MANIPAL INSTITUTE OF TECHNOLOGY

Manipal Academy of Higher Education
MANIPAL – 576104, KARNATAKA, INDIA

Advance Programming Lab Mini-Project: Project Report

Submitted by:

Name: Ankit Mishra Reg no: 200953214 Roll no: 53	Name: Shwetabh Saxena Reg no: 200953262 Roll no: 64	Name: Soham Sonaiya Reg no: 200953172 Roll no: 45

INDEX

Abstract

In today's world, everyone deals with data and data related tasks as every organisation needs to manage its data. However, not everyone who interacts with data is trained to use it in efficient way and perform all the tasks that are required in industry. It is possible that a user may need to change certain data tuples of a database or perform any operation over certain tuples of the database and they might not know how to do it. It is also possible that not everyone has a GUI access with administrator rights to append, change or delete the data in the database.

Therefore, there comes a need of a software to deal this and helps people with this very issue. The workings of Excel Workbook make it easier for user to add, update and delete data in the database. The GUI helps user to choose file and choose the operation to be performed over it and the user works in excel and that changes are done and saved in database.

Background Information

Our app takes input file name and allows the user to make changes in that file using excel and it automatically makes those changes in database without any SQL queries.

We saw the need for such app as there are many people who do not have enough knowledge or education to use such languages and they are hard to learn and use in daily life. So, this app makes their work easier by simplifying this process using the GUI and preventing them from using these languages.

We have therefore created three functionalities which the user has to choose from which are create database from excel file, sync database to excel file and sync excel file to database. Thew first functionality creates a database for the selected excel file. The other two allows the user to sync the changes done in either excel file or database.

Methodology

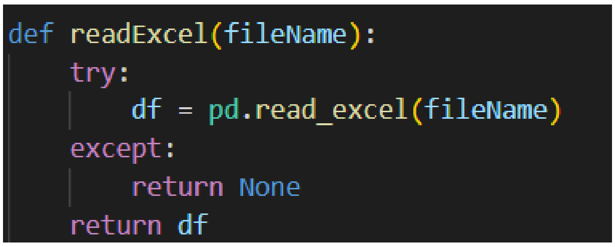
The Following Technologies/Modules has been used to make the D2X application are as follows:

Modules:

a) Tkinter: Used to create GUI for the software.
b) Sqlite3: Used to create, update and access the database.
c) Pandas: Used to read ,write and export the excel file. Some more functionalities were used regarding data frames.

