EUROPEAN UNIVERSITY OF LEFKE

DEPARTMENT OF Software engineering

Semester Project

COMP 337

DATABASE MANAGEMENT SYSTEMS

TOPIC TITLE

Claudette Umutoni(22243560)
Bruno Shema (22243402)

Submitted to

Mr Salman Khan

LEFKE 2024

Grading Sheet

Student's Name			
Student's Number			
Grading Criteria	**Marks**	**Team Member 1**	**Team Member 2**
1. Functionality	10		
2. ER Diagrams	10		
3. Database Working	05		
4. GUI Interface	03		
5. Report and Format	02		
6. Viva	10		
Total	40		

Comment's (if Required):

Teacher Signature: _____

Date: _____

Semester project REPORT:GUI_ Based Desktop Application

Table of content:

Table figures:

Figure 8:employer GUI

Abbreviations:

DBMS: DataBase Management System

SQL: Structured Query Language

GUI: Graphical User Interface

Chapter 1. Project Overview

1.1 Problem of Interest

Project description

We decided to create a desktop application that links potential employers and employees.We discovered that there are many competent people out here , who are now unemployed; this is sometimes even cause by limited area of search, employers employing people nearby only, as they are the one available; but the good thing is that our application will link them, and the service will be even supplied across borders.

We value people's money and time,this is why we dedicated this project to creating a prestigious job for deserving worker. This noble business, as reflected in the name " **Career Castle** ", we emphasize a noble connection between employer and employee.

Objectives and Goals

- Allow people to apply for jobs ,from better companies even across borders
- Allow employer to easily receive applications, and get back to the employees who meet there requirements much more easily.
- People should be able to see trending jobs and which company is offering those jobs.

1.2 Team Details

Team Members

NAME	ROLE
BRUNO SHEMA :	Front_end development
Claudette Umutoni :	Data Base design

1.3 Job Descriptions

NAME **Task description**

BRUNO SHEMA :

- Focused on developing the GUI interface
- Integrate my teammate's code into the front_end codes
- Handle the flow of the application
- Test the overall work, recursively

Claudette Umutoni :

- Design the relation tables, and optimize the schema
- Develop a back_end compatible with our project theme
- Test the overall work, recursively

Day No	Team member 1 Task	Team member 2 Task	Status
Day 1	create connection to Sqlite DBMS	check connection to Sqlite DBMS	Complete
Day 2	Develop the login page Provide multiple access to the program, according to back_end architecture.	Properly design the hierarchy of the program users	Complete
Day 3	Develop employee tab	Develop database related accesses,	Complete

		round 1	
Day 4	Develop employer tab	Develop database related accesses,round 2	Complete
Day 4	Combine login page, employee, and employer page, with the database functionalities	Start providing the documentation on ER_diagram	Half_Complete
Day 5	Produce the overall report for our project	Assist in whatever I was not very sure of	Complete

Chapter 2. Programming Language Choice

Due to very limited and short period of time provided on the project, we tried to go for an easy programming language; that is why we chose **Python.** In python we used **Customtkinter** as a dependency for our project, it is a library used to make python application.We used custom version of tkinter, out of curiosity, as the normal tkinter is provide by python package; we felt triggered to use new version which is manually installed, also hoping to having better graphical displays; which was the case.

This Customtkinter, is super easy to get along with it, as we were also learning to use Python, going the easy way was more of a must; another plus on this tkinter version, is that it is highly supported by python, and the best part is that , that very **Pthyon** we used is super compatible with **SQLite.**That is why we used, Pthon, Customtkinter, and SQLite.

Chapter 3. Database and Relations

3.1 Database Choice

We used **SQLite** it was the easiest, and it is the very database , a front_end developer was used. We chose to go with the front_end developer’s working requirements, as we

need what is flexible for him, so we won't spend time debugging things that we don't rely know how they work.

FEATURES:

- It is compatible with python, hence being easy to connect.

- it graphical used interface has auto-fill which helps to explore and know if the sql code is write or not, even before running it.

- most importantly as to make changes to the database, it requires to close the data base, this database is the same, but if you save the database with some written sql code on it, you see that fill in vs code, means you can refer to the database when it is closed.

