

# Retirement home management system

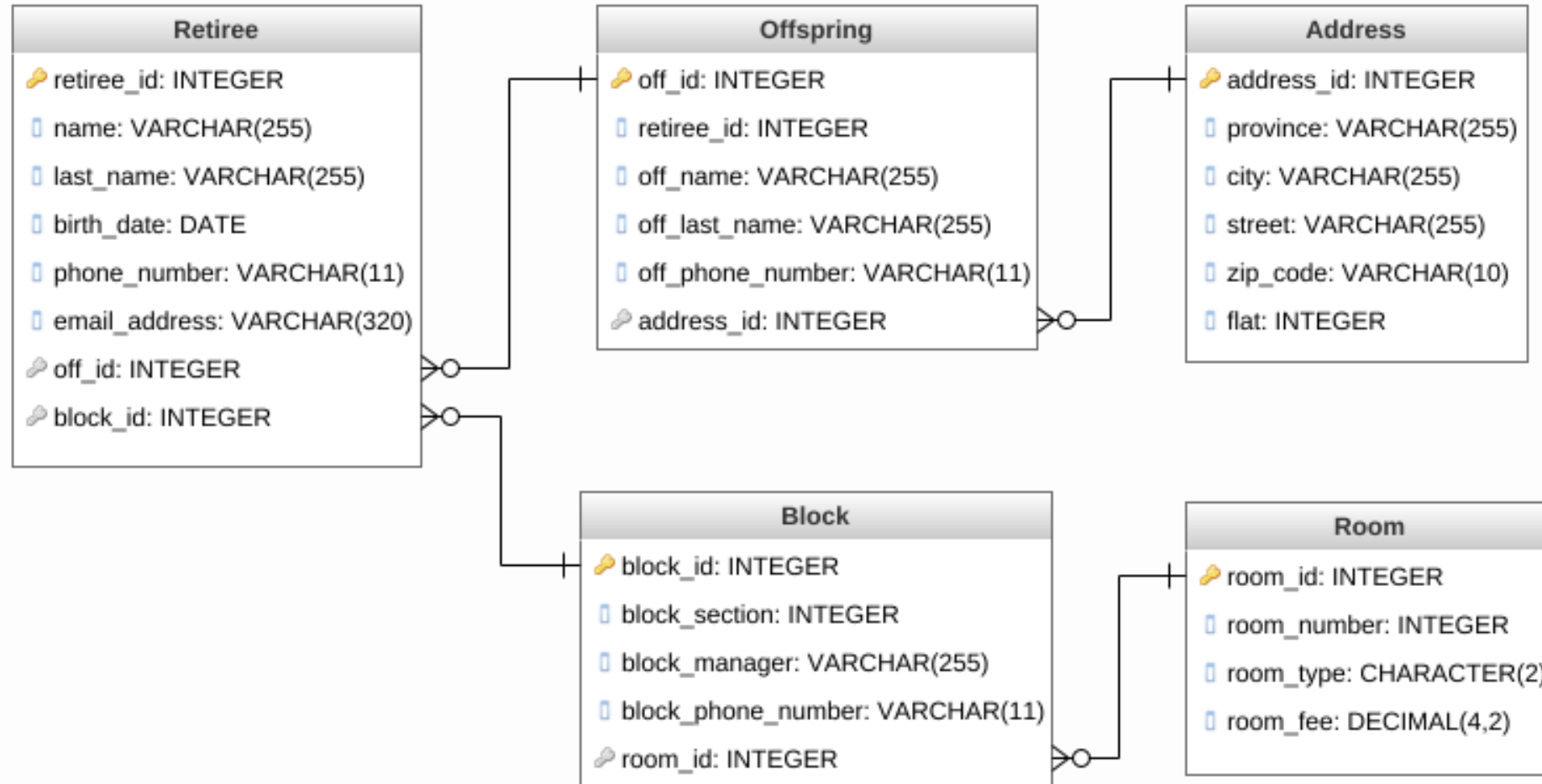
Python and SQL: intro / SQL platforms

Mohammad Saeed Pourjafar

# Introduction

The retirement home management system is a Graphical User Interface (GUI) for windows operating system which can be used to add, store and delete a retiree from the database. Also the administrator will be able to show the status quo of the retirees and update their information accordingly.

# Database scheme



# Database scheme

Database Structure			
Browse Data Edit Pragas Execute SQL			
Create Table Create Index Print			
Name	Type	Schema	
▼ Tables (7)			
▼ Address		CREATE TABLE Address (address_id INTEGER PRIMARY KEY NOT NULL, province VARCHAR(255), city VARC	
address_id	INTEGER	"address_id" INTEGER NOT NULL	
province	VARCHAR(255)	"province" VARCHAR(255)	
city	VARCHAR(255)	"city" VARCHAR(255)	
street	VARCHAR(255)	"street" VARCHAR(255)	
zip_code	VARCHAR(10)	"zip_code" VARCHAR(10)	
flat	INTEGER	"flat" INTEGER	
▼ Block		CREATE TABLE Block (block_id INTEGER PRIMARY KEY NOT NULL, block_section INTEGER, block_manager \	
block_id	INTEGER	"block_id" INTEGER NOT NULL	
block_section	INTEGER	"block_section" INTEGER	
block_manager	VARCHAR(255)	"block_manager" VARCHAR(255)	
block_phone_number	VARCHAR(11)	"block_phone_number" VARCHAR(11)	
room_id	INTEGER	"room_id" INTEGER	
▼ Log		CREATE TABLE Log ("" INTEGER, name VARCHAR(255), last_name VARCHAR(255), PRIMARY KEY("" AUTOI	
	INTEGER	"" INTEGER	
name	VARCHAR(255)	"name" VARCHAR(255)	
last_name	VARCHAR(255)	"last_name" VARCHAR(255)	
▼ Offspring		CREATE TABLE Offspring (off_id INTEGER PRIMARY KEY NOT NULL, Retireeret_id INTEGER NOT NULL, off_n	
off_id	INTEGER	"off_id" INTEGER NOT NULL	
Retireeret_id	INTEGER	"Retireeret_id" INTEGER NOT NULL	
off_name	TEXT	"off_name" TEXT NOT NULL	
off_last_name	TEXT	"off_last_name" TEXT NOT NULL	
off_phone_number	INTEGER	"off_phone_number" INTEGER	
address_id	INTEGER	"address_id" INTEGER	
▼ Retiree		CREATE TABLE "Retiree" ( "ret_id" INTEGER NOT NULL, "name" text, "last_name" text, "birth_date" TEXT, "	