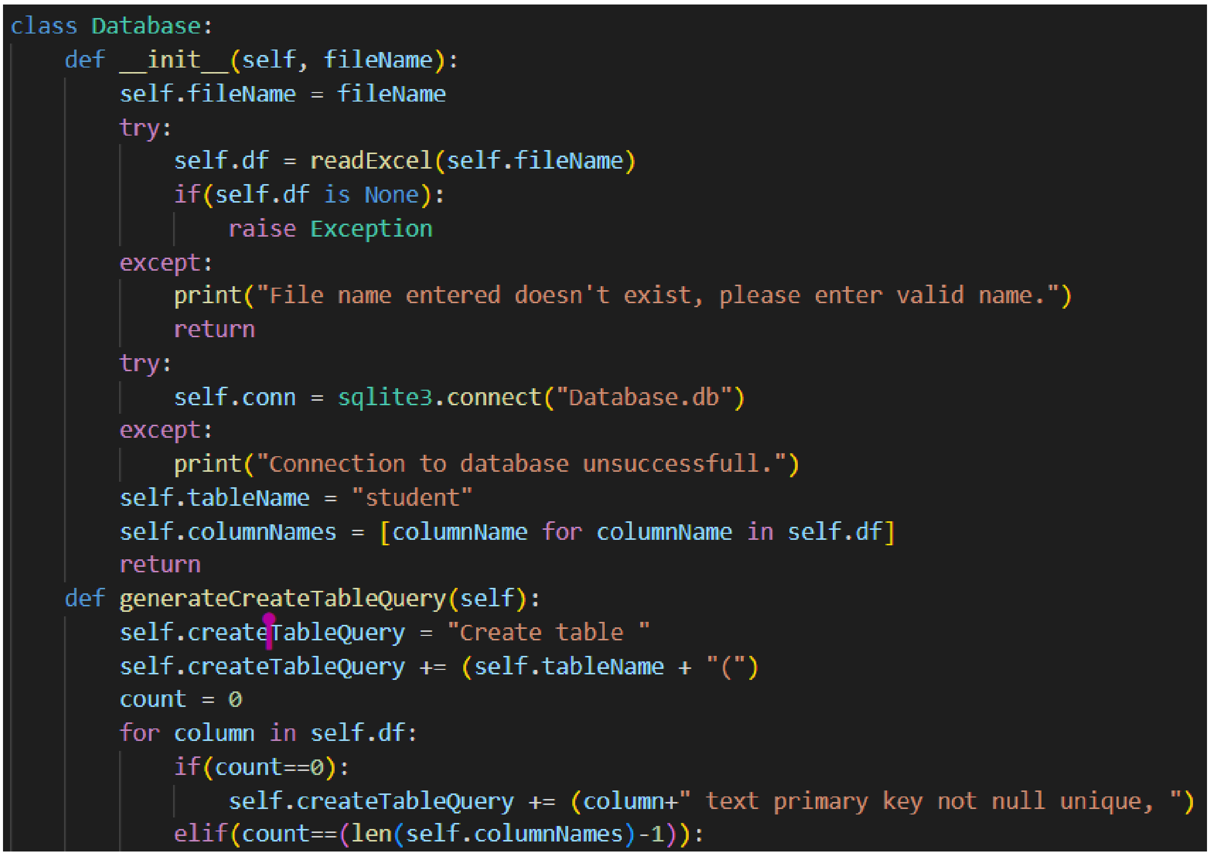
Technologies:

1. Use of functions:



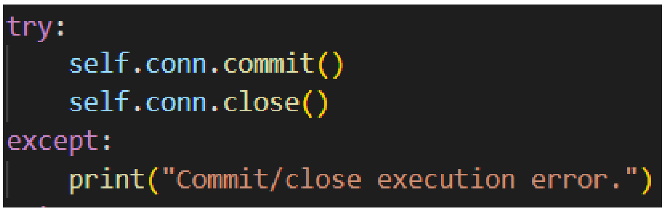
```
def readExcel(fileName):
    try:
        df = pd.read_excel(fileName)
    except:
        return None
    return df
```

2. Use of classes (includes functions and constructor):



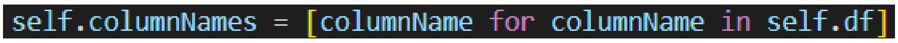
```
class Database:
    def __init__(self, fileName):
        self.fileName = fileName
        try:
            self.df = readExcel(self.fileName)
            if(self.df is None):
                raise Exception
        except:
            print("File name entered doesn't exist, please enter valid name.")
            return
        try:
            self.conn = sqlite3.connect("Database.db")
        except:
            print("Connection to database unsuccessfull.")
        self.tableName = "student"
        self.columnNames = [columnName for columnName in self.df]
        return
    def generateCreateTableQuery(self):
        self.createTableQuery = "Create table "
        self.createTableQuery += (self.tableName + "(")
        count = 0
        for column in self.df:
            if(count==0):
                self.createTableQuery += (column+" text primary key not null unique, ")
            elif(count==(len(self.columnNames)-1)):
```

3. Use of try-except to catch exception errors: It allows smooth functioning of program without traceback errors



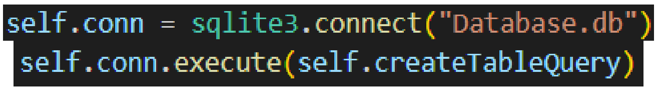
```
try:
    self.conn.commit()
    self.conn.close()
except:
    print("Commit/close execution error.")
```

