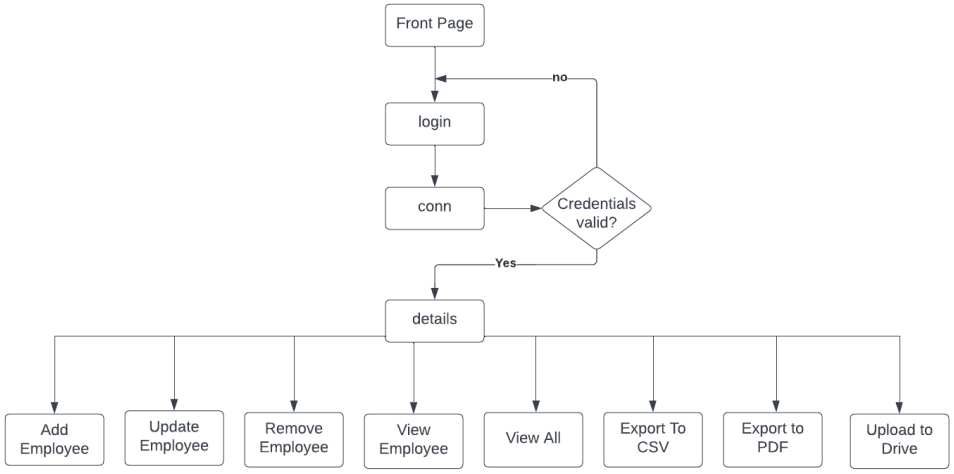


Fig. 3.12: Flowchart representation of Classes and their hierarchy

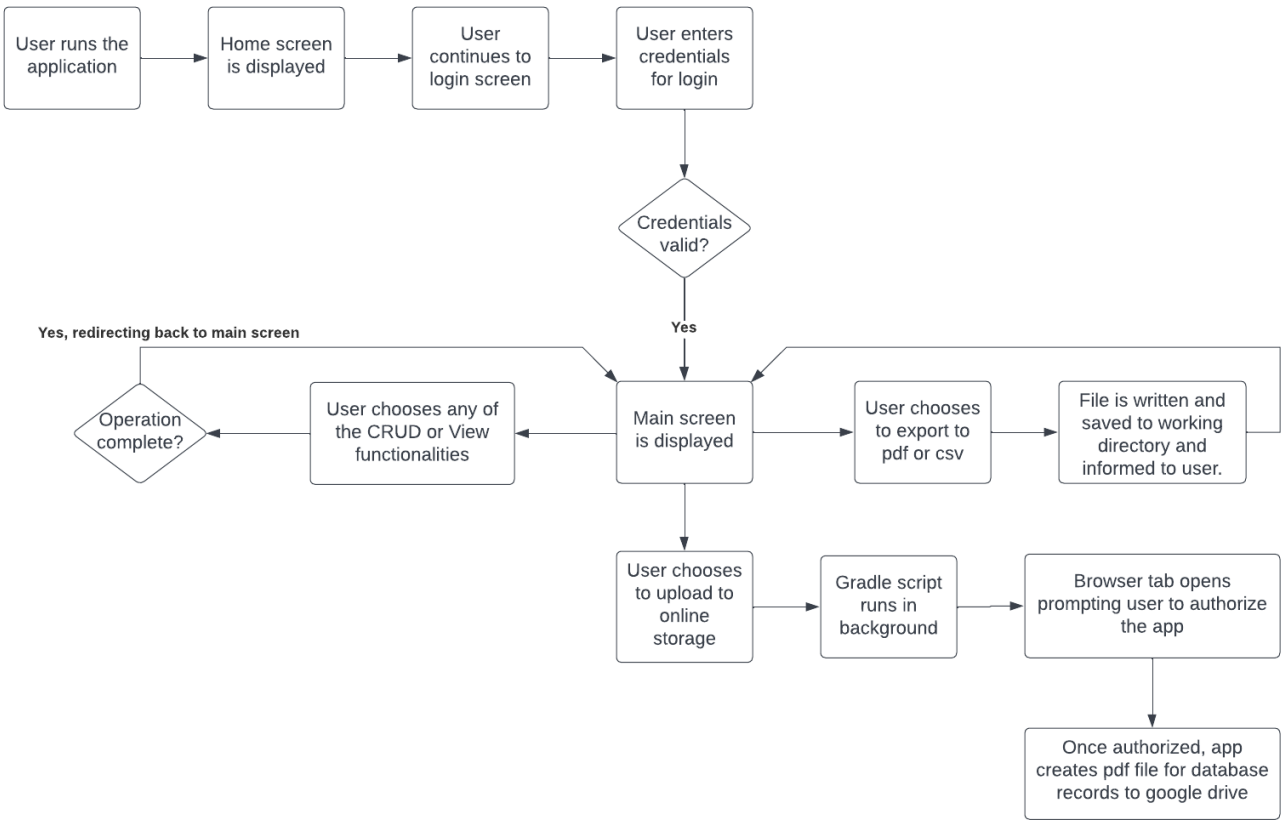


Fig. 3.13: Flow of application from user's point of view.

Chapter 4

Results and Discussion

4.1 Application Performance

This section provides screenshots of the final build of the project as well as the performance of the GUI based Employee Management