# User manual

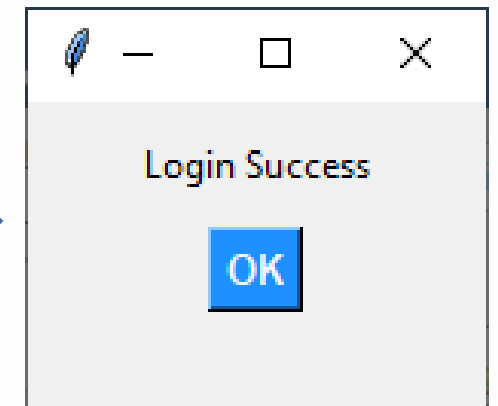
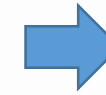
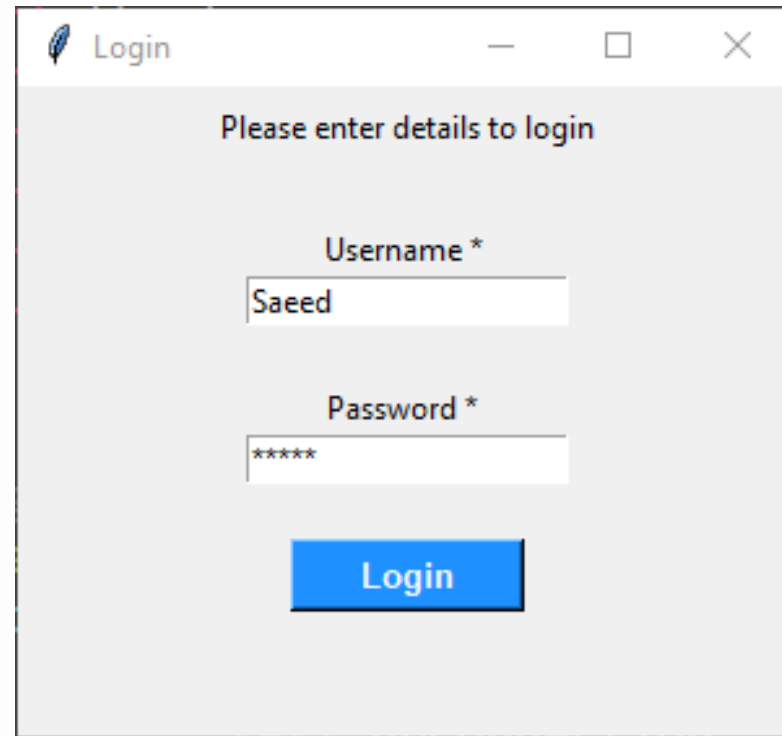
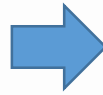
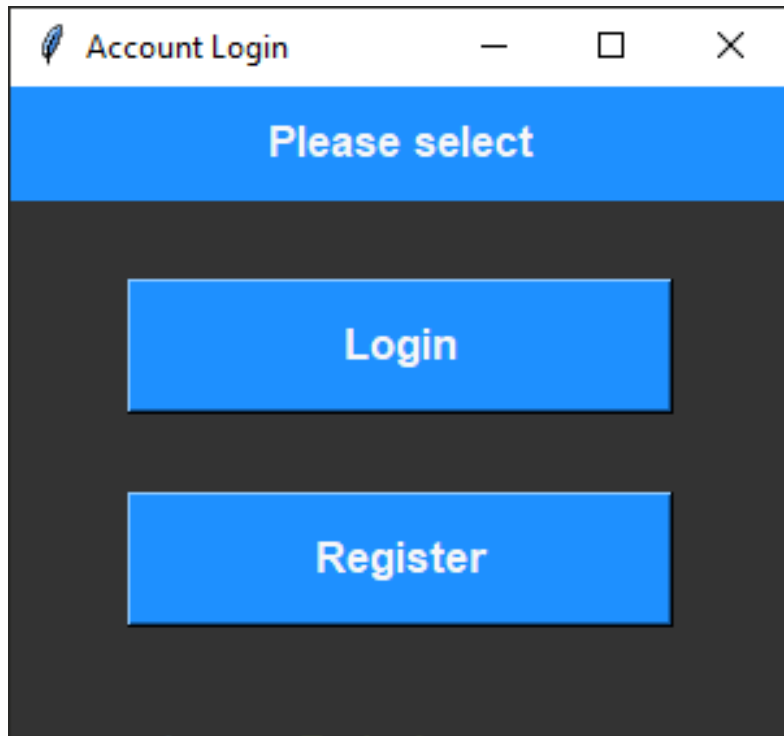
The application will be mainly used in retirement homes. Also, it can be extended and modified to any sort of application which deal with the different set of data from different segments e.g. day-care centers.

# User manual

First for running the application you can either run the Retiree.ipynb file from Jupyter Notebook or you can open the file Retiree.py with Sublime text editor, hit *Ctrl+B* to run it more quickly (recommended). Next, you will have the homepage of the retirement management system which you can then interact with by the graphical user interface and each button that is associated with the specific function and will be discussed in the upcoming section.

