VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY
INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

PRINCIPLES OF DATABASE MANAGEMENT – IT079IU

FINAL REPORT

Instructor: Assoc. Prof. Nguyen Thi Thuy Loan

Topic: Online CVs builder

By Group 02 - Member list

1. Nguyễn Nguyên Hiệu	ITDSIU21087	Team Leader
2. Lê Xuân Tâm	ITDSIU21118	Team Member
3. Nguyễn Ngọc Sang	ITDSIU21114	Team Member
4. Đỗ Tấn Lộc	ITCSIU21199	Team Member
5. Phạm Quốc Huy	ITITIU21215	Team Member
6. Nguyễn Nho Huy Hoàng	ITCSIU22046	Team Member

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURE

I. INTRODUCTION

1. Abstract

With digitalization becoming essential in today's organizations, companies, and businesses, a database management system that securely stores user information is invaluable for effective data management and tracking. This project for the Principles of Database Management course focuses on developing a CV builder—a platform where users can easily create, customize, and format their CVs or resumes. Key features include user account creation, template selection, CV data storage, and PDF exporting. Designed with user-friendliness in mind, the platform simplifies the process of crafting a polished CV, making it accessible even to those without design or formatting skills. The project highlights how database management principles can be applied practically to tackle data storage challenges. Nevertheless, implementing a database management system comes with certain obstacles, particularly in cybersecurity; digitizing CVs raises risks of data theft and misuse by cybercriminals. Additionally, system errors or lack of usability can lead to inefficiencies. Despite these challenges, the benefits of digital CV management significantly outweigh the drawbacks.

2. System overview

Historically, managing paper resumes has posed numerous challenges due to their physical limitations and the need for printing or manual distribution. Paper CVs restrict accessibility, as they are often localized and lack flexibility for quick updates or modifications, requiring reprints for any changes. These limitations frequently lead to the exclusion of relevant information, constraining the display of skills and accomplishments. Additionally, paper CVs are vulnerable to damage, loss, or misplacement, risking the loss of valuable information. The shift to online CVs has become necessary as technology advances, addressing these drawbacks. While paper CVs are still relevant in settings like in-person interviews or networking events, online CVs offer significant advantages in terms of accessibility and customization options.

The system incorporates secure features such as user registration and authentication to ensure safe access. It provides an intuitive interface, allowing users to easily enter and update personal information, work history, education, skills, and achievements. Template options help organize and visually enhance content for a polished presentation.

An online CV expands reach beyond local boundaries, enabling individuals to present their qualifications to a global audience through online platforms and professional networks. In recent years, platforms like LinkedIn, Upwork, FlexJobs, and Remote.co have flourished, connecting professionals and promoting remote job opportunities. Thus, adopting an online CV supports career growth and flexibility in the digital job market.

3. Goal

The objective of developing an online CV builder is to offer a streamlined, user-friendly platform where individuals can easily create, manage, and share professional profiles. Using Java to connect the front-end interface with a back-end database, the system simplifies CV creation through intuitive templates, customizable sections, and multimedia support. It enables users to showcase their skills, qualifications, and experience in an engaging and visually impactful way. Key features include account creation, secure login, verification code sending, template selection, information input, and PDF export. These tools aim to boost user visibility in the job market, fostering connections with potential employers, clients, and professional networks. Additionally, the system allows for testing among users using MySQL Workbench queries.

II. TASK TIMELINE AND DIVISION

1. Contribution

Full name	Contribution
Nguyễn Nguyên Hiệu	16.67%
Lê Xuân Tâm	16.67%
Nguyễn Ngọc Sang	16.67%
Đỗ Tấn Lộc	16.67%
Phạm Quốc Huy	16.67%
Nguyễn Nho Huy Hoàng	16.67%

Table 1. Individual responsibility and contribution

2. Project Timeline & Task Division

Index	Task	Responsible Member(s)	Reviewers	Deadline
Midterm: 28/10/2024				
1	ERD: Build and demonstrate diagram	Nguyên Hiệu, Quốc Huy, Xuân Tâm	Nguyên Hiệu	
2	Outlining relational database model	Huy Hoàng, Tấn Lộc	Tấn Lộc	
3	Mid-term report	Ngọc Sang	All	
4	Mid-term project submission	Nguyên Hiệu	All	09/11
Final: 10/12/2024				
5	Demo	All	Nguyên Hiệu	27/11
6	Project Report	Quốc Huy, Ngọc Sang, Nguyên Hiệu	All	6/12
7	Slides for presentation	Huy Hoàng, Xuân Tâm	All	6/12
8	Synthesize request information and complete report	All	All	7/12
9	Meeting review	All	All	8/12
10	Presentation and submission	All	All	10/12

Table 2. Planning and task divisions for member of team

III. PROJECT ANALYSIS

1. Requirement Analysis

The Online CV Builder project aims to provide users with a straightforward, user-friendly platform for creating professional CVs. By focusing on modularity and scalability, this project is designed to accommodate a range of customizable templates and sections, including education, work experience, and skills. The platform's database architecture ensures efficient storage and retrieval of CV data, while secure user authentication protects sensitive information. This project serves as a practical application of database management principles, offering students an opportunity to explore real-world data handling and user experience design in the field of Data Science

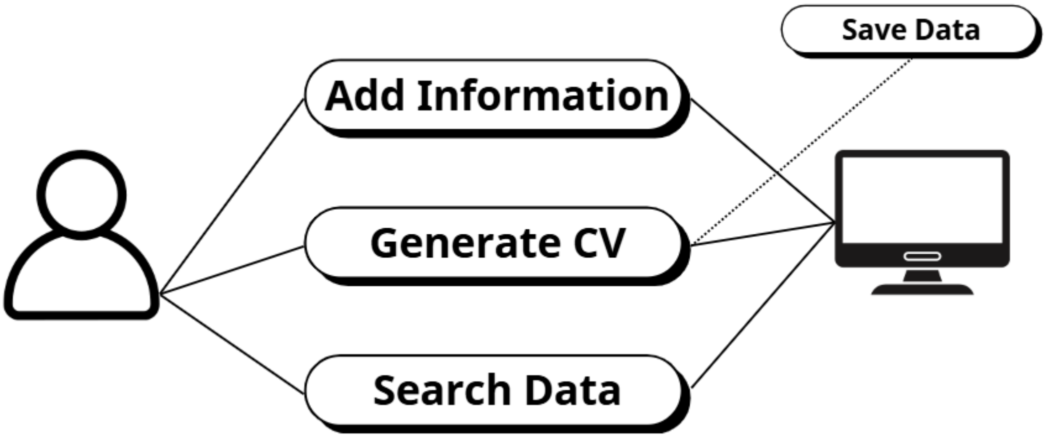


Figure 3.1 Diagram of Online CV Builder

a) **Add Information**
Users will fill in their information in the available information boxes. Then the information will be included in the CV.

b) **Generate CV**

After filling in the information, the CV will be created and can be edited to suit the user. In addition, in this step, the user's information will be saved to the database.

c) **Search Data**
This is a function that allows users to search for information of other users that has been previously saved in the database.