4. List comprehension in Python:



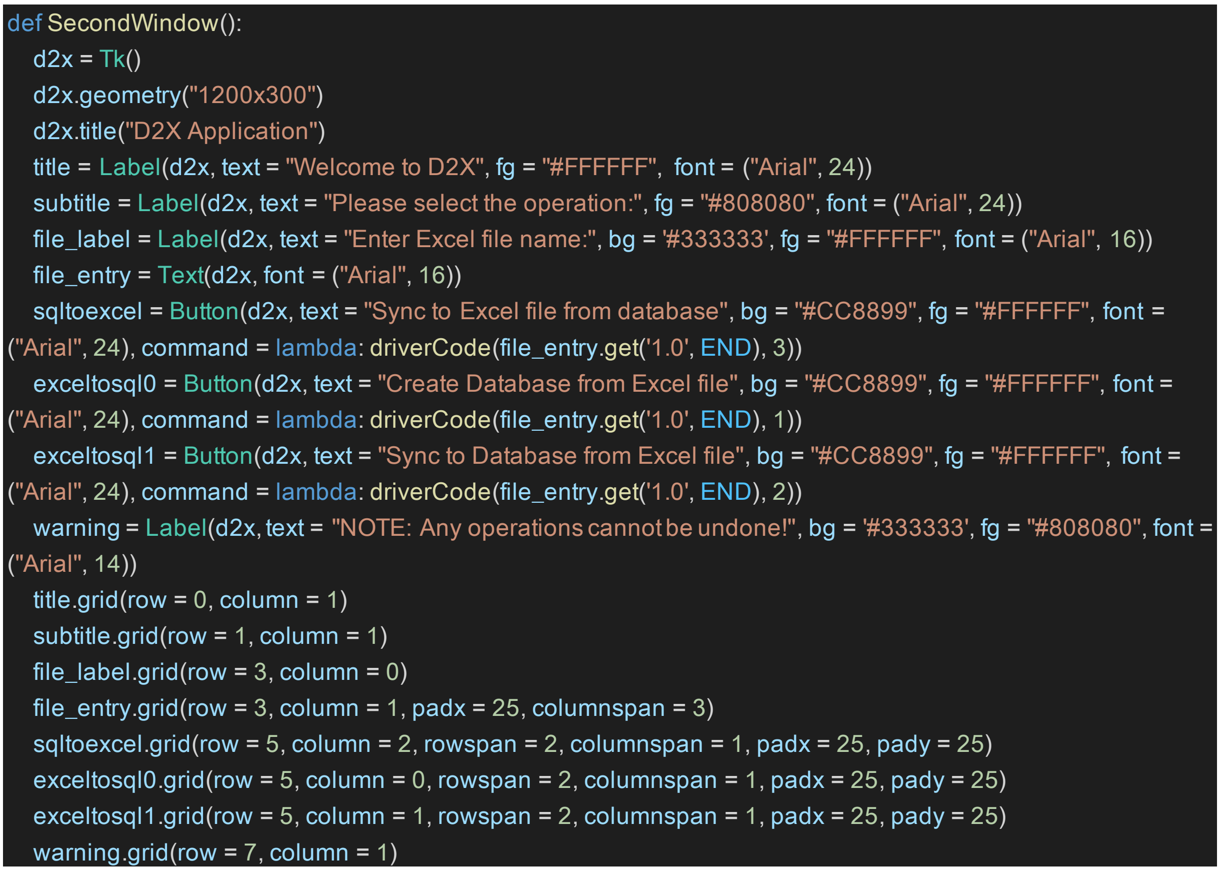
```
self.columnNames = [columnName for columnName in self.df]
```

5. Database connectivity and design:



```
self.conn = sqlite3.connect("Database.db")
 self.conn.execute(self.createTableQuery)
```