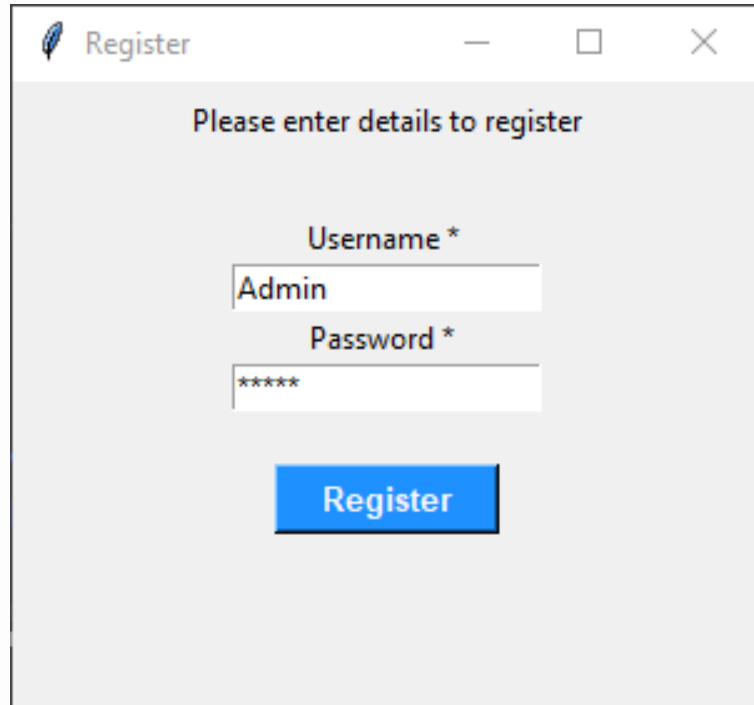
# Screenshots

## Login



# Screenshots

## Register



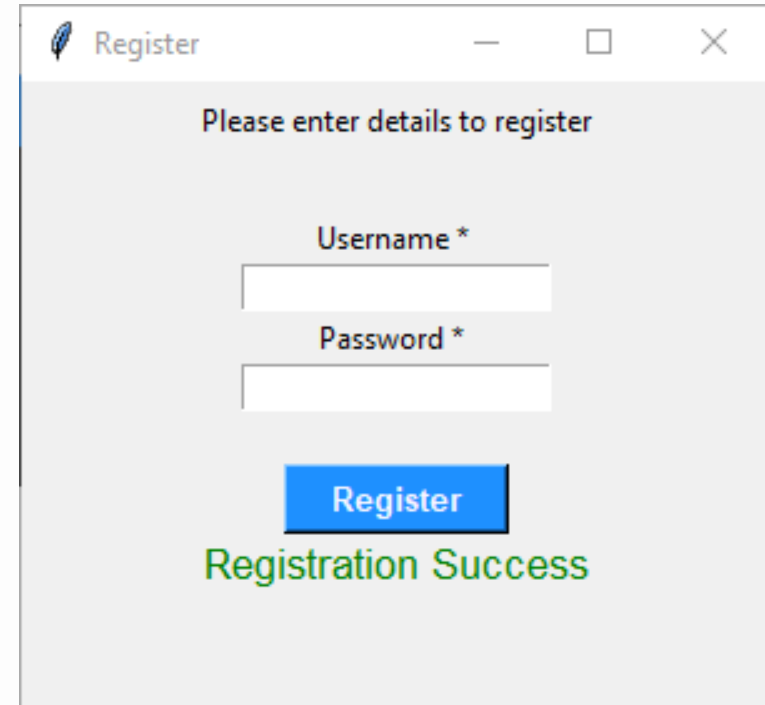
Register

Please enter details to register

Username \*

Password \*

Register



Register

Please enter details to register

Username \*

Password \*

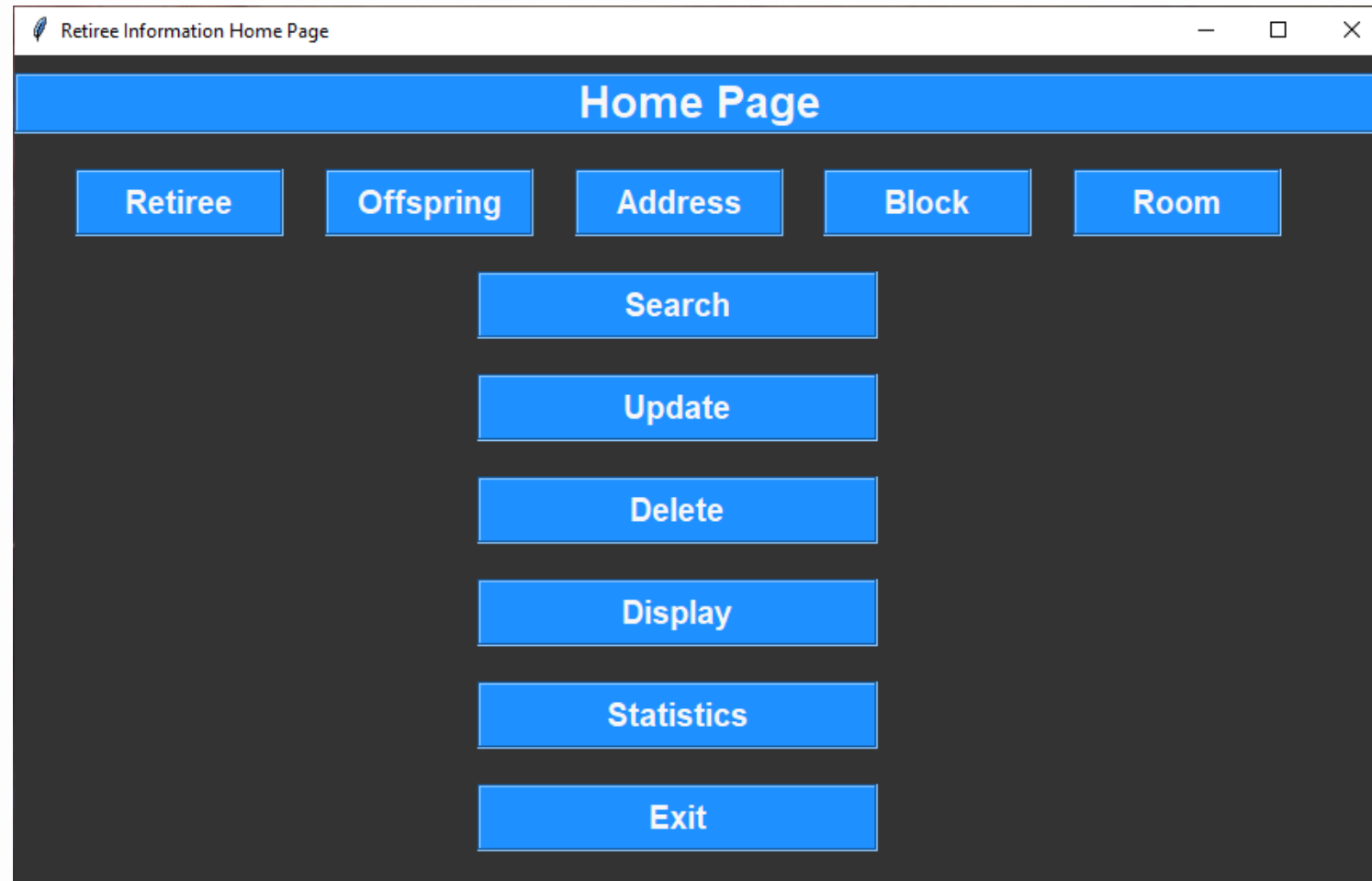
Register

Registration Success



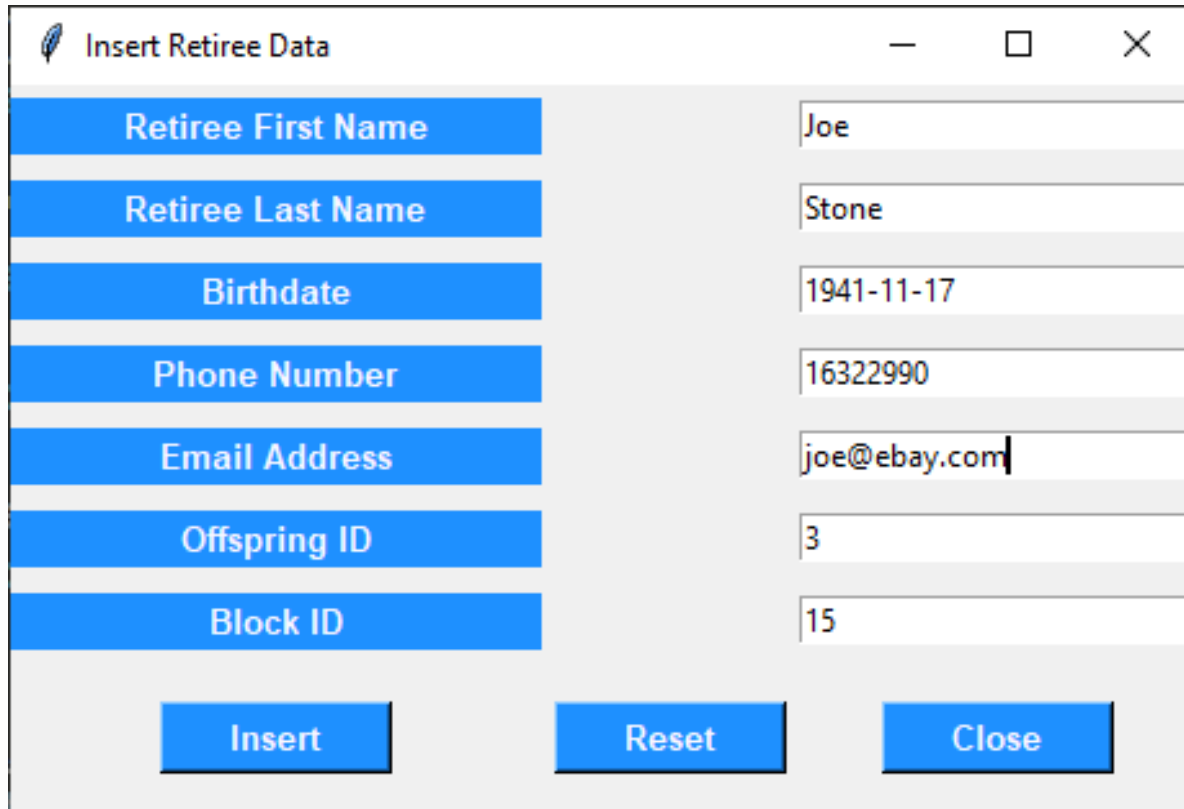
# Screenshots

**Homepage:** The front page of the application



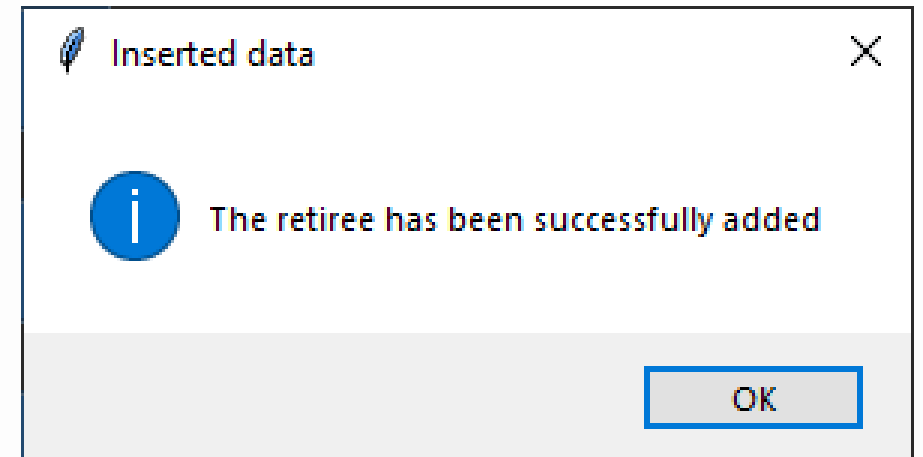
# Screenshots

- **Inserts:** They are designed in a separate buttons and therefore windows in order to give the user the ability to enter the information from top to bottom (Retiree to Room) or from bottom to top (Room to Retiree)