2. System Analysis

2.1 Database Design: Entity Description to ERD Model

a. **User**: This entity stores the sign in credentials of users who generate their own CV. The attributes of this entity are:

- UserID(Primary Key): An unique identifier for each user
- UserName: Name of user for searching
- Status: the status of user whether if user generate a CV

b. **Information:**

- ID (Primary Key): An unique identify
- firstname: the user's first name
- surname: the uses’s surname
- add: the user’s address
- postcode: the user’s postcode
- nationality: the user’s nationality
- dob: the date of birth
- telephone: the phone number of user
- company: the company that user have ever working
- wdone: time worked at that company
- university: The university that the user has studied
- email: the user’s email
- skill: the skills that user has
- qual: the qualification that user has and bring benefits for company

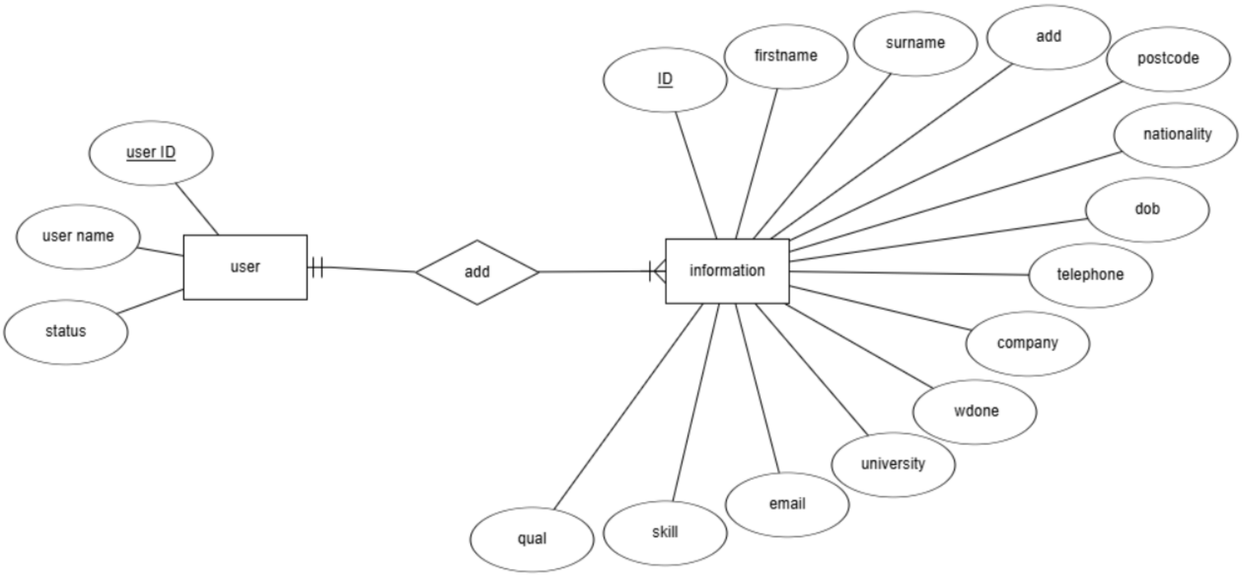


Figure 3.2 ERD

2.2 Analyzing Relationship of System

The relationship between 'User' and 'Information' is a mandatory one-to-many relationship

- Information can be added in many rows by a single user. All information rows are linked to single user
- Each user is individually identified by their userID, a primary key in user table
- ID is a primary key in the information table that unique identifies rows of information
- A foreign key named userId exists in the information table and refers to the userId found in the user table

2.3 Relation Model Analysis

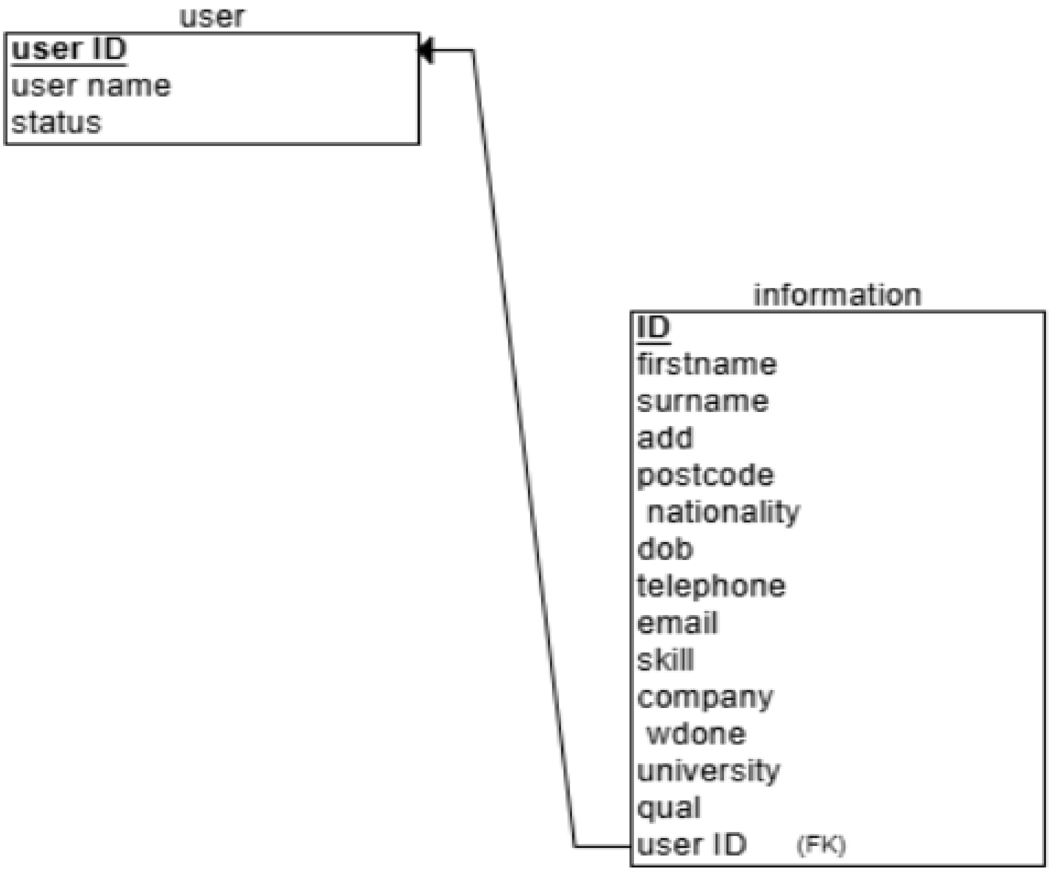
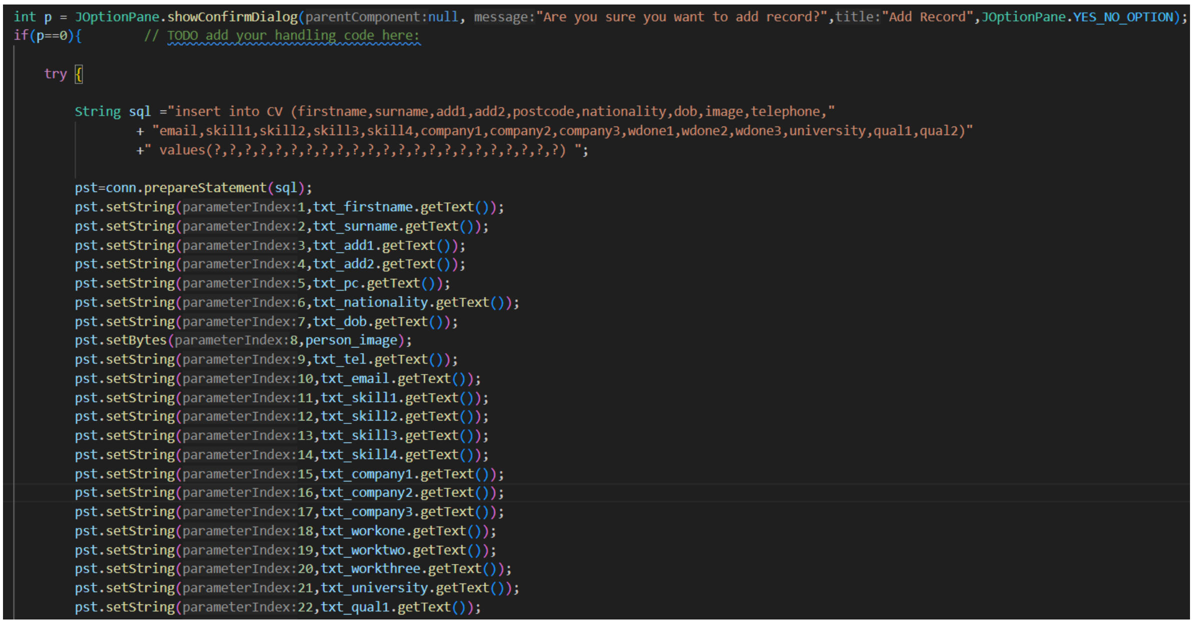