System from the user's point of view. Each function and its usage has been shown so as to demonstrate the utility of the application.

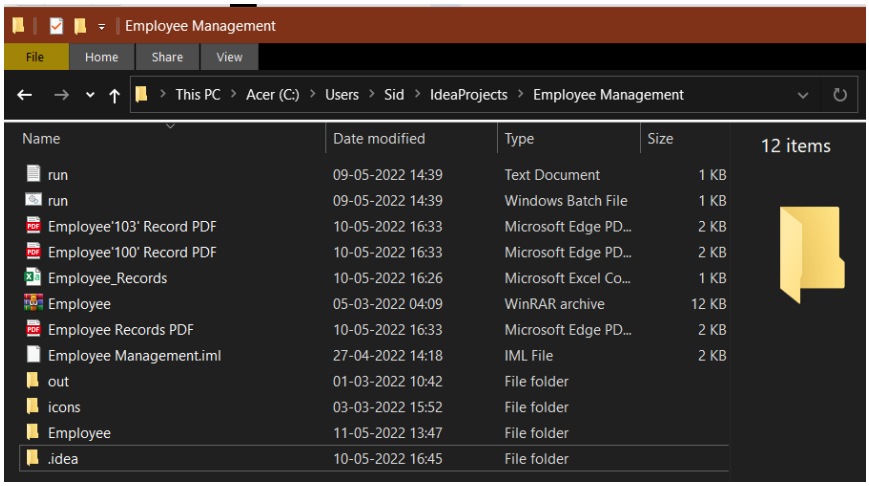


Fig. 4.1: Final Project build folder

In order to run the application, the user can simply click on the run.bat batch file and the java application is run through the cmd process. This will display the home screen as depicted below.