3.2 Database Relations

Relational tables:

- Employee

- Employer

- Job_application

- Job_posting

- job

- job_offer

ER-DIAGRAM

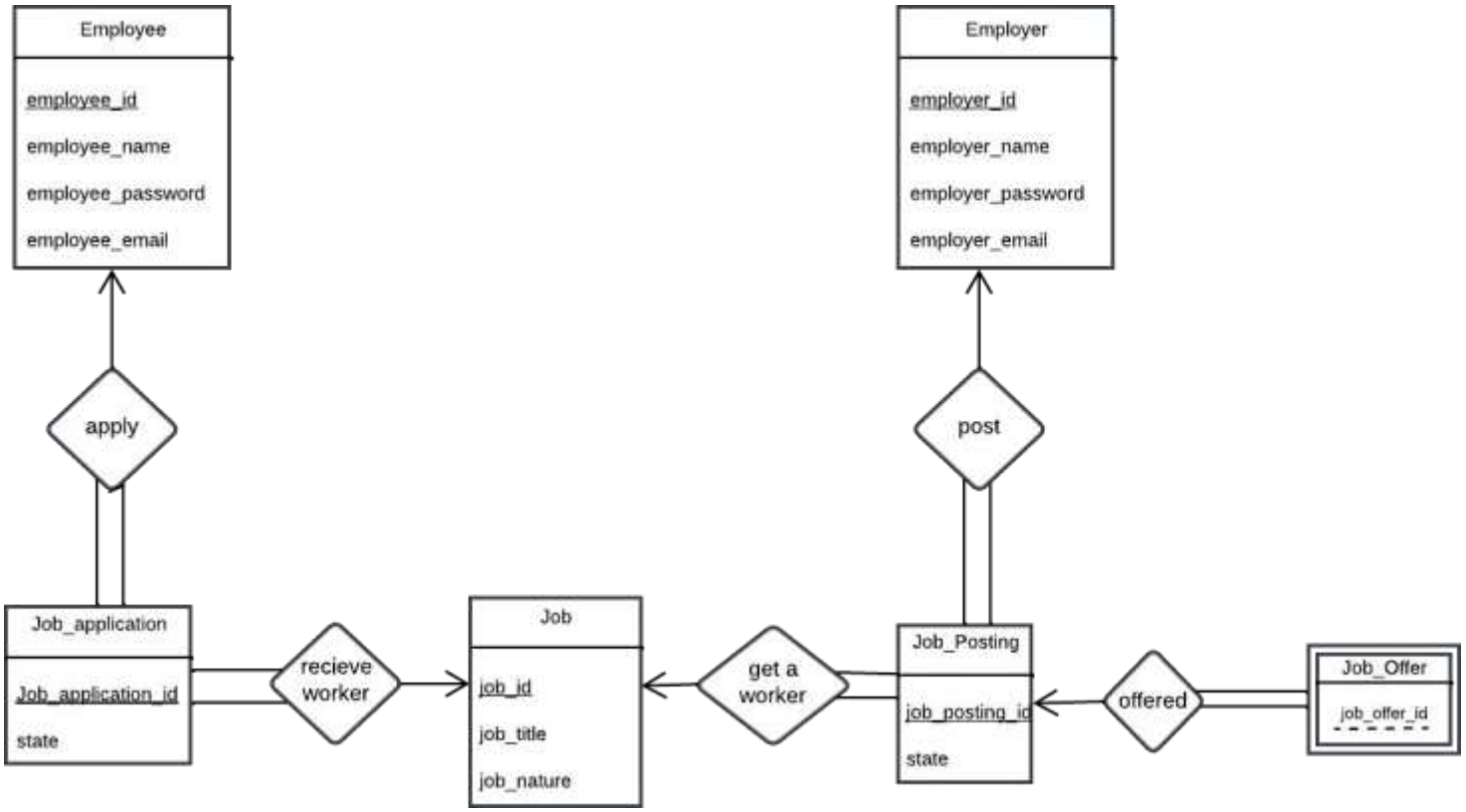

Figure 1: ER DIAGRAM

Relational schema:

Employee(employee_id,employee_name,employee_password,employee_email)

Employer(employer_id,employer_name,employer_password,employer_email)

Job(job_id, job_title,job_nature)

Job_posting(job_posting_id,employer_id,Job_id,state)

Job_application(job_application_id,employee_id,job_posting_id,education_level,experience)

Job_offer(job_offer_id , job_posting_id)