Figure 3.3 Relation Model

IV. SQL

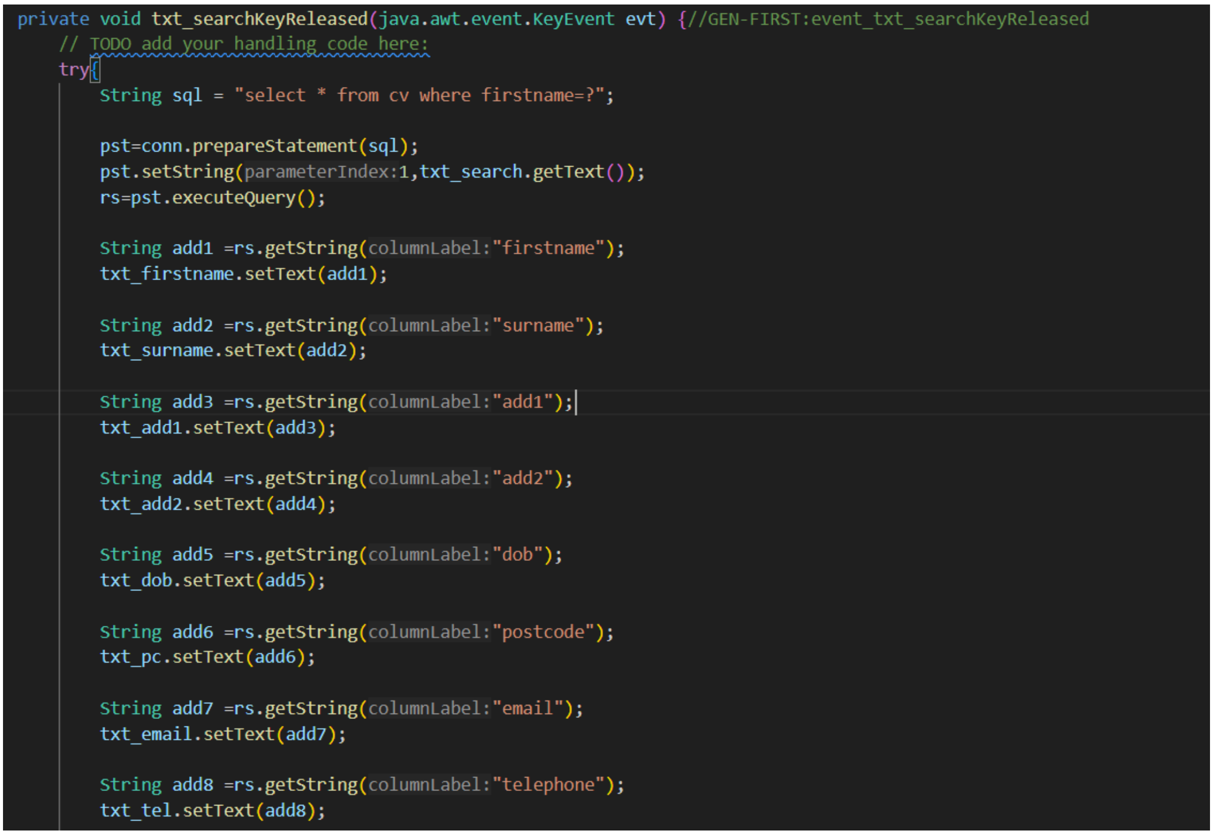
1. SQL Syntax in Application



```
int p = JOptionPane.showConfirmDialog(parentComponent:null, message:"Are you sure you want to add record?",title:"Add Record",JOptionPane.YES_NO_OPTION);
if(p==0){        // TODO add your handling code here:

    try {

        String sql ="insert into CV (firstname,surname,add1,add2,postcode,nationality,dob,image,telephone,"
                + "email,skill1,skill2,skill3,skill4,company1,company2,company3,wdone1,wdone2,wdone3,university,qual1,qual2)"
                +" values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) ";

        pst=conn.prepareStatement(sql);
        pst.setString(parameterIndex:1,txt_firstname.getText());
        pst.setString(parameterIndex:2,txt_surname.getText());
        pst.setString(parameterIndex:3,txt_add1.getText());
        pst.setString(parameterIndex:4,txt_add2.getText());
        pst.setString(parameterIndex:5,txt_pc.getText());
        pst.setString(parameterIndex:6,txt_nationality.getText());
        pst.setString(parameterIndex:7,txt_dob.getText());
        pst.setBytes(parameterIndex:8,person_image);
        pst.setString(parameterIndex:9,txt_tel.getText());
        pst.setString(parameterIndex:10,txt_email.getText());
        pst.setString(parameterIndex:11,txt_skill1.getText());
        pst.setString(parameterIndex:12,txt_skill2.getText());
        pst.setString(parameterIndex:13,txt_skill3.getText());
        pst.setString(parameterIndex:14,txt_skill4.getText());
        pst.setString(parameterIndex:15,txt_company1.getText());
        pst.setString(parameterIndex:16,txt_company2.getText());
        pst.setString(parameterIndex:17,txt_company3.getText());
        pst.setString(parameterIndex:18,txt_workone.getText());
        pst.setString(parameterIndex:19,txt_worktwo.getText());
        pst.setString(parameterIndex:20,txt_workthree.getText());
        pst.setString(parameterIndex:21,txt_university.getText());
        pst.setString(parameterIndex:22,txt_qual1.getText());
```