Fig. 4.2: Home screen

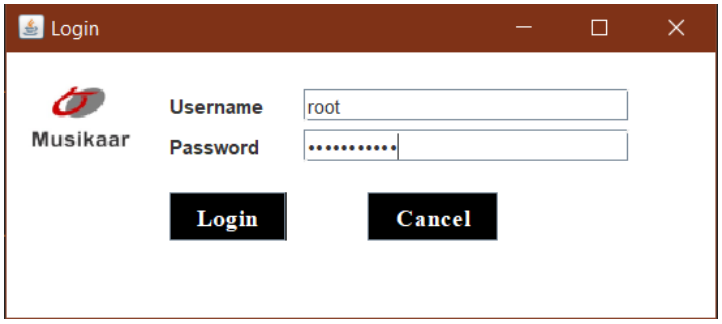


Fig. 4.3: Login using admin credentials

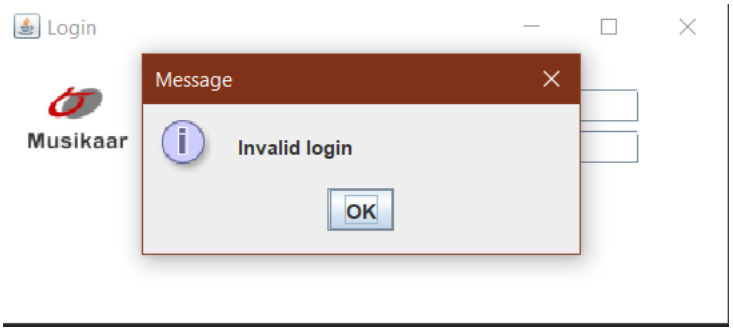


Fig. 4.4: Error in case of invalid login

Fig. 4.5: Main Menu

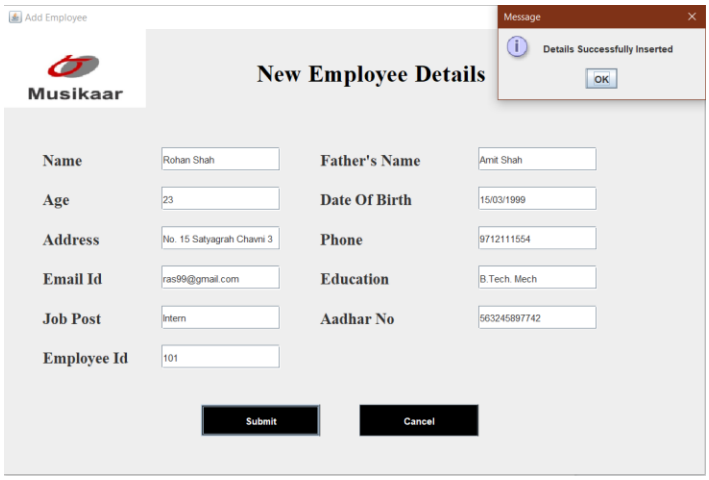


Fig. 4.6 Adding a new employee

Fig. 4.7: Viewing existing employee

Fig. 4.8: Details of entered employee ID

Fig. 4.9: PDF of entered employee is generated and saved

---

name	fname	age	dob	address	phone	email	education	post	aadhar	emp_id
Siddhant Patel	Gaurav Patel	21	01/01/2001	A-1 Himmatlal Park	9712111656	siddhantgpatel01@gmail.com	B.Tech. ECE	Intern	980522977 19	100