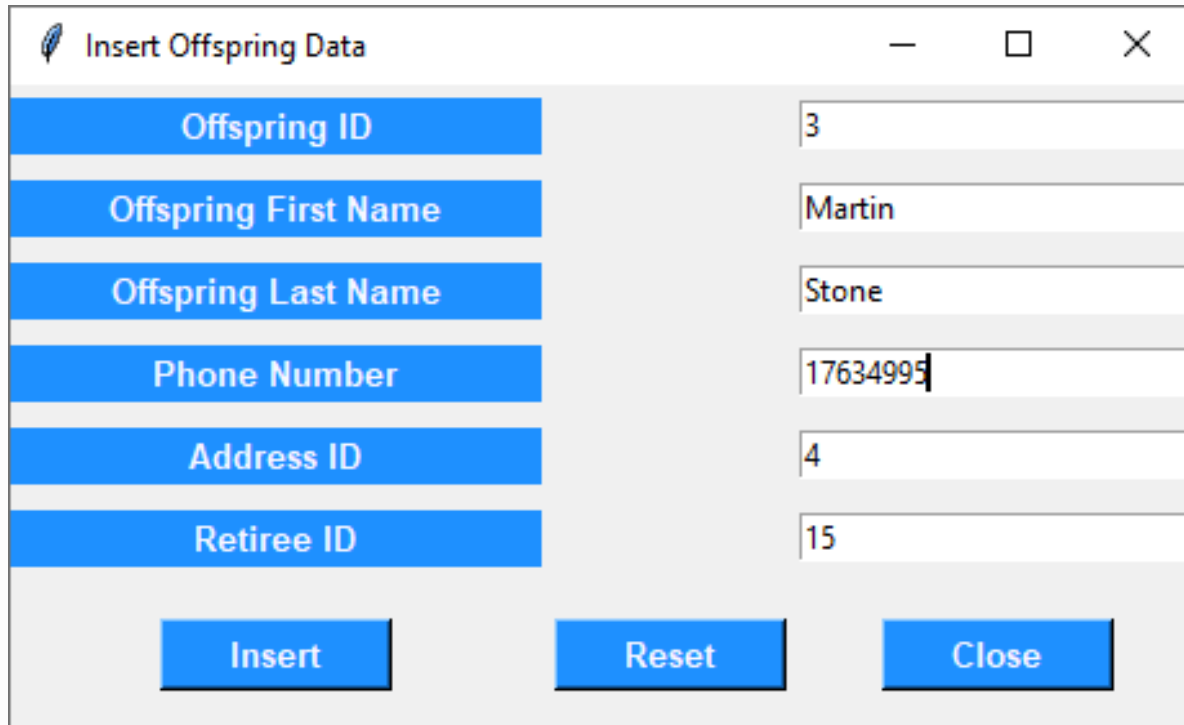
The 'Insert Retiree Data' window is a standard application window with a title bar containing a feather icon, the title 'Insert Retiree Data', and standard window controls (minimize, maximize, close). The main area contains seven blue buttons stacked vertically on the left, each corresponding to a text input field on the right. The buttons are labeled: 'Retiree First Name', 'Retiree Last Name', 'Birthdate', 'Phone Number', 'Email Address', 'Offspring ID', and 'Block ID'. The input fields contain the following data: 'Joe', 'Stone', '1941-11-17', '16322990', 'joe@ebay.com', '3', and '15'. At the bottom of the window are three blue buttons: 'Insert', 'Reset', and 'Close'.

Field Label	Value
Retiree First Name	Joe
Retiree Last Name	Stone
Birthdate	1941-11-17
Phone Number	16322990
Email Address	joe@ebay.com
Offspring ID	3
Block ID	15



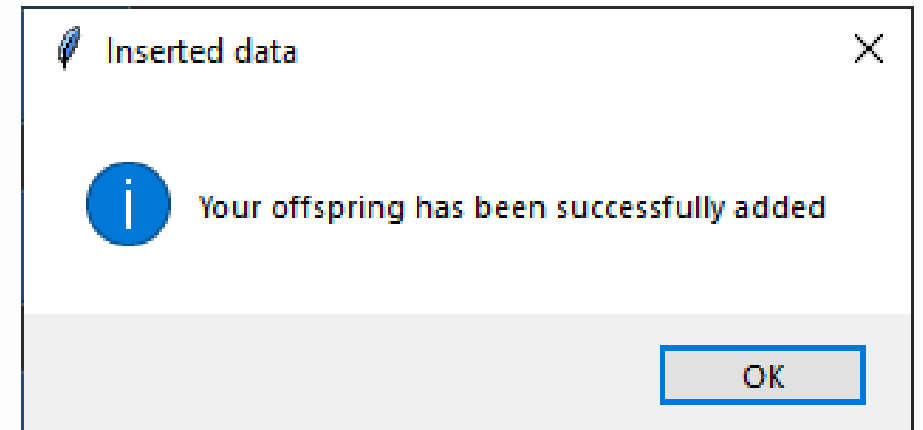
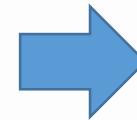
# Screenshots

- **Inserts:** They are designed in a separate buttons and therefore windows in order to give the user the ability to enter the information from top to bottom (Retiree to Room) or from bottom to top (Room to Retiree)



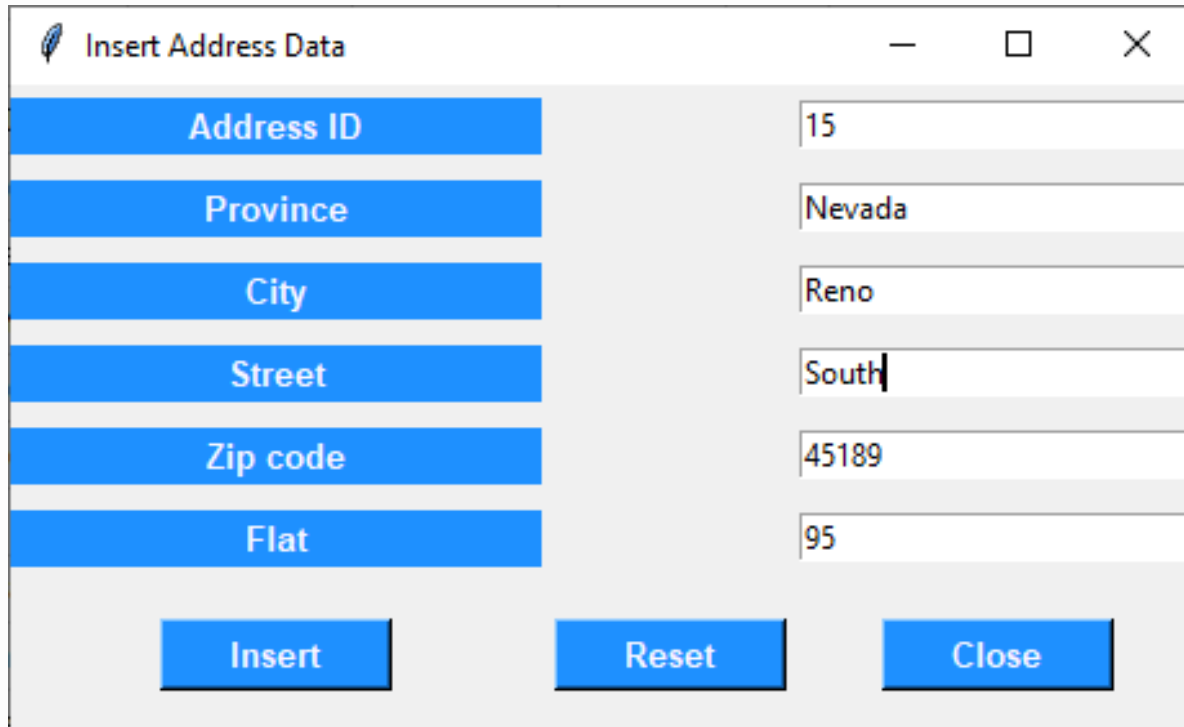
The 'Insert Offspring Data' window is a standard application window with a title bar containing a feather icon, the title 'Insert Offspring Data', and standard window controls (minimize, maximize, close). The main area contains six blue buttons on the left, each corresponding to a text input field on the right. The buttons are labeled 'Offspring ID', 'Offspring First Name', 'Offspring Last Name', 'Phone Number', 'Address ID', and 'Retiree ID'. The input fields contain the values '3', 'Martin', 'Stone', '17634995', '4', and '15' respectively. At the bottom of the window are three blue buttons labeled 'Insert', 'Reset', and 'Close'.