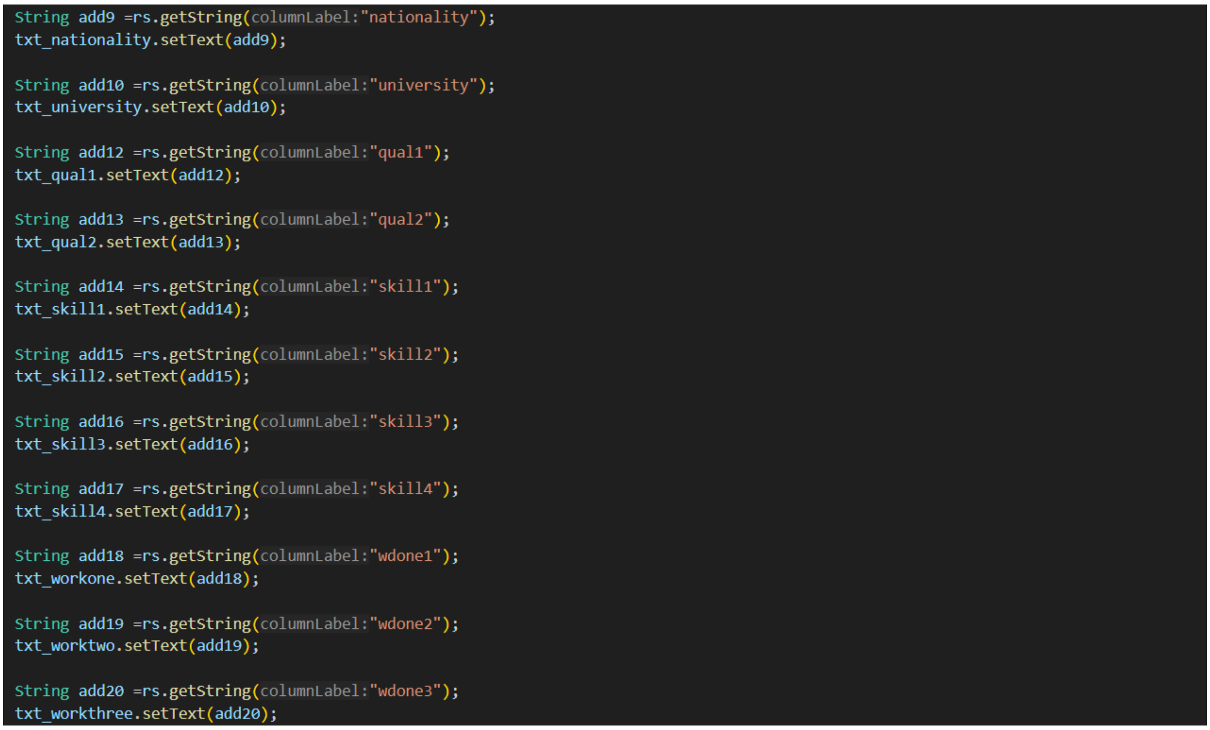
Figure 4.1 SQL Syntax in Application 1

The coding helps users save their information to the database. By using "INSERT INTO", the database receives the data corresponding to a certain user.



```
private void txt_searchKeyReleased(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_txt_searchKeyReleased
    // TODO add your handling code here:
    try{
        String sql = "select * from cv where firstname=?";

        pst=conn.prepareStatement(sql);
        pst.setString(parameterIndex:1,txt_search.getText());
        rs=pst.executeQuery();

        String add1 =rs.getString(columnLabel:"firstname");
        txt_firstname.setText(add1);

        String add2 =rs.getString(columnLabel:"surname");
        txt_surname.setText(add2);

        String add3 =rs.getString(columnLabel:"add1");
        txt_add1.setText(add3);

        String add4 =rs.getString(columnLabel:"add2");
        txt_add2.setText(add4);

        String add5 =rs.getString(columnLabel:"dob");
        txt_dob.setText(add5);

        String add6 =rs.getString(columnLabel:"postcode");
        txt_pc.setText(add6);

        String add7 =rs.getString(columnLabel:"email");
        txt_email.setText(add7);

        String add8 =rs.getString(columnLabel:"telephone");
        txt_tel.setText(add8);
```

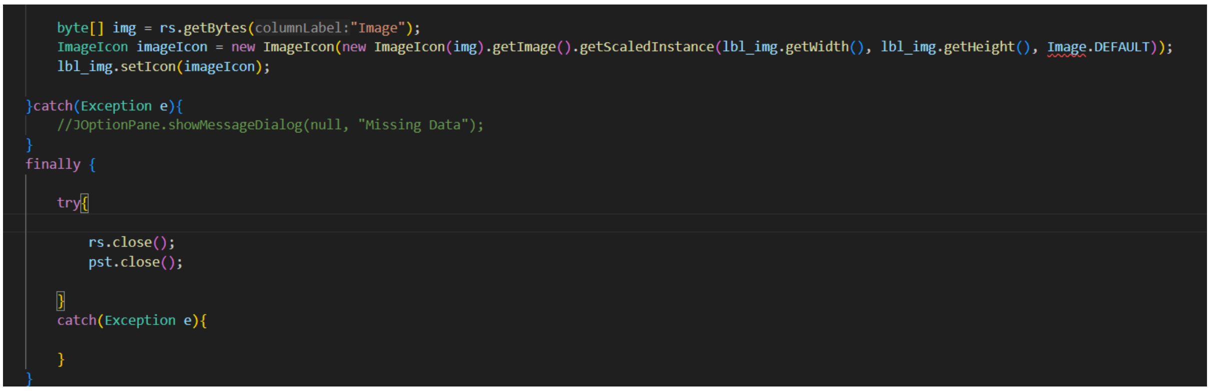
Figure 4.2 SQL Syntax in Application 2



```
String add9 =rs.getString(columnLabel:"nationality");
txt_nationality.setText(add9);

String add10 =rs.getString(columnLabel:"university");
txt_university.setText(add10);

String add12 =rs.getString(columnLabel:"qual1");
txt_qual1.setText(add12);

String add13 =rs.getString(columnLabel:"qual2");
txt_qual2.setText(add13);

String add14 =rs.getString(columnLabel:"skill1");
txt_skill1.setText(add14);

String add15 =rs.getString(columnLabel:"skill2");
txt_skill2.setText(add15);

String add16 =rs.getString(columnLabel:"skill3");
txt_skill3.setText(add16);

String add17 =rs.getString(columnLabel:"skill4");
txt_skill4.setText(add17);

String add18 =rs.getString(columnLabel:"wdone1");
txt_workone.setText(add18);

String add19 =rs.getString(columnLabel:"wdone2");
txt_worktwo.setText(add19);

String add20 =rs.getString(columnLabel:"wdone3");
txt_workthree.setText(add20);
```

Figure 4.3 SQL Syntax in Application 3

Both Figure 4.2 and Figure 4.3 showing the method displays detailed information about a person from the database when a name is typed into the “txt_search” text field**.** It updates the user interface (text fields) with values such as ‘firstname’, ‘surname’, ‘address’, ‘email’, etc., from the matched record in the ‘cv’ table