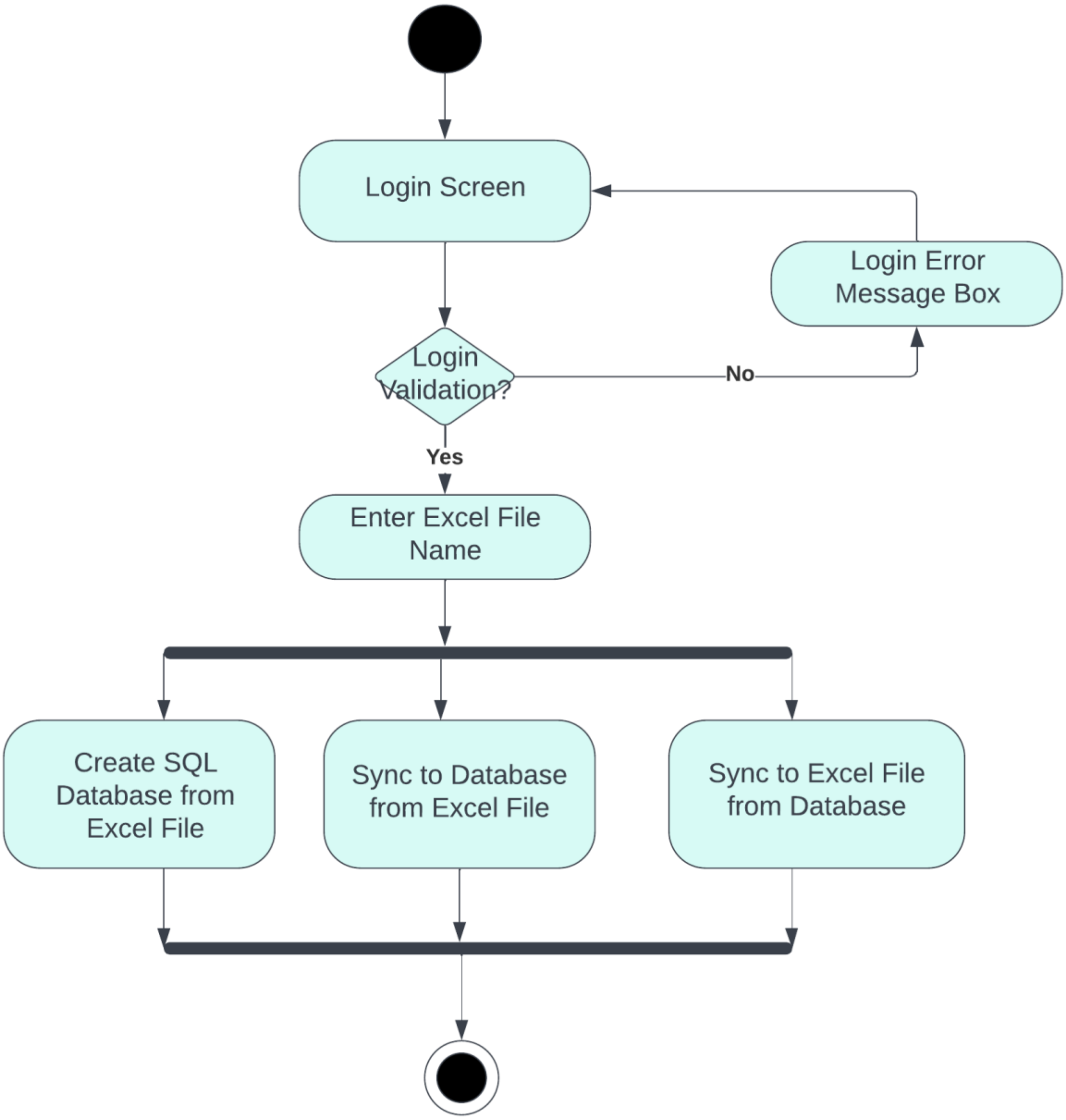
6. Tkinter: For GUI frontend designing



```
def SecondWindow():
   d2x = Tk()
   d2x.geometry("1200x300")
   d2x.title("D2X Application")
   title = Label(d2x, text = "Welcome to D2X", fg = "#FFFFFF",  font = ("Arial", 24))
   subtitle = Label(d2x, text = "Please select the operation:", fg = "#808080", font = ("Arial", 24))
   file_label = Label(d2x, text = "Enter Excel file name:", bg = '#333333', fg = "#FFFFFF", font = ("Arial", 16))
   file_entry = Text(d2x, font = ("Arial", 16))
   sqltoexcel = Button(d2x, text = "Sync to Excel file from database", bg = "#CC8899", fg = "#FFFFFF", font =
("Arial", 24), command = lambda: driverCode(file_entry.get('1.0', END), 3))
   exceltosql0 = Button(d2x, text = "Create Database from Excel file", bg = "#CC8899", fg = "#FFFFFF", font =
("Arial", 24), command = lambda: driverCode(file_entry.get('1.0', END), 1))
   exceltosql1 = Button(d2x, text = "Sync to Database from Excel file", bg = "#CC8899", fg = "#FFFFFF",  font =
("Arial", 24), command = lambda: driverCode(file_entry.get('1.0', END), 2))
   warning = Label(d2x, text = "NOTE: Any operations cannot be undone!", bg = '#333333', fg = "#808080", font =
("Arial", 14))
   title.grid(row = 0, column = 1)
   subtitle.grid(row = 1, column = 1)
   file_label.grid(row = 3, column = 0)
   file_entry.grid(row = 3, column = 1, padx = 25, columnspan = 3)
   sqltoexcel.grid(row = 5, column = 2, rowspan = 2, columnspan = 1, padx = 25, pady = 25)
   exceltosql0.grid(row = 5, column = 0, rowspan = 2, columnspan = 1, padx = 25, pady = 25)
   exceltosql1.grid(row = 5, column = 1, rowspan = 2, columnspan = 1, padx = 25, pady = 25)
   warning.grid(row = 7, column = 1)
```