Fig. 4.10: PDF of entered employee

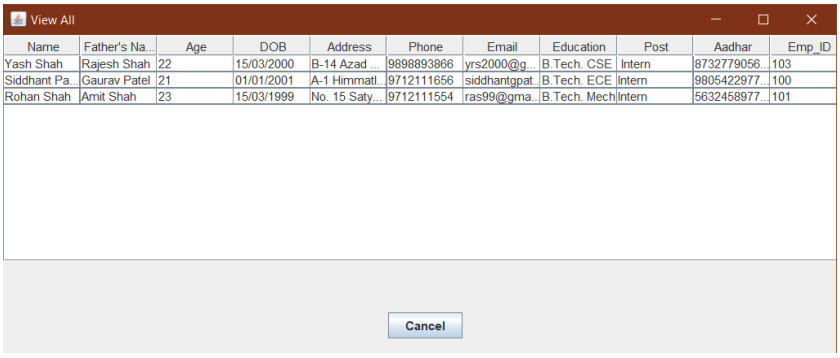


Fig. 4.11: Viewing all the records currently in database

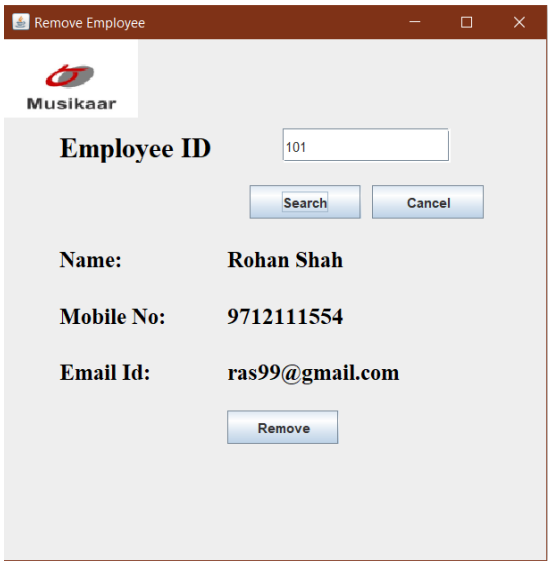


Fig. 4.12: Removing specific employee

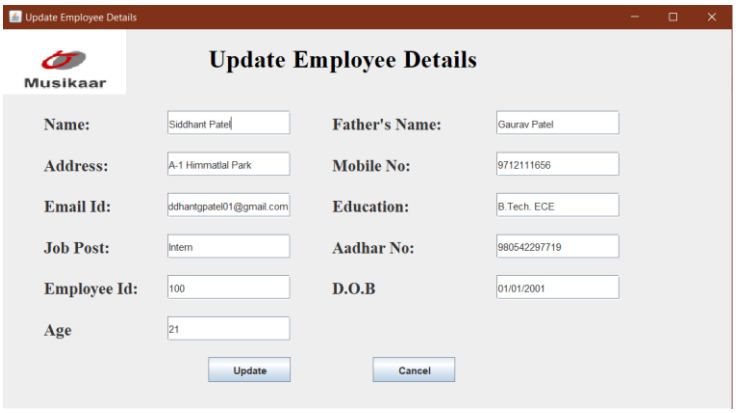


Fig. 4.13: Updating an employee's records

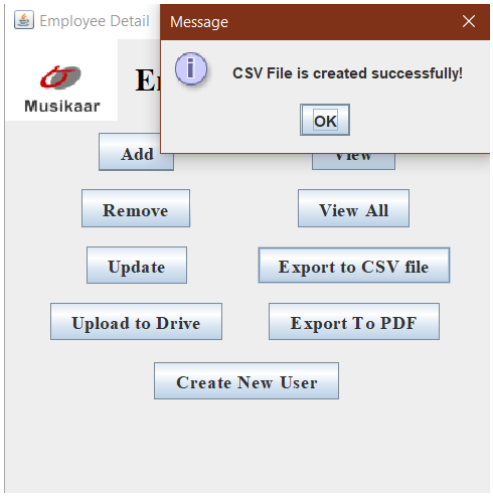


Fig. 4.14: Export database to csv

	A	B	C	D	E	F	G	H	I	J	K
1	name	fname	age	dob	address	phone	email	education	post	aadhar	emp_id
2	Yash Shah	Rajesh Sha	22	########	B-14 Azad	9.9E+09	yrs2000@	B.Tech. CS	Intern	8.73E+11	103
3	Siddhant P	Gaurav Pa	21	########	A-1 Himma	9.71E+09	siddhantgp	B.Tech. EC	Intern	9.81E+11	100
4	Rohan Sha	Amit Shah	23	########	No. 15 Sat	9.71E+09	ras99@gm	B.Tech. M	Intern	5.63E+11	101

Fig. 4.15: CSV file of entire database records

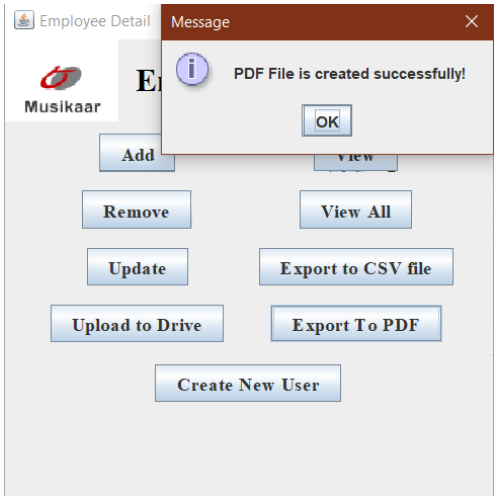


Fig. 4.16: Export database to pdf

name	fname	age	dob	address	phone	email	education	post	aadhar	emp_id
Yash Shah	Rajesh Shah	22	15/03/2000	B-14 Azad Society	9898893866	yrs2000@gmail.com	B.Tech. CSE	Intern	873277905623	103
Siddhant Patel	Gaurav Patel	21	01/01/2001	A-1 Himmatlal Park	9712111656	siddhantgpatel01@gmail.com	B.Tech. ECE	Intern	9805422977719	100
Rohan Shah	Amit Shah	23	15/03/1999	No. 15 Satyagrah Chavni 3	9712111554	ras99@gmail.com	B.Tech. Mech	Intern	5632458977742	101