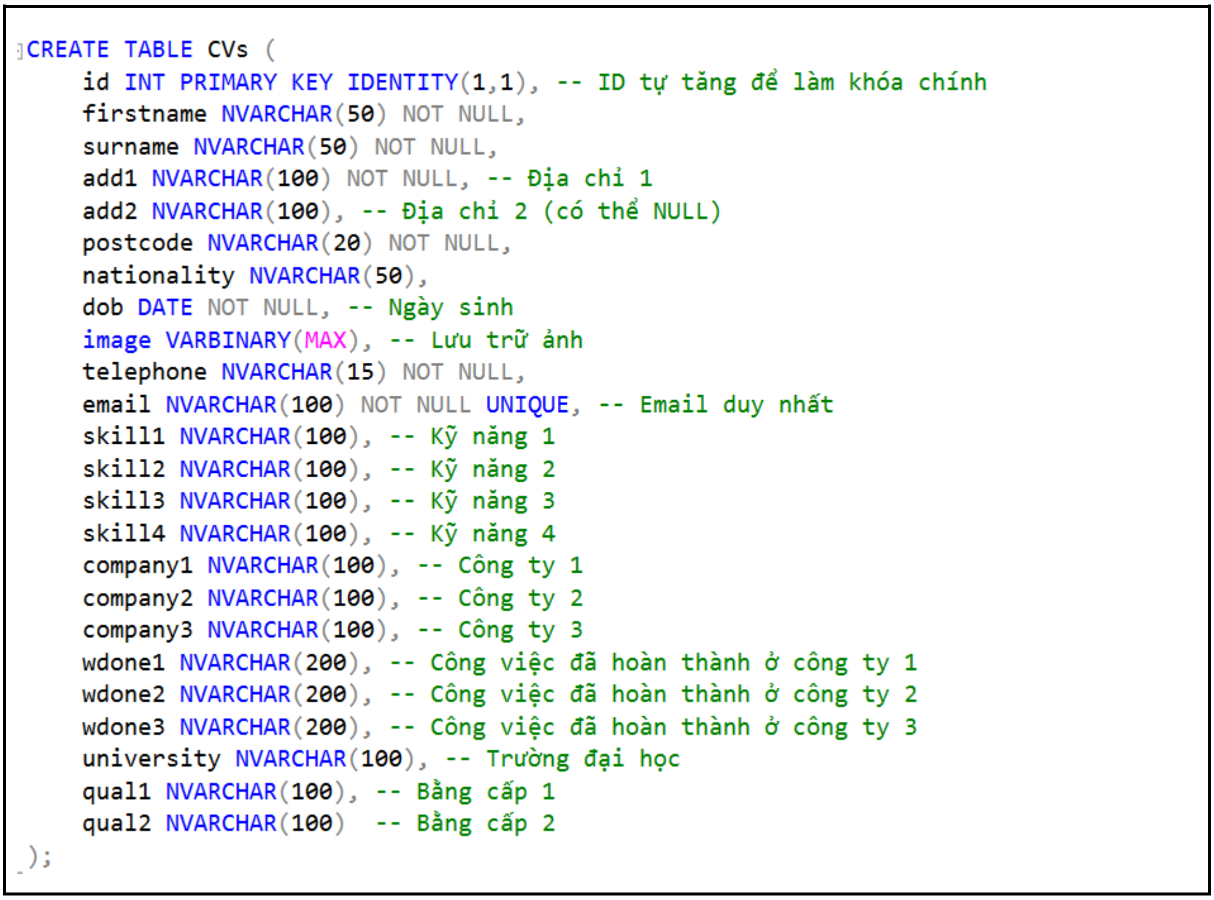


```
    byte[] img = rs.getBytes(columnLabel:"Image");
    ImageIcon imageIcon = new ImageIcon(new ImageIcon(img).getImage().getScaledInstance(lbl_img.getWidth(), lbl_img.getHeight(), Image.DEFAULT));
    lbl_img.setIcon(imageIcon);

}catch(Exception e){
    //JOptionPane.showMessageDialog(null, "Missing Data");
}
finally {

    try{

        rs.close();
        pst.close();

    }
    catch(Exception e){

    }
}
```

Figure 4.4 SQL Syntax in Application 4

This part of code is responsible for Retrieving an image from a database stored as a byte[]. Scaling it to fit dimensions of a jLabel. Displaying image in GUI.

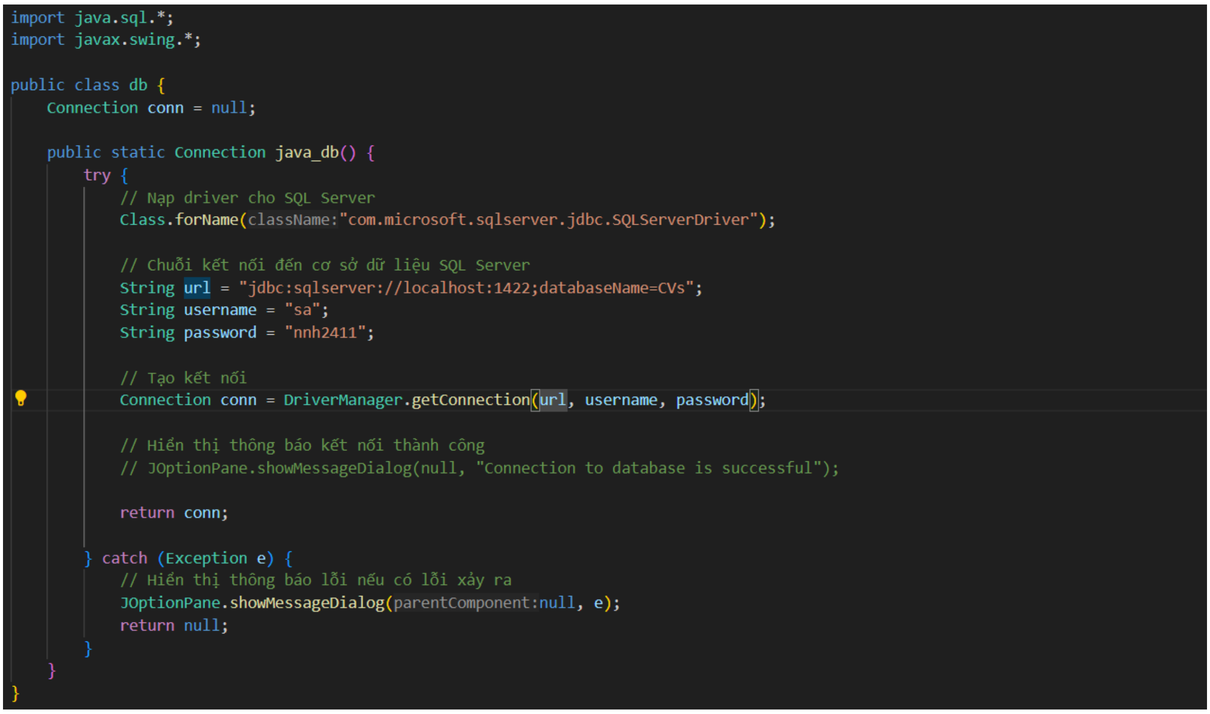
2. Query



```
CREATE TABLE CVs (
    id INT PRIMARY KEY IDENTITY(1,1), -- ID tự tăng để làm khóa chính
    firstname NVARCHAR(50) NOT NULL,
    surname NVARCHAR(50) NOT NULL,
    add1 NVARCHAR(100) NOT NULL, -- Địa chỉ 1
    add2 NVARCHAR(100), -- Địa chỉ 2 (có thể NULL)
    postcode NVARCHAR(20) NOT NULL,
    nationality NVARCHAR(50),
    dob DATE NOT NULL, -- Ngày sinh
    image VARBINARY(MAX), -- Lưu trữ ảnh
    telephone NVARCHAR(15) NOT NULL,
    email NVARCHAR(100) NOT NULL UNIQUE, -- Email duy nhất
    skill1 NVARCHAR(100), -- Kỹ năng 1
    skill2 NVARCHAR(100), -- Kỹ năng 2
    skill3 NVARCHAR(100), -- Kỹ năng 3
    skill4 NVARCHAR(100), -- Kỹ năng 4
    company1 NVARCHAR(100), -- Công ty 1
    company2 NVARCHAR(100), -- Công ty 2
    company3 NVARCHAR(100), -- Công ty 3
    wdone1 NVARCHAR(200), -- Công việc đã hoàn thành ở công ty 1
    wdone2 NVARCHAR(200), -- Công việc đã hoàn thành ở công ty 2
    wdone3 NVARCHAR(200), -- Công việc đã hoàn thành ở công ty 3
    university NVARCHAR(100), -- Trường đại học
    qual1 NVARCHAR(100), -- Bằng cấp 1
    qual2 NVARCHAR(100)  -- Bằng cấp 2
);
```