Implementation

UML DIAGRAM:


Login Screen
Login Error Message Box
Login Validation?
No
Yes
Enter Excel File Name
Create SQL Database from Excel File
Sync to Database from Excel File
Sync to Excel File from Database

We have implemented the frontend using tkinter (A python Module). When the user runs the python script, the application will open which will ask for username and password.

After successfully logging in and entering the excel file name present in the same directory, There will be three options available to user, where user can choose any following from excel to SQL operations and vice versa.

A) On clicking “Sync to Excel file from Database”, it will import all the created or updated data from SQL to the Excel File.

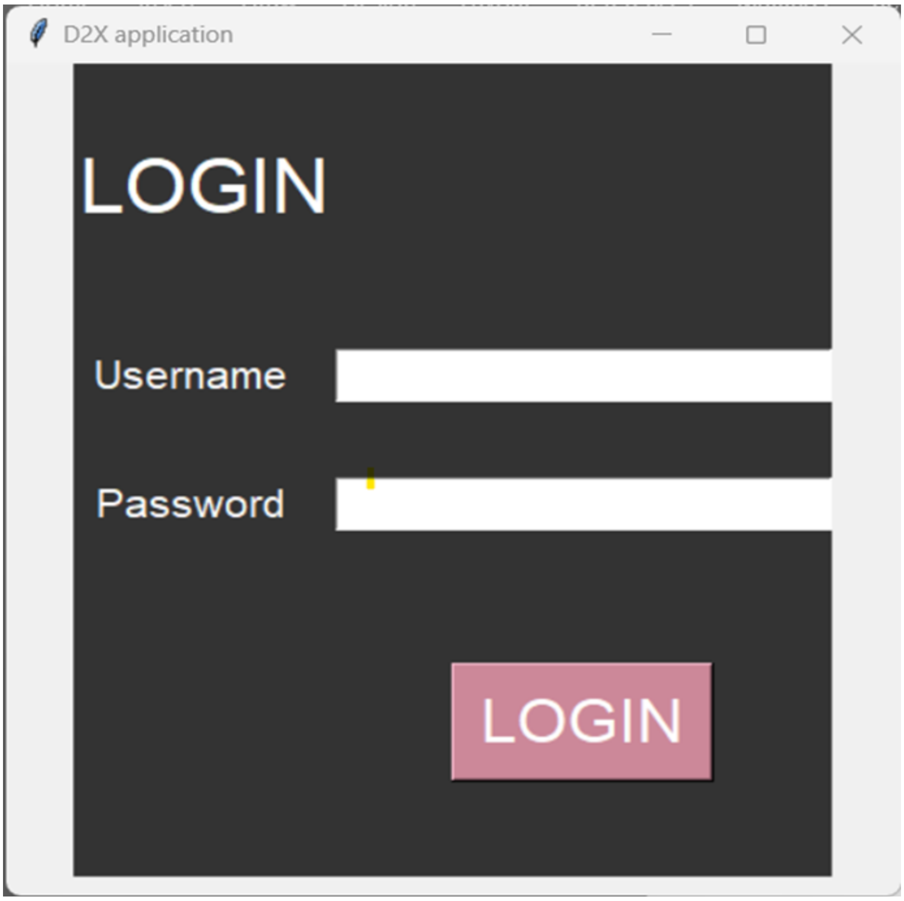
B) On clicking “Create Database from Excel File”, it will create a blank database file “database.db” and export all the data from Excel file to SQL.