Fig. 4.17: PDF file of entire database records

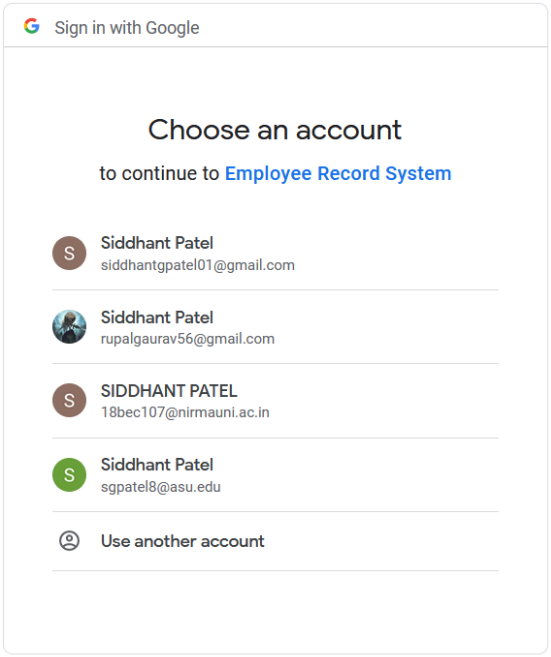


Fig. 4.18: Clicking on upload button redirects to google login

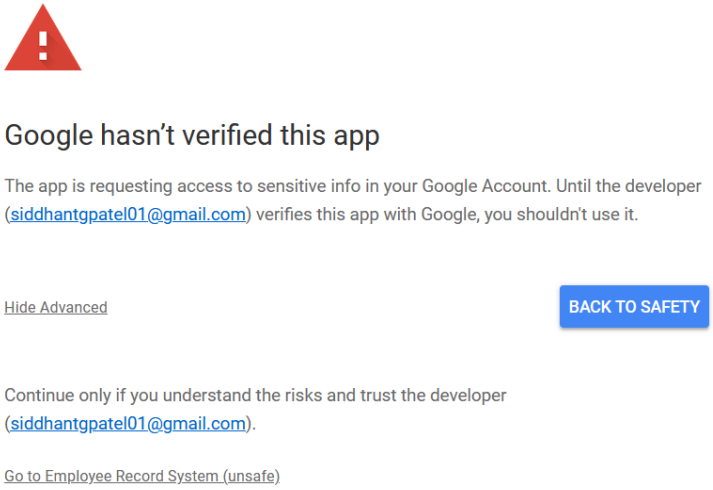


Fig. 4.19: Authorizing the application

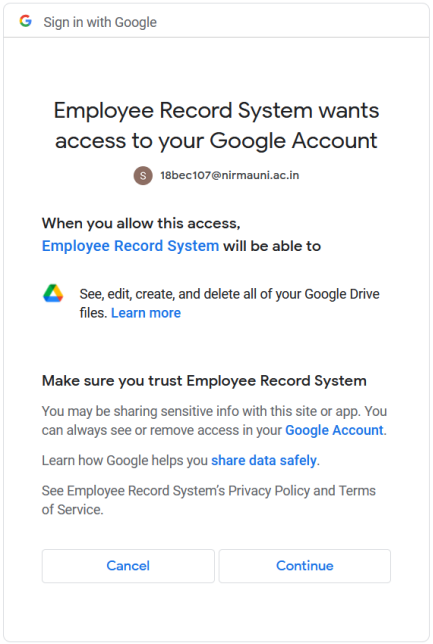


Fig. 4.20: Allowing application to access drive

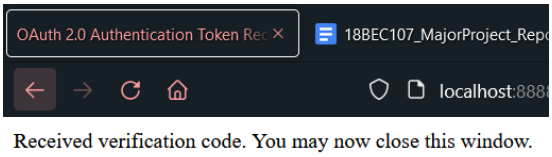


Fig. 4.21: Authorization complete message from OAuth 2.0

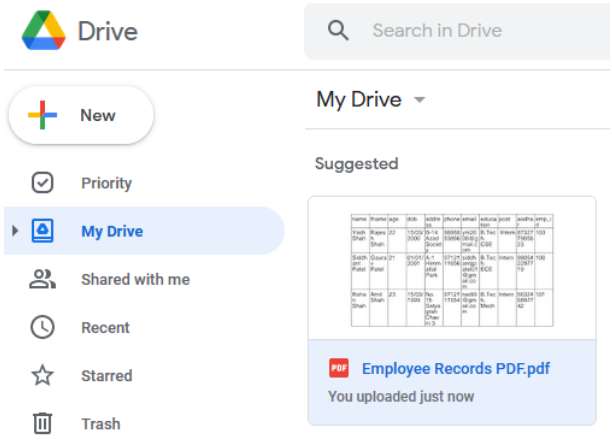


Fig. 4.22: PDF uploaded to Google Drive

4.2 Project Functionalities

The project is an extremely easy to use and user friendly application designed to provide the following functionalities all through a single GUI:-

Adding New Records

The user can easily enter in the information that is prompted by the application and the backend automatically connects to the database table and inputs that information at the press of a button. This can be seen in figures 4.6 which demonstrates the operation of adding a new employee's data to the database.

