



# A2 UNIT 5

## 6. Software Development

Ryan Beattie  
Candidate Number: 8002  
Centre Number: 71741

# Database Tables

## Code

```
File Edit Selection View Go Run Terminal Help
tables.py - Visual Studio Code

C:\Users> Ryan > OneDrive - C2K > A2 2020-2021 > Computer Science > A2 Unit 5 > Code > tables.py > ...
1 import sqlite3 #Imports sqlite3 to create and write to the database file.
2 with sqlite3.connect("lec09.db") as db: #Connects to the database file.
3     cursor = db.cursor() #Creates the cursor.
4
5 #Create a table to store branch details.
6 cursor.execute("""
7 CREATE TABLE IF NOT EXISTS Branches(
8 BranchID INTEGER PRIMARY KEY,
9 Town VARCHAR(30) NOT NULL,
10 Postcode VARCHAR(7) NOT NULL,
11 Email VARCHAR(30) NOT NULL,
12 Telephone VARCHAR(15) NOT NULL;
13 """)
14
15 #Create a table for customer records.
16 cursor.execute("""
17 CREATE TABLE IF NOT EXISTS Customer(
18 CustomerID INTEGER PRIMARY KEY,
19 BranchID INTEGER NOT NULL,
20 Name VARCHAR(20) NOT NULL,
21 Surname VARCHAR(30) NOT NULL,
22 DateOfBirth DATE NOT NULL,
23 Town VARCHAR(30) NOT NULL,
24 Postcode VARCHAR(7) NOT NULL,
25 EmailAddress VARCHAR(30) NOT NULL,
26 TelephoneNo VARCHAR(15) NOT NULL,
27 MedicalConditions TEXT,
28 FOREIGN KEY(BranchID) REFERENCES Branches(BranchID));
29 """)
30
31 #Create a table for staff records.
32 cursor.execute("""
33 CREATE TABLE IF NOT EXISTS Staff(
34 StaffID INTEGER PRIMARY KEY,
35 BranchID INTEGER NOT NULL,
36 Name VARCHAR(20) NOT NULL,
37 Surname VARCHAR(30) NOT NULL,
38 Position VARCHAR(15) NOT NULL,
39 Email VARCHAR(30) NOT NULL,
40 Telephone VARCHAR(15) NOT NULL,
41 FOREIGN KEY(BranchID) REFERENCES Branches(BranchID));
42 """)
43
44 #Create a table for supplier records.
45 cursor.execute("""
46 CREATE TABLE IF NOT EXISTS Suppliers(
47 SupplierID INTEGER PRIMARY KEY,
48 SupplierName VARCHAR(40) NOT NULL,
49 Postcode VARCHAR(7) NOT NULL,
50
51 Postcode VARCHAR(8) NOT NULL,
52 Email VARCHAR(30) NOT NULL,
53 Telephone VARCHAR(15) NOT NULL;
54 """)
55
56 #Create a table for product records.
57 cursor.execute("""
58 CREATE TABLE IF NOT EXISTS Products(
59 ProductID INTEGER PRIMARY KEY,
60 SupplierID INTEGER,
61 ProductName VARCHAR(30) NOT NULL,
62 ProductDescription TEXT,
63 Price REAL NOT NULL,
64 FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID);
65 """)
66
67 #Create a table for appointments.
68 cursor.execute("""
69 CREATE TABLE IF NOT EXISTS Appointments(
70 AppointmentID INTEGER PRIMARY KEY,
71 CustomerID INTEGER NOT NULL,
72 AppointmentDate DATE NOT NULL,
73 AppointmentTime TEXT NOT NULL,
74 StaffID INTEGER NOT NULL,
75 FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID),
76 FOREIGN KEY(StaffID) REFERENCES Staff(StaffID);
77 """)
78
79 #Create a table for order records.
80 cursor.execute("""
81 CREATE TABLE IF NOT EXISTS Orders(
82 OrderID INTEGER PRIMARY KEY,
83 OrderDate DATE NOT NULL,
84 BranchID INTEGER NOT NULL,
85 SupplierID INTEGER NOT NULL,
86 ProductID INTEGER NOT NULL,
87 Quantity INTEGER NOT NULL,
88 OrderTotal REAL,
89 FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID),
90 FOREIGN KEY(ProductID) REFERENCES Products(ProductID),
91 FOREIGN KEY(BranchID) REFERENCES Branches(BranchID);
92 """)
93
94 #Create a table to store login details.
95 cursor.execute("""
96 CREATE TABLE IF NOT EXISTS Login(
97 StaffID INTEGER NOT NULL,
98 Name VARCHAR(30) NOT NULL,
99 Password VARCHAR(30) NOT NULL;
100 """)
101
Python 3.8.3 32-bit Ln 1, Col 74 Spaces: 4 UTF-8 CRLF Python
```

```
File Edit Selection View Go Run Terminal Help
tables.py - Visual Studio Code

C:\Users> Ryan > OneDrive - C2K > A2 2020-2021 > Computer Science > A2 Unit 5 > Code > tables.py > ...
49 Postcode VARCHAR(8) NOT NULL,
50 Email VARCHAR(30) NOT NULL,
51 Telephone VARCHAR(15) NOT NULL;
52 """)
53
54 #Create a table for product records.
55 cursor.execute("""
56 CREATE TABLE IF NOT EXISTS Products(
57 ProductID INTEGER PRIMARY KEY,
58 SupplierID INTEGER,
59 ProductName VARCHAR(30) NOT NULL,
60 ProductDescription TEXT,
61 Price REAL NOT NULL,
62 FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID);
63 """)
64
65 #Create a table for appointments.
66 cursor.execute("""
67 CREATE TABLE IF NOT EXISTS Appointments(
68 AppointmentID INTEGER PRIMARY KEY,
69 CustomerID INTEGER NOT NULL,
70 AppointmentDate DATE NOT NULL,
71 AppointmentTime TEXT NOT NULL,
72 StaffID INTEGER NOT NULL,
73 FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID),
74 FOREIGN KEY(StaffID) REFERENCES Staff(StaffID);
75 """)
76
77 #Create a table for order records.
78 cursor.execute("""
79 CREATE TABLE IF NOT EXISTS Orders(
80 OrderID INTEGER PRIMARY KEY,
81 OrderDate DATE NOT NULL,
82 BranchID INTEGER NOT NULL,
83 SupplierID INTEGER NOT NULL,
84 ProductID INTEGER NOT NULL,
85 Quantity INTEGER NOT NULL,
86 OrderTotal REAL,
87 FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID),
88 FOREIGN KEY(ProductID) REFERENCES Products(ProductID),
89 FOREIGN KEY(BranchID) REFERENCES Branches(BranchID);
90 """)
91
92 #Create a table to store login details.
93 cursor.execute("""
94 CREATE TABLE IF NOT EXISTS Login(
95 StaffID INTEGER NOT NULL,
96 Name VARCHAR(30) NOT NULL,
97 Password VARCHAR(30) NOT NULL;
98 """)
99
Python 3.8.3 32-bit Ln 43, Col 1 Spaces: 4 UTF-8 CRLF Python
```

```

106 Name VARCHAR(30) NOT NULL,
107 Password VARCHAR(30) NOT NULL,
108 AccessLevel INTEGER NOT NULL,
109 FOREIGN KEY(StaffID) REFERENCES Staff(StaffID);
110
111
112
113 #Create a table for prescriptions.
114 cursor.execute('''
115 CREATE TABLE IF NOT EXISTS Prescriptions(
116 PrescriptionID INTEGER NOT NULL,
117 PrescriptionDate VARCHAR(10) NOT NULL,
118 CustomerID INTEGER NOT NULL,
119 PrescriptionDetails TEXT,
120 StaffID INTEGER NOT NULL,
121 FOREIGN KEY(StaffID) REFERENCES Staff(StaffID));
122
123
124 #Insert the details of the Lee Opticians branches into the branches table.
125 cursor.execute('''
126 INSERT INTO Branches (Town, Postcode, Email, Telephone)
127 VALUES ('Warrenpoint', 'BT34 3LF', 'info@leeopticians.com', '028 4175 3030'),
128 ('Crossmaglen', 'BT35 9HG', 'crossmaglen@leeopticians.com', '028 3086 8866'),
129 ('Camlough', 'BT35 7JG', 'camlough@leeopticians.com', '028 3044 2612');
130
131
132
133 db.commit() #Save the changes made to the database file.
134
135 #Insert an admin record into the login table for development and maintenance.
136 cursor.execute('''
137 INSERT INTO Login (StaffID, Name, Password, AccessLevel)
138 VALUES ('0', 'Ryan', 'admin123', '1');
139
140
141
142 db.commit() #Save the changes made to the database file.
143
144
145