Data Flow Diagram

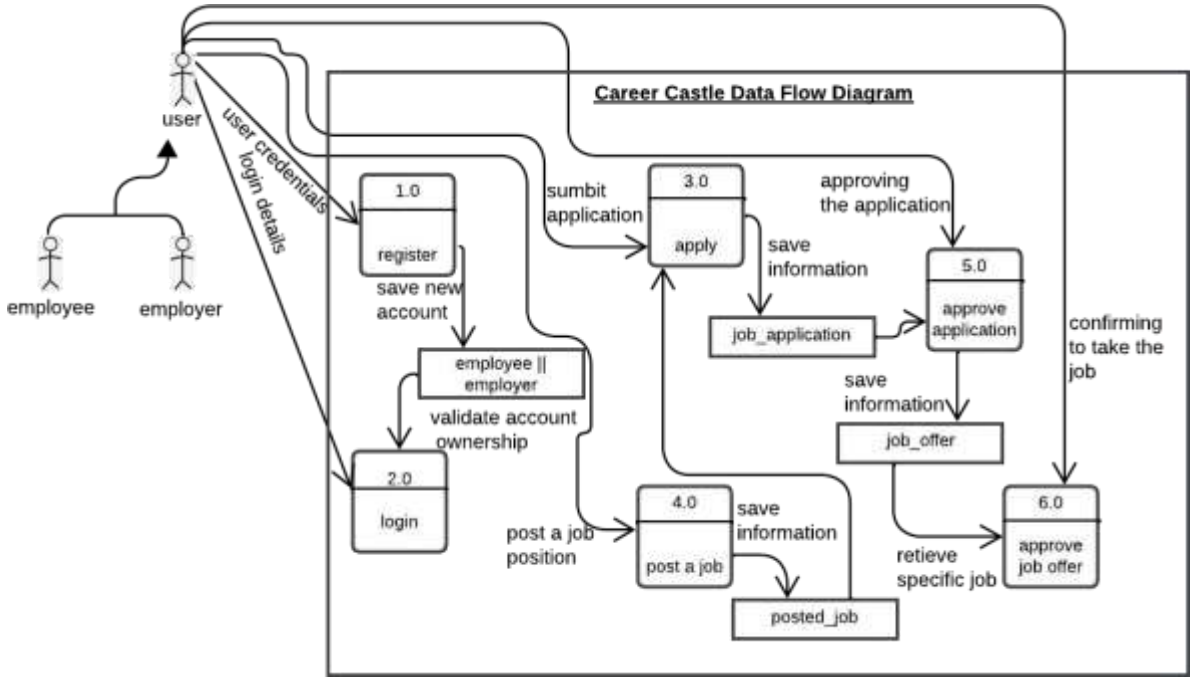

Figure 2: Data Flow Diagram

Chapter 4. Functionality

Functionality:

Mainly this desktop application, has two GUI, one for the employee and another for employer; the naming is a little bit off, but we took it in terms of advertising, calling an applicant an employee, it build that positive spirit of getting a job, and seeing all companies as employers it is more comforting.

For employee;

He is prompted to a login page, where by if he/she is a new user , a registration form is provided, relevant to the kind of user; and after ward logs in.

Up on access his/her platform, there is all his/her credentials, that were saved in the database as he registered, there is a button , to change those credential,

In the main frame, there is an advertisement above, attracting the user for companies's products, bellow, there displays all posted jobs, and each with option to apply for it.The application is always almost done, all needed is the desired position.whenever any company accepts him or her, there is frame which displays that, with a custom message, for the employee, for further communication.

More over a user has option to delete his or her account

For employer:

The scenario, is nearly the same; one searches in the drop down-list the company and only provide the password; if the list is not there, it is better to first register.after registering one can access the account;.

With in the GUI there is a log image, description of their products, available jobs,. in the main frame too; there is an advertisement section, beneath having, list of applications,the employer, has option to post a job, as well as delete the account.

Chapter 5. Screenshots and Demonstration

Starting with loging in: we have an employee and employer login form

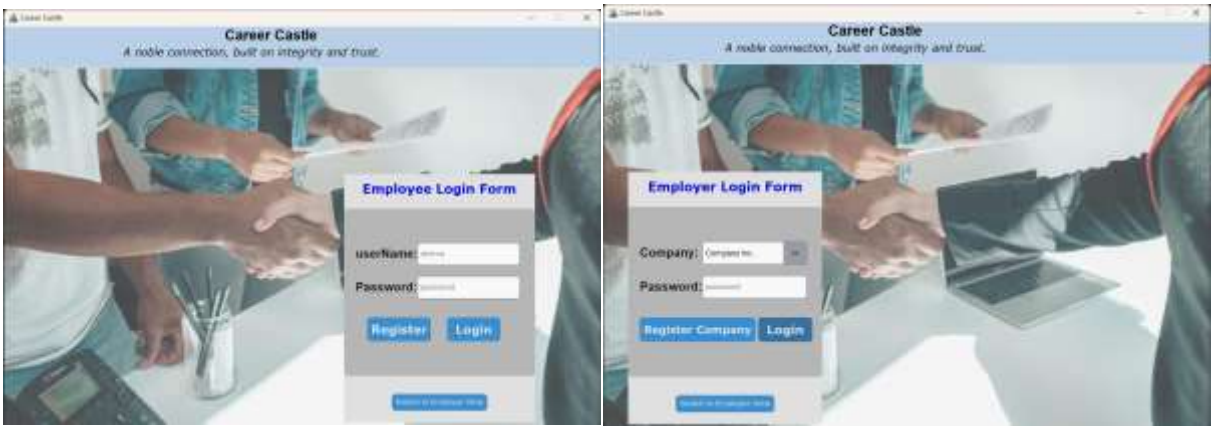

Figure 3: login for employee and employer

Each with its own window prompted when registering; collected values are all captelized; due to shortage of time; we prefered to short it in the terminal.