Field Label	Value
Offspring ID	3
Offspring First Name	Martin
Offspring Last Name	Stone
Phone Number	17634995
Address ID	4
Retiree ID	15



# Screenshots

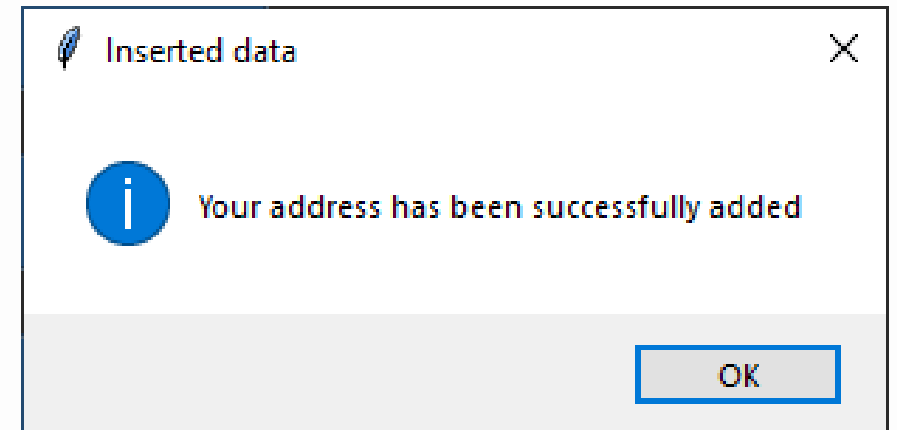
- **Inserts:** They are designed in a separate buttons and therefore windows in order to give the user the ability to enter the information from top to bottom (Retiree to Room) or from bottom to top (Room to Retiree)



Insert Address Data

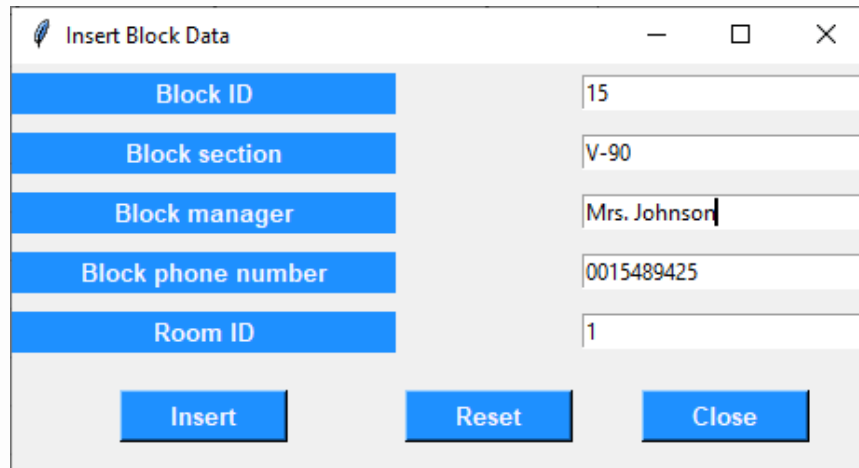
Address ID	15
Province	Nevada
City	Reno
Street	South
Zip code	45189
Flat	95

Insert Reset Close



# Screenshots

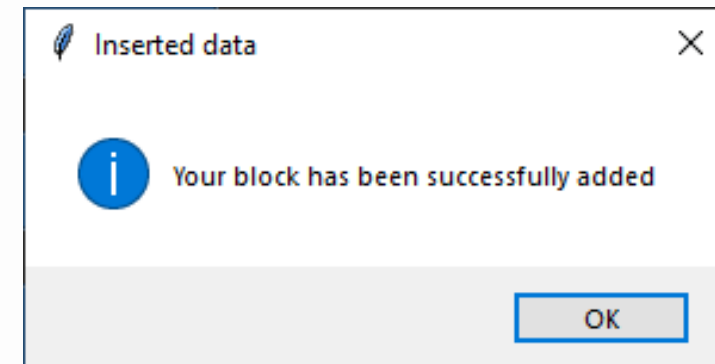
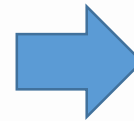
- **Inserts:** They are designed in a separate buttons and therefore windows in order to give the user the ability to enter the information from top to bottom (Retiree to Room) or from bottom to top (Room to Retiree)



Insert Block Data

Block ID	15
Block section	V-90
Block manager	Mrs. Johnson
Block phone number	0015489425
Room ID	1

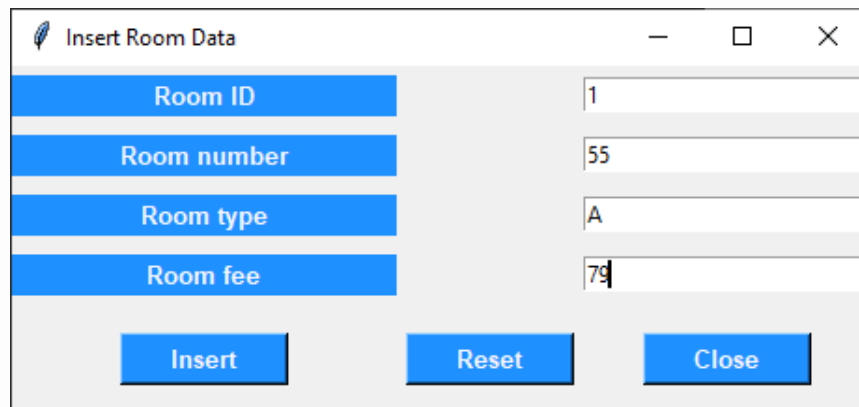
Insert Reset Close



Inserted data

*i* Your block has been successfully added

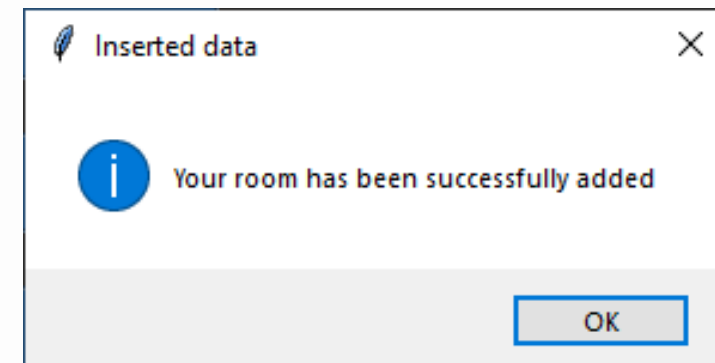
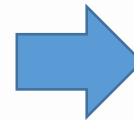
OK



