**CHAPTER 1**

**INTRODUCTION**

**Problem Statement**

Hundreds of students are recruited by an assortment of companies. Build an application that can store information pertaining students’ recruitment status such as number of applications submitted, interview rounds cleared in total, number of offer letters received and the package amount, if any, number of students selected by a certain company and students’ success rate.

**Objectives**

Placement tracker and analysis acts as a portal application for Teachers and students to check on students’ progress and provide valuable insights.

The frontend is built with Tkinter, while the backend is connected to SQLite database. It would contain tables with recruitment information. The user can fetch details of a particular student by typing his/her name in the input field in the GUI. Operations that can be carried over by teachers module of the application:

1. Add value rows to the table
2. Delete information
3. Update information such as name and other fields that are allowed to be edited due to security concerns
4. Display information according to what the user wants to see. Such as:

* Number of applications submitted by a student,
* Interview rounds cleared,
* Number of offer letter received, package amount if any,
* Students selected by a certain recruiter/company.

**Analysis**: representation of recruitment trends followed by companies in the last 3 years.

Operations that can be carried over by students module of the application:

1. View the number of applications submitted by him/her,
2. View number of rounds cleared,
3. Number of offer letter received, package amount if any,

A student cannot update his/her information. This is controlled by the teachers to ensure 0 discrepancies and fairness.

**Analysis**: representation of recruitment trends followed by companies in the last 3 years.

# Requirement Specifications

HARDWARE REQUIREMENTS:

Processor: Intel(R) Core(TM) i3-10110U CPU Processor Speed: 2.59 GHz

RAM: 8 GB RAM

System type: 64 –bit operating system Hard disk : 512 GB

SOFTWARE REQUIREMENTS:

Language: Python and Database Compiler: Any python compiler

**CHAPTER 2**

**FUNDAMENTALS OF PYTHON**

**2.1 INTRODUCTION TO PYTHON**

[Python](https://www.geeksforgeeks.org/python-programming-language/) is a commonly and extensively used general-purpose, high-level programming language. Guido van Rossum in 1991 was the founder of Python and was later developed by Python Software Foundation. It was primarily designed to emphasize on code readability, and its syntax allows programmers to express ideas in few lines of code. Python can be used for things like:

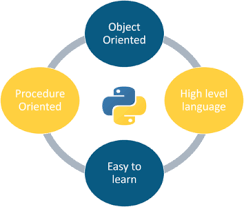
****

Figure 2.1: Different versions of Python over the years

**2.1.1 ENTITY-RELATIONSHIP (ER) MODEL**

The Entity-Relationship (ER) Model is an attractive high level conceptual data model. It has an entity which may be an object with a physical existence like a particular car, house, person or employee or it may be an object with a conceptual existence like an organization, a profession, or a university course. Each entity has attributes—the definite properties that characterize it. For example, a student entity may be described by the student’s name, age, address, USN etc.