C) On clicking “Sync to Database from Excel File”, whatever changes that has been made in the excel file will reflect in the Database with a click.

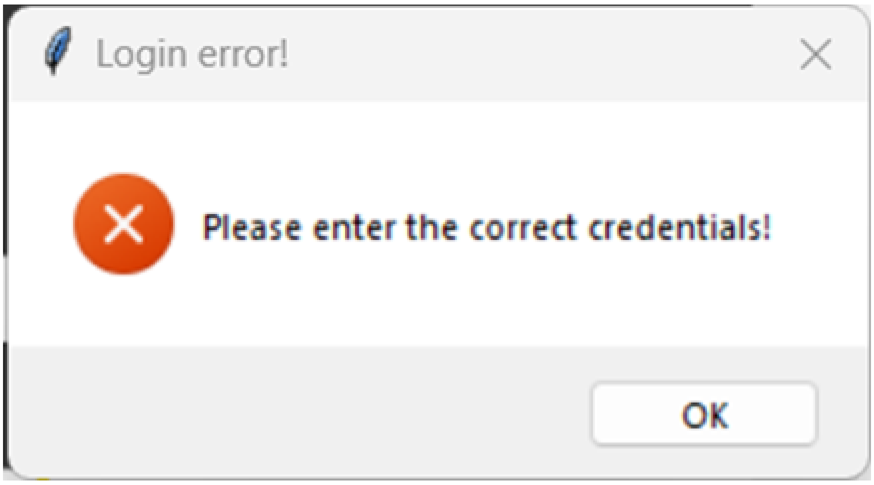
Result

As a result, The GUI application for D2X application is made as follows:

- Login Page: Where user can enter username and password to enter the main interface.

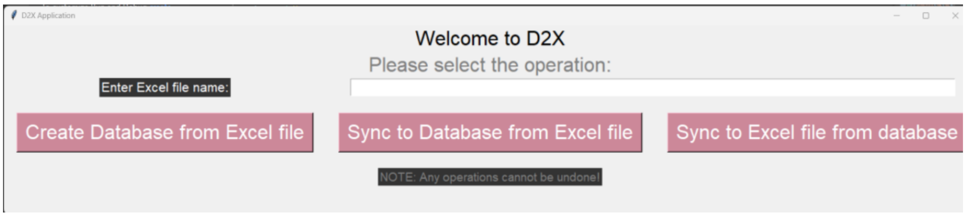


- Error message Box: This appears when username and password doesn’t match in Login Page and allows to reattempt login.



Please enter the correct credentials!

- Landing Page: After Successfully log in, User has to enter the file name of the excel, can do the following operations present in window.


D2X Application
Enter Excel file name:
Create Database from Excel file
Sync to Database from Excel file
Sync to Excel file from database

Welcome to D2X

Please select the operation:

NOTE: Any operations cannot be undone!

Conclusion

In D2X application, we provide an easy and simple application to clients. They can easily update and maintain the data in the Excel Sheet and easily keep in sync with the SQL database without using SQL Queries.

The software comes with authentication and easy-to-use interface in order to provide a smooth experience and get the desired functionality they need while manipulation of data between SQL and Excel.

Reference

1) https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_excel.html

2) https://docs.python.org/3/library/tk.html