Insert Room Data

Room ID	1
Room number	55
Room type	A
Room fee	79

Insert Reset Close



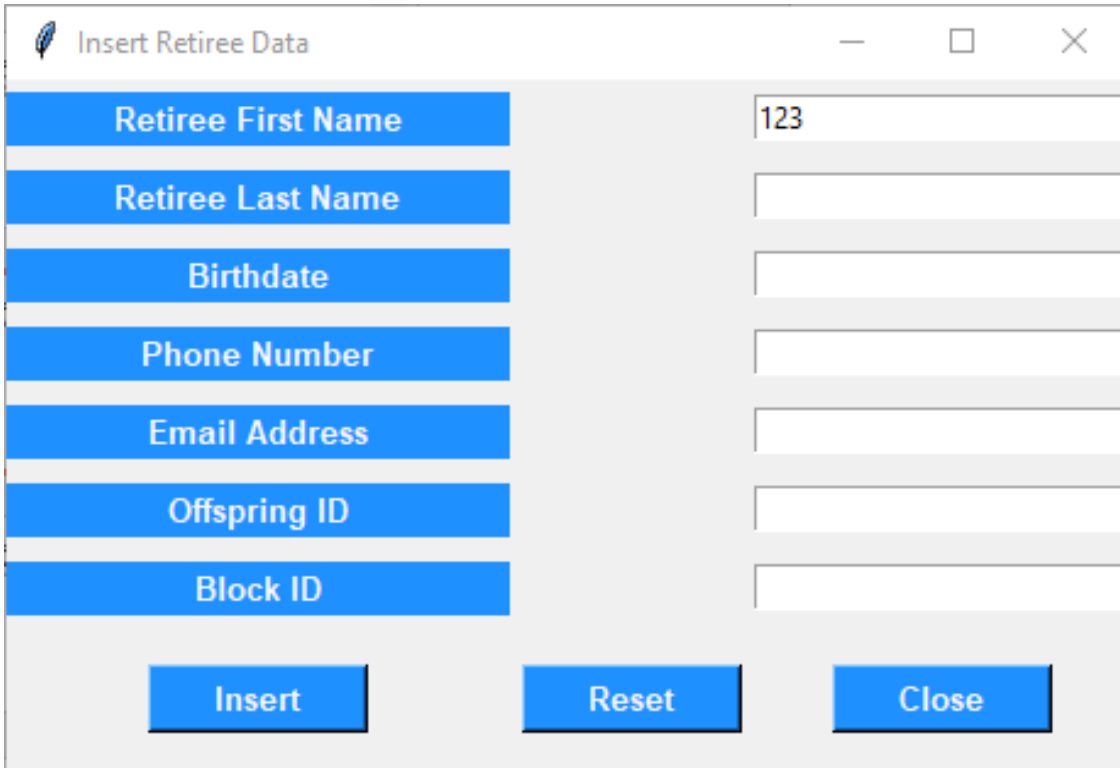
Inserted data

*i* Your room has been successfully added

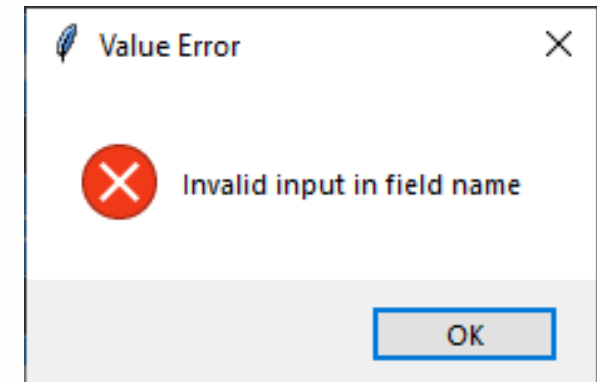
OK

# Screenshots

- **Inserts:** Will raise value error for wrong inputs

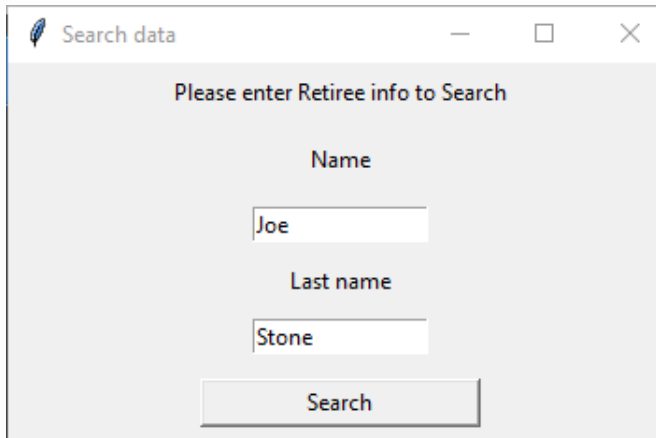


A screenshot of a web form titled "Insert Retiree Data". The form has a light gray background and a white border. It contains seven input fields, each with a blue label on the left and a white input area on the right. The labels are "Retiree First Name", "Retiree Last Name", "Birthdate", "Phone Number", "Email Address", "Offspring ID", and "Block ID". The "Retiree First Name" field contains the text "123". Below the input fields are three blue buttons labeled "Insert", "Reset", and "Close".



# Screenshots

**Search:** By clicking on this button, we will be able to search the retirees based on their name and last name



Search data

Please enter Retiree info to Search

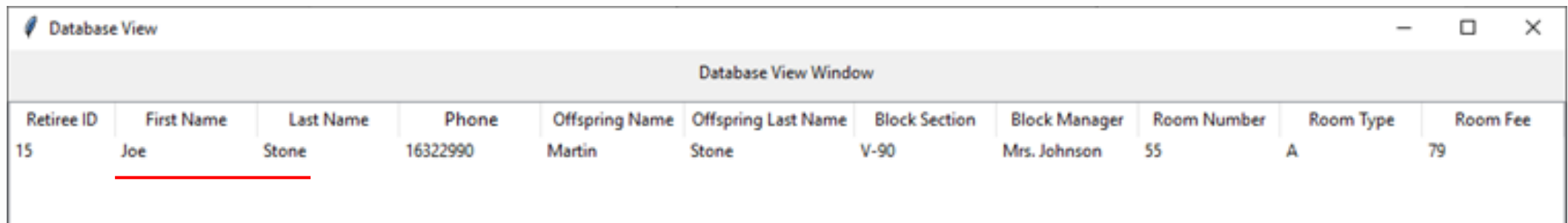
Name

Joe

Last name

Stone

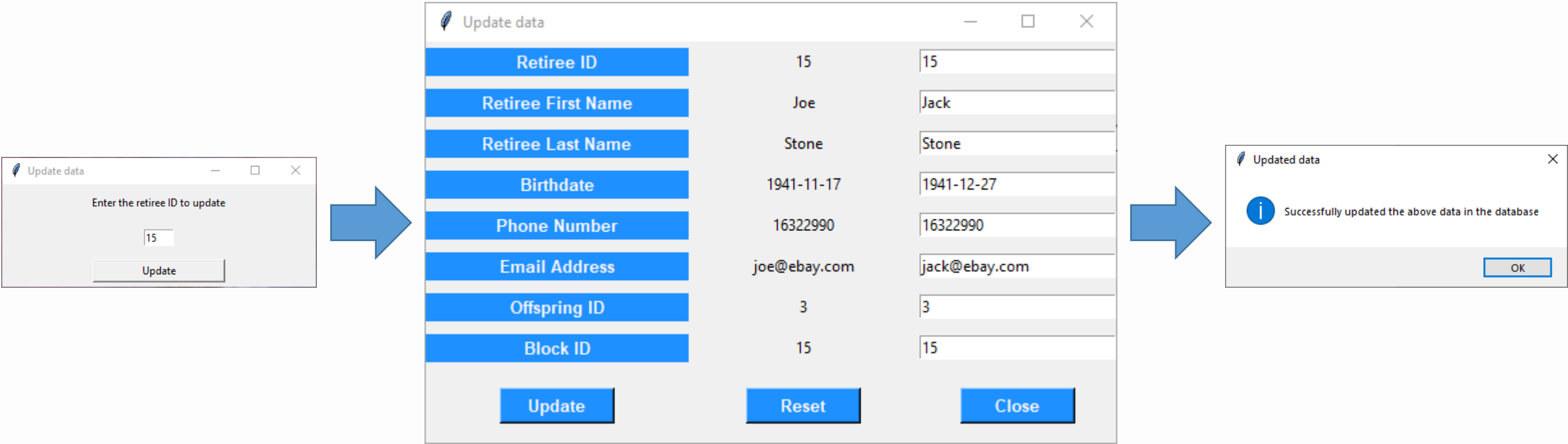
Search



Database View Window										
Retiree ID	First Name	Last Name	Phone	Offspring Name	Offspring Last Name	Block Section	Block Manager	Room Number	Room Type	Room Fee
15	<u>Joe</u>	Stone	16322990	Martin	Stone	V-90	Mrs. Johnson	55	A	79

# Screenshots

**Update:** Will update the information in Retiree table according to their ID. Previous values also shown in order to make it easier to compare the old and the new value.