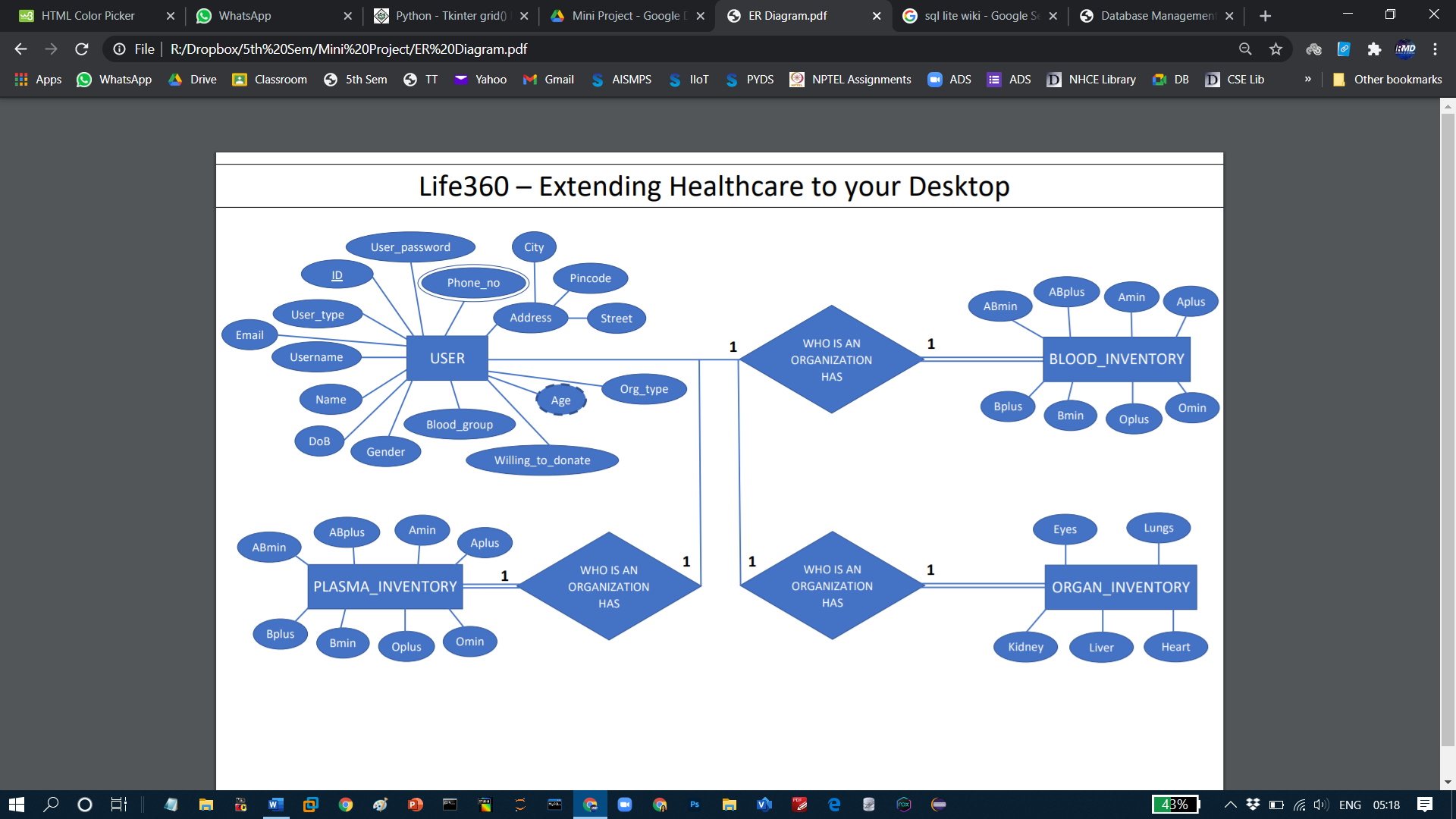


Figure 2.2: DBMS Component Modules

**2.1.1.1 ENTITY-**

**RELATIONSHIP (ER) MODEL**

The Entity-Relationship (ER) Model is an attractive high level conceptual data model. It has an entity which may be an object with a physical existence like a particular car, house, person or employee or it may be an object with a conceptual existence like an organization, a profession, or a university course. Each entity has attributes—the definite properties that characterize it. For example, a student entity may be described by the student’s name, age, address, USN etc.

**CHAPTER 4**

**FUNDAMENTALS OF DBMS**

**4.1 DESIGN GOALS**

This mini project has ensured that the user has an interactive and explorable environment. The interface is user friendly, simple to understand and has tried to ensure that there are no bugs.

Table 4.1: Various widgets available in Tkinter

|  |  |
| --- | --- |
| WIDGETS | DESCRIPTION |
| Label | This widget is used to display text or image on the window/frame |
| Button | This widget is used to add buttons to the user interface |
| Canvas | This widget allows one to draw pictures and different types of layouts like texts, graphics etc. |
| Entry | This widget is used to take as input, a single line text entry from user |
| Frame | This widget is used as box or container. It holds and organizes the widgets in an orderly fashion |
| SpinBox | This widget allows users to select from a given number of values |
| ComboBox | This widget contains a down arrow to select from a list of options |
| CheckButton | This widget displays a number toggle buttons which represent various options from which user can select any number of options. |
| RadioButton | This widget is similar to the CheckButton but allows only one option to be selected |
| Scale | This widget is used to provide a slider which allows the user to select any value from the scale |

Various triggers were also used to ensure referential integrity and data integrity:

* Age\_Calc: This trigger calculates the age from the attribute ‘date\_of\_birth’ when a new tuple is inserted into the table.

|  |
| --- |
| CREATE TRIGGER age\_calc  AFTER INSERT ON users  FOR EACH ROW  BEGIN  UPDATE users  SET age = CAST (strftime('%Y.%m%d', 'now') - strftime('%Y.%m%d', new.dob) AS INT)  WHERE ID = new.ID AND new.usertype = "Individual";  END; |

**CHAPTER 4**

**PARTIAL IMPLEMENTATION DESIGN**

This code contains two tables: **student details** and **recruitment details**.

* Fields in student\_details table: student name, USN, phone number, department and semester.
* Fields in recruitment\_details table: student USN, company ID, company name eligibility of student to sit for the rounds, round 1 status, round 2 status, round 3 status, round 4 status, round 5 status, and the negotiated salary (if selected).
* Data can be inserted in both of said tables. The data added can be seen in **DB browser for SQLite** app.
* Menu is built. User can choose to add student details, recruitment details, or view result of predefined queries like:

1- Companies student is selected by

2 - Companies student was rejected by

3 - Companies student applied to

4 – List of companies

* And more queries.

**CHAPTER 8**

**RESULTS**

# 

# Fig 8.1 Menu of the program

# 

# Fig 8.2 Entering student details

# Once student details are added, it will be fed into the student\_details table in the database.db file. After insertion, the menu will reappear.

# 

# Fig 8.3 Data entered previously, is added to the database student\_details

# 

# Fig 8.4 Recruitment details being entered

# 

# Fig 8.5 recruitment\_details database with new value