V. CONNECTION BETWEEN SQL AND JAVA

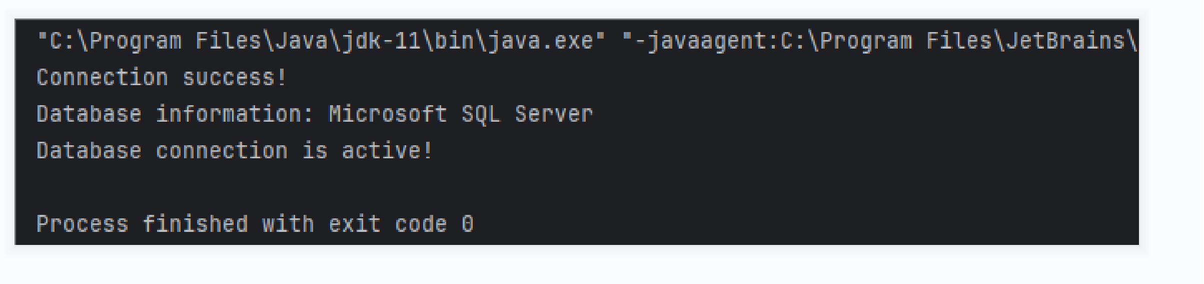
1. Syntax in Connection between SQL and Java



```java
import java.sql.*;
import javax.swing.*;

public class db {
    Connection conn = null;

    public static Connection java_db() {
        try {
            // Nạp driver cho SQL Server
            Class.forName(className:"com.microsoft.sqlserver.jdbc.SQLServerDriver");

            // Chuỗi kết nối đến cơ sở dữ liệu SQL Server
            String url = "jdbc:sqlserver://localhost:1422;databaseName=CVs";
            String username = "sa";
            String password = "nnh2411";

            // Tạo kết nối
            Connection conn = DriverManager.getConnection(url, username, password);

            // Hiển thị thông báo kết nối thành công
            // JOptionPane.showMessageDialog(null, "Connection to database is successful");

            return conn;

        } catch (Exception e) {
            // Hiển thị thông báo lỗi nếu có lỗi xảy ra
            JOptionPane.showMessageDialog(parentComponent:null, e);
            return null;
        }
    }
}
```

Figure 5.1 Syntax in Connection between SQL and Java 1

Connect to SQL by loading the drive. After creating the connection, there is a pre-programmed command to signal that the connection was successful.



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\
Connection success!
Database information: Microsoft SQL Server
Database connection is active!