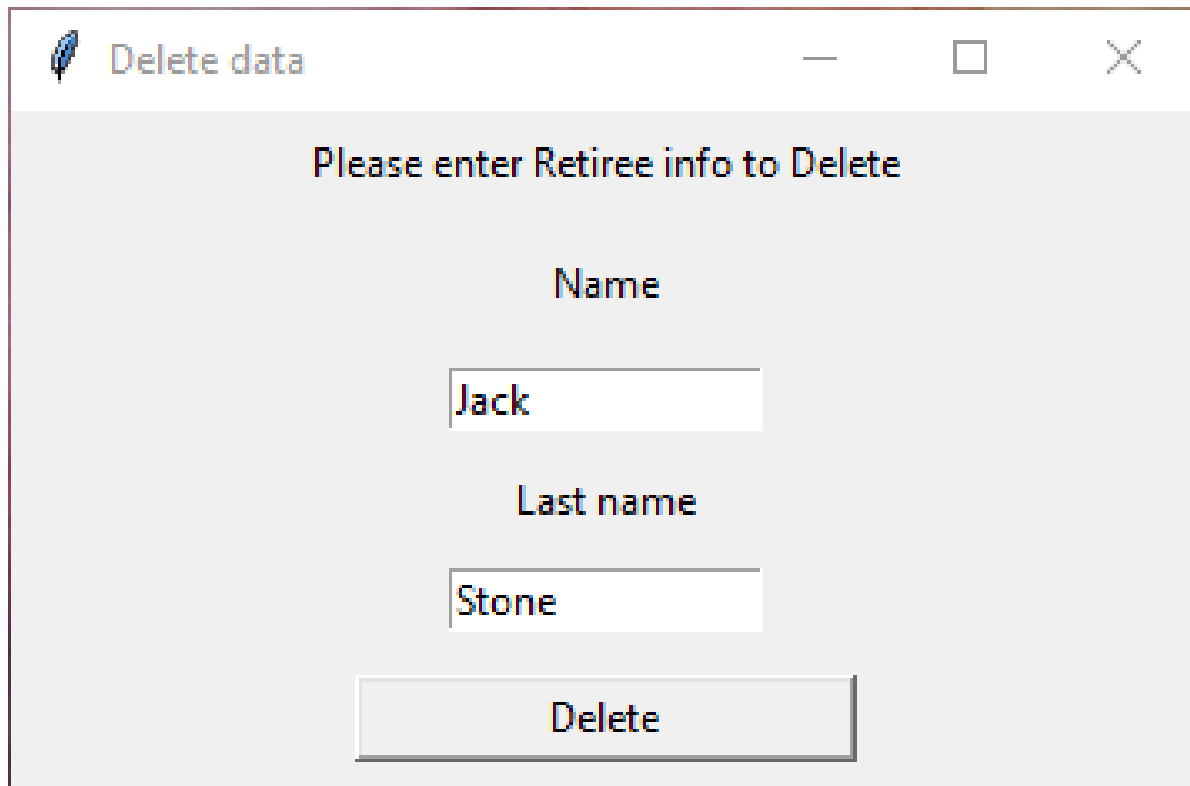


14	15	Jack	Stone	1941-12-27	16322990	jack@ebay.com	3	15
----	----	------	-------	------------	----------	---------------	---	----

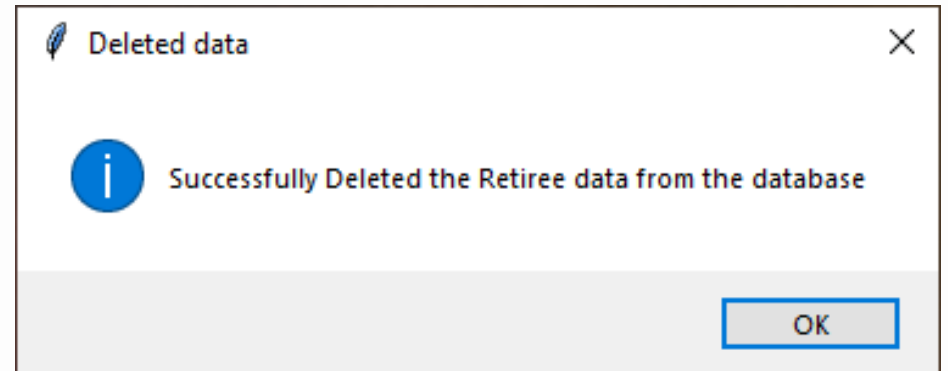
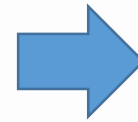


# Screenshots

**Delete:** Will delete a retirees from Retiree table based on their name and surname



A screenshot of a Windows-style dialog box titled "Delete data". The dialog has a light gray background and a title bar with a feather icon, a minus button, a maximize button, and a close button. The main content area contains the text "Please enter Retiree info to Delete". Below this, there are two text input fields. The first field is labeled "Name" and contains the text "Jack". The second field is labeled "Last name" and contains the text "Stone". At the bottom of the dialog, there is a button labeled "Delete".



A screenshot of a Windows-style dialog box titled "Deleted data". The dialog has a light gray background and a title bar with a feather icon and a close button. The main content area contains a blue information icon (a circle with a lowercase 'i') followed by the text "Successfully Deleted the Retiree data from the database". At the bottom right of the dialog, there is a button labeled "OK".