Figure 4: registration window for employee

```
Collected Values: ['BRUNO', '123', 'BRUNO@GMAIL.COM', 'TURKEY']
```

Here is the registration window for employer;

Figure 5: registration for employer

This is how login look like; for employee , you provide name and password, but for employer, you search you name in the drop down list, and provide a relevant password

Figure 6: login for both GUI

Here is the GUI for the employee; in here you can modify the user credentials

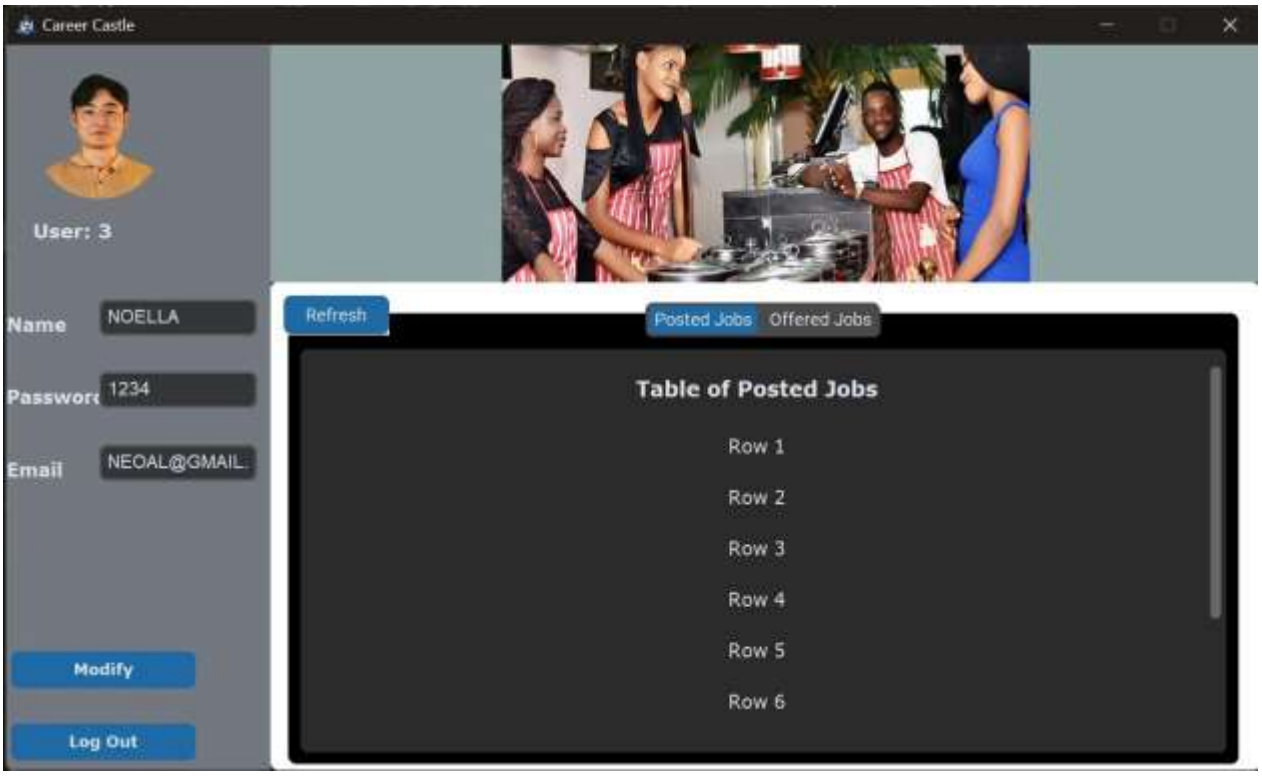

Figure 7: employee gui

This one is for the employer

Figure 8:employer GUI

Chapter 6. Conclusion

In brief this project we, can say that it is a success at a certain level, we did learn and experience how real world programming feel like; more over we saw how crucial the time management, is. For both back_end and front_end, we saw multiple limitation, explaining why this typical python is not used on large scale basis.

First of all, even though it is simple, it require longer code lines, and most of the features are not well supported, for example image rendering is not well or easily done, it required us to use PIL , which is manually installed.switching from one file to another is also a challenge, even using view tabs will somehow crash. The only tactic is to use processes.

But apart from these cons, the pros to this python, documentations are allover the internet; you just have to imagine, and google a bit, the codes are somewhere on the internet.

Chapter 7. Conclusion

1. Documentation on customtkinter:

https://github.com/TomSchimansky/CustomTkinter

2. Programmer with MOsh:

https://www.youtube.com/watch?v=yVl_G-F7m8c

3. Documentation on tkinter:

https://realpython.com/python-gui-tkinter/