Process finished with exit code 0
```

Figure 5.2 Syntax in Connection between SQL and Java 2

The notification that the connection was successful

2. Application forms

In this section, we will show the application

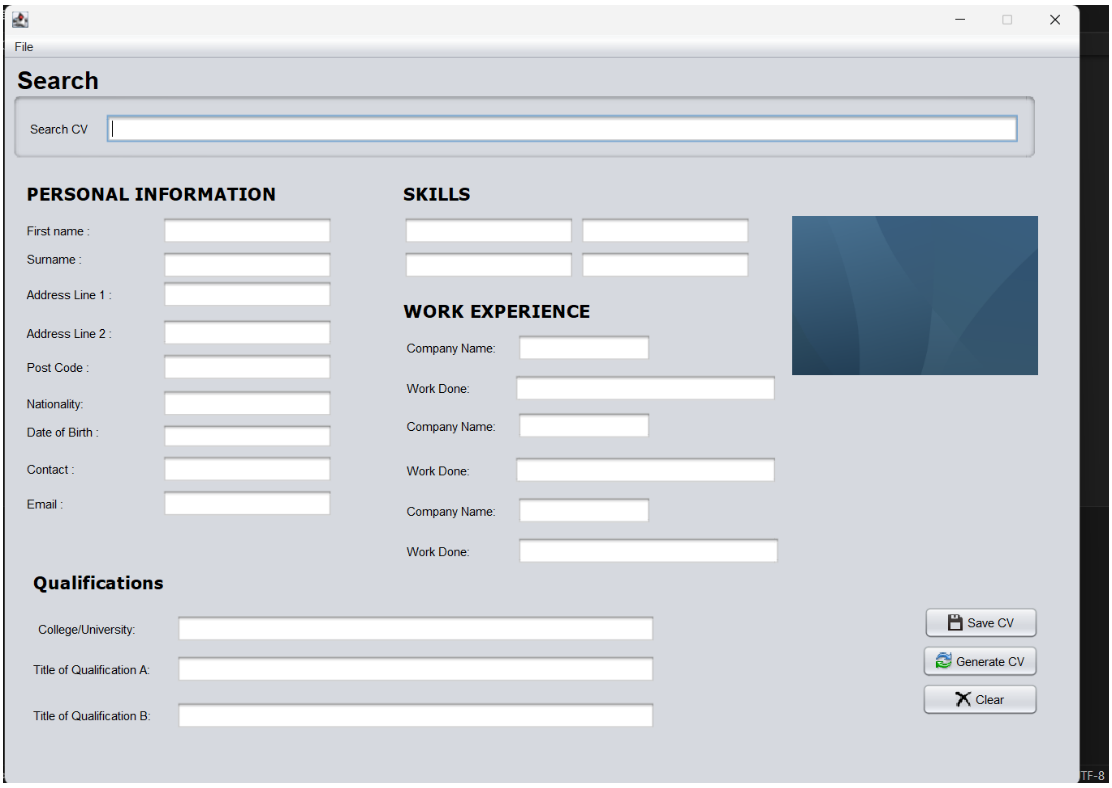
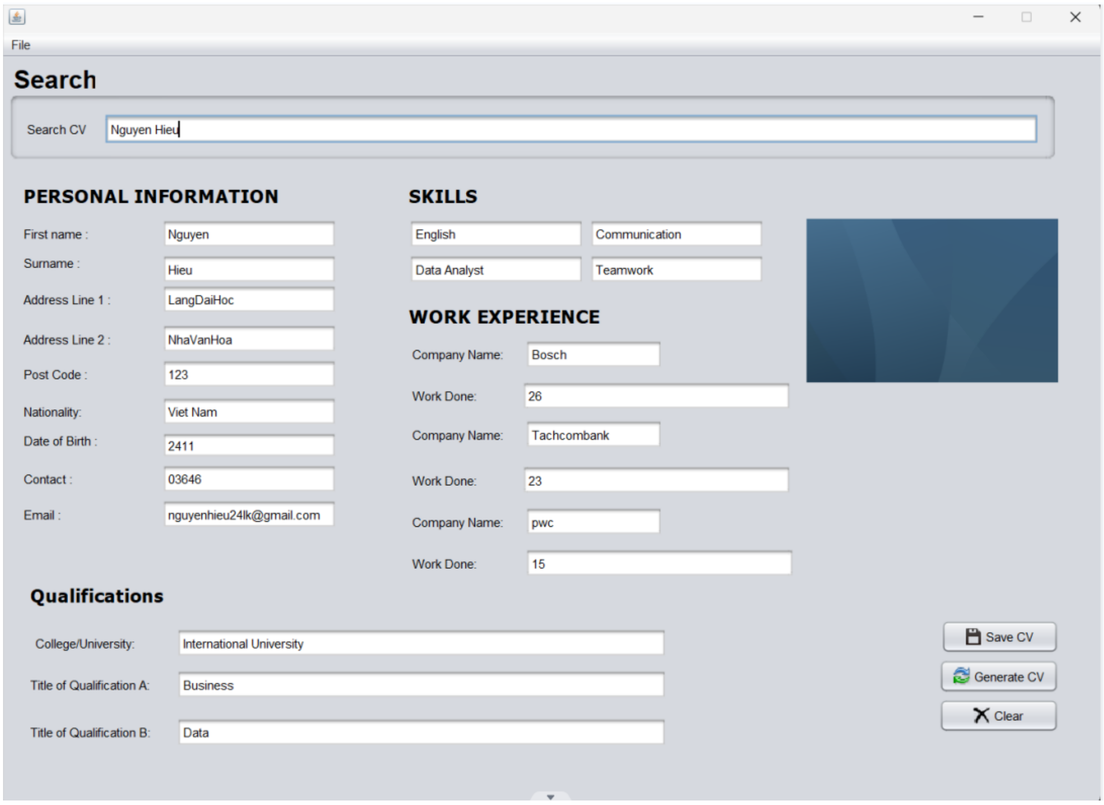


Figure 5.3 Add Information Frame



Search

Search CV: Nguyen Hieu

PERSONAL INFORMATION

First name :	Nguyen
Surname :	Hieu
Address Line 1 :	LangDaiHoc
Address Line 2 :	NhaVanHoa
Post Code :	123
Nationality:	Viet Nam
Date of Birth :	2411
Contact :	03646
Email :	nguyenhieu24lk@gmail.com

SKILLS

English	Communication
Data Analyst	Teamwork

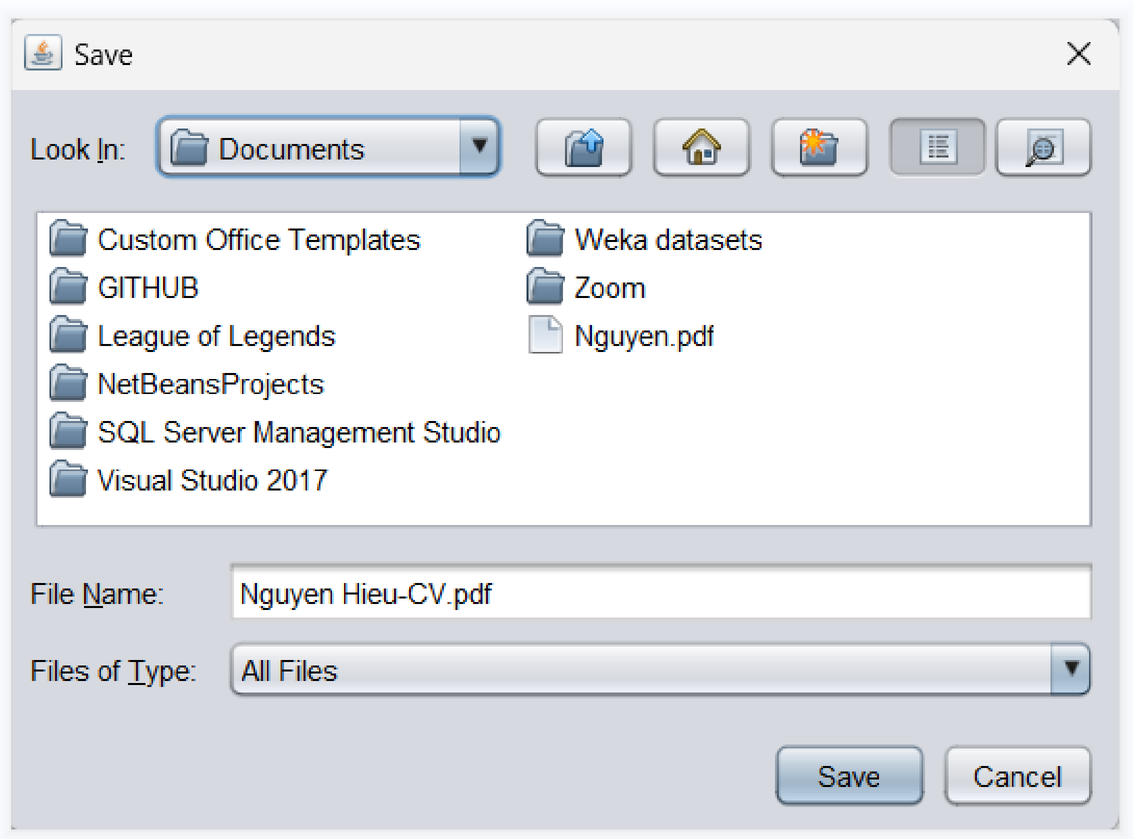
WORK EXPERIENCE

Company Name:	Bosch
Work Done:	26
Company Name:	Tachcombank
Work Done:	23
Company Name:	pwc
Work Done:	15

Qualifications

College/University:	International University
Title of Qualification A:	Business
Title of Qualification B:	Data

Figure 5.4 Added Information



Save

Look In: Documents

Custom Office Templates
GITHUB
League of Legends
NetBeansProjects
SQL Server Management Studio
Visual Studio 2017
Weka datasets
Zoom
Nguyen.pdf

File Name: Nguyen Hieu-CV.pdf

Files of Type: All Files

Figure 5.5 Interface of Saving CV

VI. APPLICATION

1. Used tools and application

- Programming Language

  The primary programming language used in this project is Java. It has many powerful standard libraries that support a wide range of functionalities.

  The version of JDK is 11 or greater.

- Development Environment

  The project was developed using the Integrated Development Environment (IDE) Intellij. Intellij offers features like code completion, debugging, and refactoring that enhance productivity and code quality.

- GUI Library

  The graphical user interface (GUI) of the game was created using JavaFX library. JavaFX provides a set of 'lightweight' (all-Java language) components that, to the maximum degree possible, work the same on all platforms.

- Database

  We use SQL server to store data in the system. SQL Server is a relational database management system (RDBMS) that offers a wide range of features for managing and querying data effectively developed by Microsoft.

- Version Control

  We used Git for version control, with a shared repository on GitHub. This allowed the team to collaborate effectively, track changes, and maintain different versions of the project.

- Tools
  - Github/Github Git for overseeing versions and importing data.
  - JDBC Driver for SQL Server provides database connectivity through the standard JDBC application program interfaces (APIs) available on the Java platform.
  - Java Development Kit is a bundle of software development tools and supporting libraries combined with the Java Runtime Environment (JRE) and Java Virtual Machine (JVM).

- Microsoft SQL Server is a relational database management system (RDBMS)

2. Framework

- Graphic User Interface (GUI) Framework:
    - The Java AWT (Abstract Window Toolkit) package offers classes for basic graphic interactions and the creation and administration of graphical user interfaces (GUIs) in Java. It has GUI building pieces like buttons, text fields, and labels, as well as containers like Frame, Panel, and Dialog to house these components. These elements are arranged inside containers with the use of layout managers like BorderLayout, FlowLayout, and GridLayout. Additionally, AWT manages event processing, which allows programs to react to user input like as keystrokes and mouse events. Because of this, AWT is a fundamental tool for creating interactive apps.
    - A library called Java Swing enhances AWT and offers more complex GUI elements like ‘JButton’, ‘JFrame’, ‘JPanel’, and others. Compared to AWT, Swing provides a more robust and adaptable collection of components. enabling a more advanced and aesthetically pleasing user interface. Other characteristics include lightweight components, pluggable look-and-feel, and sophisticated controls like trees and tables. Swing components offer platform independence and a uniform user experience across several operating systems since they are fully built in Java. Because of this, Swing is a popular option for creating intricate and dynamic Java applications.