# Screenshots

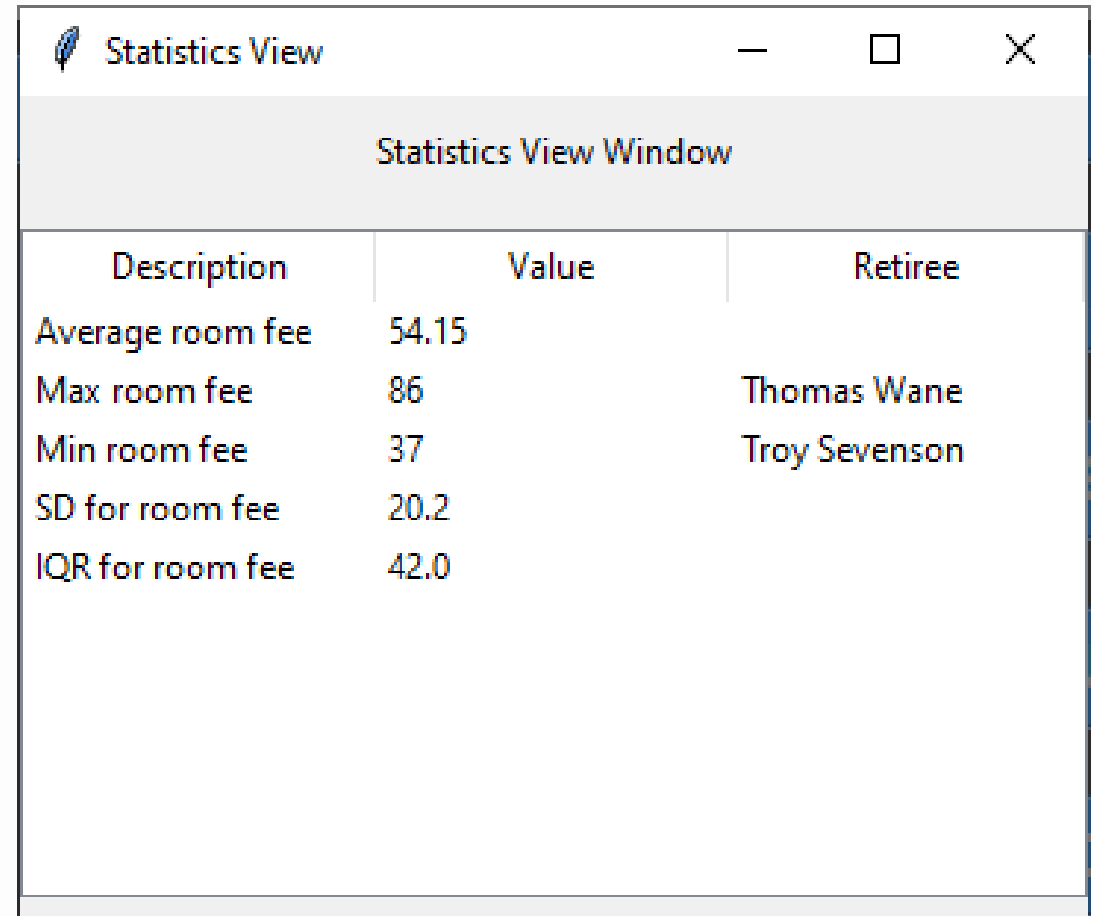
**Display:** Will display the scrollable information about retiree ID, name, last name, phone number, offspring name, offspring last name, block section, block manager, room number, room type and room fee.

Database View Window										
Retiree ID	First Name	Last Name	Phone	Offspring Name	Offspring Last Name	Block Section	Block Manager	Room Number	Room Type	Room Fee
1	Elizabeth	Webb	15296456	Tod	Webb	K-20	Mr. Burns	111	B	40
2	Jerry	Simpsons	15594196	Bart	Simpsons	K-20	Mr. Burns	114	B	43
3	Amantha	Green	19934941	Barney	Green	V-90	Mrs. Johnson	110	B	42
5	Troy	Sevenson	15521215	Sam	Sevenson	K-20	Mr. Burns	112	B	37
6	Amy	Kennedy	18842953	Homer	Kennedy	K-20	Mr. Burns	255	A	82
7	David	Owen	19463186	Lucas	Owen	K-20	Mr. Burns	112	B	37
8	Oscar	Robin	16386132	Stacy	Robin	V-90	Mrs. Johnson	209	A	84
9	Cathy	Turner	16521876	Shelby	Ville	K-20	Mr. Burns	243	A	85
10	Jeremy	Frank	19963290	Fred	Rowan	V-90	Mrs. Johnson	143	B	39
11	Sonia	Hanks	14589234	Eric	Flanders	V-90	Mrs. Johnson	143	B	45

# Screenshots

**Statistics:** Will display the statistics from the database based on the room fee and return the following values:

- Average room fee
- Max/Min room fee: Which retiree paid the maximum/minimum price for his/her?
- Standard deviation for the room fee
- Inter quartile range for the room fee



The screenshot shows a window titled 'Statistics View' with a standard Windows-style title bar (minimize, maximize, close buttons). Below the title bar is a header bar labeled 'Statistics View Window'. The main content area contains a table with three columns: 'Description', 'Value', and 'Retiree'. The table lists five statistics: Average room fee (54.15), Max room fee (86), Min room fee (37), SD for room fee (20.2), and IQR for room fee (42.0). The 'Max room fee' and 'Min room fee' rows have the name 'Thomas Wane' in the 'Retiree' column, while the 'SD for room fee' and 'IQR for room fee' rows have the name 'Troy Severson'.

Description	Value	Retiree
Average room fee	54.15	
Max room fee	86	Thomas Wane
Min room fee	37	Troy Severson
SD for room fee	20.2	
IQR for room fee	42.0	

# Screenshots

## SQL queries

Creating tables such as Retiree

```
self.dbCursor.execute("""CREATE TABLE IF NOT EXISTS Retiree (ret_id INTEGER PRIMARY KEY AUTOINCREMENT,  
name TEXT NOT NULL,  
last_name TEXT NOT NULL,  
birth_date TEXT,  
phone_number INTEGER,  
email_address TEXT,  
off_id INTEGER,  
block_id INTEGER,  
FOREIGN KEY (off_id)  
REFERENCES Offspring (off_id),  
FOREIGN KEY (block_id)  
REFERENCES Block (block_id))""")
```

# Screenshots

## SQL queries

### Creating indices

```
self.dbCursor.execute("""CREATE INDEX IF NOT EXISTS idx_Retiree_id ON Retiree (ret_id)""")
self.dbCursor.execute("""CREATE INDEX IF NOT EXISTS idx_name_lastname ON Retiree (name,last_name)""")
```

### Search function with SQL query based on inner join

```
def Search(self, name, last_name):
    self.dbCursor.execute("""SELECT ret_id,name,last_name,phone_number,off_name,
        off_last_name,block_section,block_manager,room_number,room_type,room_fee FROM Retiree
        INNER JOIN Offspring ON Retiree.off_id = Offspring.off_id
        INNER JOIN Block ON Retiree.block_id = Block.block_id
        INNER JOIN Room ON Block.room_id = Room.room_id WHERE name = ? AND last_name = ?""",
        (name,last_name))
    searchResults = self.dbCursor.fetchall()
    return searchResults
```

# Screenshots

## SQL queries

Statistics function SQL query for retrieving the value of room fee based on three tables

```
def statistics(self):  
    self.dbCursor.execute("""SELECT Room.room_fee FROM Retiree  
        INNER JOIN Block ON Block.block_id = Retiree.block_id  
        INNER JOIN Room ON Room.room_id = Block.room_id""")  
    records = self.dbCursor.fetchall()  
    return records
```

# Screenshots

## SQL queries

Max and Min query for room fee

```
def maxfee(self):
    self.dbCursor.execute("""SELECT name,last_name,MAX(Room.room_fee) FROM Retiree
        INNER JOIN Block ON Block.block_id = Retiree.block_id
        INNER JOIN Room ON Room.room_id = Block.room_id""")
    records = self.dbCursor.fetchall()
    return (records[0][0],records[0][1])

def minfee(self):
    self.dbCursor.execute("""SELECT name,last_name,MIN(Room.room_fee) FROM Retiree
        INNER JOIN Block ON Block.block_id = Retiree.block_id
        INNER JOIN Room ON Room.room_id = Block.room_id""")
    records = self.dbCursor.fetchall()
    return (records[0][0],records[0][1])
```

# Screenshots

## SQL queries

Creating a view named *v\_mostData* for the most important fields of database

```
self.dbCursor.execute("""CREATE VIEW IF NOT EXISTS v_mostData AS SELECT ret_id,  
name,last_name,phone_number,off_name, off_last_name,block_section,block_manager,  
room_number,room_type,room_fee FROM Retiree  
INNER JOIN Offspring ON Retiree.off_id = Offspring.off_id  
INNER JOIN Block ON Retiree.block_id = Block.block_id  
INNER JOIN Room ON Block.room_id = Room.room_id UNION ALL SELECT * FROM Address""")
```



# External packages

For this project, the Python's de facto standard GUI interface **Tkinter** is used which is a toolkit for developing applications in different operating systems also the **numpy** package is used for some statistical inference such as standard deviation and inter quartile range for the room fee.