Removing Existing Records

Consider the case of a company with a staff of hundreds of employees and one of them is relieved from their service. The user can click on the Remove button and enter the ID of the employee being relieved and the GUI displays the contact details of that employee and prompts the user to confirm the deletion of the records from the database. This is depicted in figure 4.12

Updating Existing Records

Updating employee details is always a tedious task when it comes to database management especially when there are lots of data entries with multiple attributes. However, this application eases that process by automatically filling out the text fields with the existing data of the employee whose ID is entered into the GUI. The user can then proceed to update only the attributes that are required to be updated henceforth making the task a lot more efficient. This is depicted in figure 4.13

Viewing Specific Employee Records

Say the user wants to find a certain employee's details, they can click on the View button, supply the ID of that employee and the GUI displays all the information of that employee. Furthermore, the user is also provided with the option of printing that particular employee's records in the form of a pdf file named "Employee 'ID' Records" so that the user can easily identify the file corresponding to that employee. In case the user does not recall the ID of the employee, they can choose the following function. This can be seen in figures 4.7 through 4.10.

Viewing All Employee Records

Clicking on the View All button opens up a new window wherein the entire database table records are printed in a tabular manner. Moreover, the table consists of interactable text boxes so the user can easily copy any of the information regarding any employee easily. This is represented in figure 4.11.

Exporting Database Table as CSV

If the user wants to extract the database records to the localhost system, they can do so at the click of a button and save the records as csv files which can then be opened in any editor like Excel or Sheets to perform further operations on. This can be seen in figures 4.14 and 4.15

Exporting Database Table as PDF

Similar to the CSV file format, the database entries can also be extracted and saved in a pdf format to be shared via Mail or to be printed. This can be seen in figures 4.16 and 4.17

Uploading Database Table Records to Google Drive

If the user wishes to upload the extracted pdf file to an online storage like Google Drive, the Upload button lets the user do exactly that. It opens up a window on the user's device asking the user if they want to authorize the application. Once that is done, the window displays a message saying verification received which means that the application has access to the Drive and the file can now be uploaded. The user can now check the Drive to see that the file will be uploaded to their storage. This can be seen in figures 4.18 through 4.22.

4.3 Project Utility

The final build of the project is an easy to use application that holds a lot of utility in Database Management Systems. As demonstrated in the previous sections, the application provides a lot of functionalities that can be used by a person with any type of background be it technical or non-technical and it serves as an efficient solution to the problem of manual database management. By incorporating a powerful programming language and a well planned GUI, the application not only makes the tedious record management process a lot faster but also much more convenient.

The project not only accomplished the expected objectives but the extra functionalities that I added were appreciated by my mentor at the industry. For the same reason, this project has been presented for further development and potentially being incorporated into the current system at the company.

4.4 Challenges

Prior to starting the internship, I had very little experience in Java and MySQL. Over the course of the last few years, I had mostly undertaken projects that involved programming in Python as well as topics that covered ML/DL. I had never worked with a utility based application that involved lots of error handling

as well as use case planning. The training that I underwent for about a month during the start did help out with the syntax and logic part of the project. However, looking at the project from a user's point of view and finding out bugs and places where there could be errors was challenging. However, through the stages of the project I became more and more familiar with that process and in the end I was able to deliver a more than satisfactory application.

In terms of technical challenges, there were many that I faced when creating the application and its constituent classes. If I was not working with an IDE like IntelliJ, which provides smart code completion and library inclusion, I would have a hard time finding errors in dependencies and syntax. Learning how to plan the GUI, assigning the GUI elements and interlinking them with different events was another challenge to conquer. Once I got familiar with this, I faced yet another challenge of linking the different java classes together. For example, trying to incorporate features like Inheritance between the Update and Add classes for the auto-fill functionality was tough to decipher since I encountered many run time errors. A simple solution to the same was to create a buffer class that separated two functions that were previously running simultaneously. Experiences like these developed not only my familiarity with Java but also error handling as well.