- SQL Framework:
    - Classes and interfaces for using JDBC (Java Database Connectivity) to deal with databases are provided by the ‘java.sql’ package. This package makes it possible for Java apps to run SQL queries, retrieve data, and connect to a large number of databases. ‘Connection’, ‘Statement’, ‘PreparedStatement’, and ‘ResultSet’ are essential elements that make it easier to create connections, run queries, and process data. ‘java.sql’ is a crucial component for enterprise-level software that needs dependable and effective database access and manipulation as it abstracts database interactions, enabling developers to create database-agnostic applications.

VII. CONCLUSION

1. Future work

- Achieving these goals opens up exciting possibilities for the future development of the project into a complete real-world application. Our team is committed to enhancing the user experience by implementing more advanced features.
- Online CVs builder saves time, allows for easy customization, and shows the material professionally. With the various templates, you can rapidly construct a beautiful CV that is appropriate for any position. At the same time, an online CV provides for flexible storage and distribution in digital form, facilitates speedy communication with employers, and raises the likelihood of a successful application.

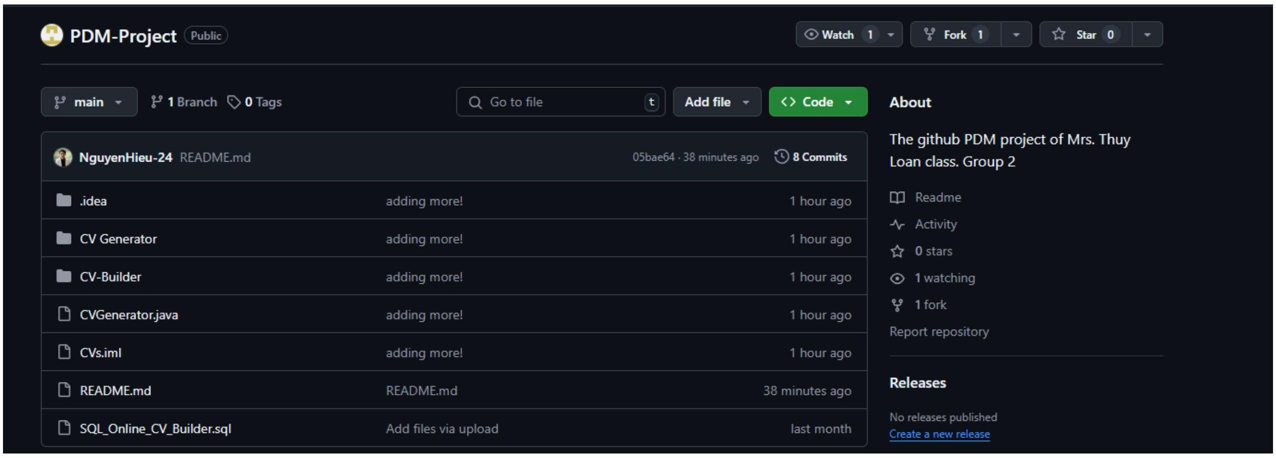
2. Summary

In conclusion, the project so far has successfully completed the core database design, meeting the initial objectives set out for mid-term phase. Through careful database modeling and normalization up to the BCNF, the project has achieved a robust, efficient data structure that supports customizable CV sections. This design ensures efficient data storage and retrieval while establishing the groundwork for secure data handling and user authentication in future stages.

Moving forward, the focus will be on integrating this database with other components of the platform, allowing for additional features and functionalities to be developed. These enhancements and final integrations will be detailed in the final report, showcasing the database’s pivotal role in the platform’s overall functionality.

VIII. PROJECT'S GITHUB

Link to our GitHub repository: https://github.com/HuyPham144/PDM-Project



NguyenHieu-24 README.md 05bae64 · 38 minutes ago 8 Commits

.idea	adding more!	1 hour ago
CV Generator	adding more!	1 hour ago
CV-Builder	adding more!	1 hour ago
CVGenerator.java	adding more!	1 hour ago
CVs.iml	adding more!	1 hour ago
README.md	README.md	38 minutes ago
SQL_Online_CV_Builder.sql	Add files via upload	last month

About

The github PDM project of Mrs. Thuy Loan class. Group 2

Readme
Activity
0 stars
1 watching
1 fork
Report repository

Releases

No releases published
Create a new release

IX. REFERENCES

1. Ramakrishnan, R., & Gehrke, J. (2002). *Database Management Systems (3rd ed.)*. McGraw-Hill Education.
2. Schildt, H. (2018). *Java: The Complete Reference (11th ed.)*. McGraw-Hill Education.
3. Article, (2024, September 26). Download SQL Server Management Studio (SSMS). *Microsoft*. Microsoft – AI, Đám mây, Năng suất, Điện toán, Trò chơi & Ứng dụng
4. Normalization and Database Design - W3Schools. (n.d.). *W3school*. https://www.w3schools.com/sql/sql_normalization.asp
5. (2020). Case Study: Online Resume Builder - A Study on Modern CV Management. *Journal of Database Applications and Information Systems*, 15(2), 134-145.
6. *SQL Server Database Programming with Java: Concepts, Designs and Implementations | SpringerLink*
7. *Java Tutorial - An Introduction to Java Database Programming (JDBC) by Examples with MySQL (ntu.edu.sg)*