One of the major challenges that I faced was the Google Drive API connection to the Java application. Behind the click and go function that somehow instantly uploads the files to the google drive storage, were countless steps that went into the setup and creation of the backend scripts.

However, all these challenges motivated me even more and because of that I was able to perform well above my own as well as my mentor’s expectations.

Chapter 5

Concluding Remarks

5.1 Conclusion

After undergoing proper training in Java and MySQL, I was assigned with the task to create an efficient solution to manual DBMS. I designed an easy to use GUI based Java application that establishes a safe connection between the application and the database. It provides multiple useful functionalities like CRUD operations, Viewing and Display operations, Extracting and Exporting functionalities and even Online Storage capabilities all through the press of a few buttons through a GUI. It is efficient, time saving and easy to understand and it holds a lot of potential to be improved even further in the future. It really makes the task of DB management easy and extremely accessible.

I hope to

5.2 Future Scope

Currently, the application depends on an IDE but in the near future, I plan on making the application platform independent with minimal dependencies so that its utility can increase even more and be more versatile.

There are a few formatting issues in the PDF and CSV files that extract the database data. They are in no way hindering the operation of the DB management however, their visual appeal is subject to improvement so that the user experience can be made even richer.

There are a few other functionalities that I have in mind for the project which are already under development, such as creating new users that can access the database through the GUI itself as well as managing permissions of existing users. However, at the current stage, the project is being deployed to a small team for testing and if in the case the project works well, it might be due for further development which is where this functionality will be planned to be incorporated.

References

[1] Rishabh Bajpayi, Prof. M L Sharma, K C Tripathi, “Employee Management System”, International Journal for Modern Trends in Science and Technology, 6(12): 225-234.

[2] Prety Diawati, Vip Paramarta, Djoko Pitoyo, Tomy Fitrio, Sri Wiyati Mahrani, “Challenges of Implementing an Employee Management System for Improving Workplace Management Effectiveness”, Journal of Environmental 234 International Journal for Modern Trends in Science and Technology Treatment Techniques , Special Issue on Environment, Management and Economy, Pages: 1200-1203,2019.

[3] Poonam Walimbe, “Review on Java Database Connectivity”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03, Mar-2018.

[4] W. Wei and Z. Lou, "Design and Implementation of Hotel Room Management System," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019, pp. 956-961.

[5] Sayali Pramod Dalke, Shruti Anil Deshmukh, Janabai Govind Dalave, Vaishnavi Nitin Sasane, Pooja K Dhule, “Web Based Staff Management System”,IJSTE Vol-3, March-2017.

[6] Pratik Udayshankar Singh, Hemant Singh Fartyal, Khan Abdul Ahad Zubair, Prof. Akshata Laddha, “Employee Management System ”, IRJET, Vol-06, 5 may 2019.

[7] Sadhana J. Kamatkar , Amarapali Tayade , Amelec Viloria , and Ana Hernández-Chacín, “Application of Classification Technique of Data Mining for Employee Management System”, Springer International Publishing AG, part of Springer Nature 2018

[8] Albert C., Hyde J. and M. Shafritz, “Introduction To Tomorrow’s System For Managing Human Resources”, Public Personnel Management. Academic Journal, Vol. 6 Issue 2, pp.70 – 77

[9] Sahajik Ahmed, Xing Ke, “Human Resource Management, Employees career development impact on organizational performance”, ICSSSM, 2016.

Appendix

Websites

[1] https://www.tutorialspoint.com/java_mysql/index.htm

[2] https://www.geeksforgeeks.org/java-database-connectivity-with-mysql/

[3] https://www.javatpoint.com/steps-to-connect-to-the-database-in-java

[4] https://www.javatpoint.com/sql-tutorial

[5]https://stackoverflow.com/questions/7358775/java-gui-frameworks-what-to-choose-swing-swt-awt-swingx-jgoodies-javafx

[6] https://docs.oracle.com/javase/9/docs/api/java/awt/package-summary.html

[7] https://www.javatpoint.com/java-swing

[8] https://www.service-architecture.com/articles/database/mapping-sql-and-java-data-types.html

[9] https://docs.oracle.com/javase/tutorial/uiswing/events/actionlistener.html

[10] https://www.ibm.com/docs/en/i/7.1?topic=topics-running-java-in-batch-job

[11] https://bretcameron.medium.com/how-to-use-the-google-drive-api-with-javascript-57a6cc9e5262