```

## Result

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CJK\A2 2020-2021\Computer Science\A2 Unit 5\Code\LeeOpt.db

File Edit View Tools Help

Database Structure Browse Data Edit Primitives Execute SQL

Tables: Customer

Filter	CustomerID	BranchID	Name	Surname	DateOfBirth	Town	Postcode	EmailAddress	TelephoneNo	MedicalConditions
1	1	3	Robert	Dunne	26/08/1973	Camlough	BT35 4FA	robert772@hotmail.com	07674868732	
2	2	3	Bob	McShane	05/06/1985	Newry	BT34 3BG	bmcshane05@gmail.com	07865344789	Asthma
3	3	3	John	Murphy	07/02/1982	Armagh	BT60 7GH	johnmurphy07@gmail.com	07853965356	
4	4	3	Samantha	O'Hare	30/05/1999	Newry	BT10 1AA	samohare3005@outlook.com	07464328827	Asthma
5	5	3	Jane	McFarland	05/07/1982	Newry	BT34 1FC	jmcparland82@hotmail.com	07462842664	
6	6	3	John	Smyth	10/07/1989	Camlough	BT34 5AC	johnsmyth89@gmail.com	07058534087	
7	7	3	Mary	Hughes	06/06/1995	Camlough	BT10 7EF	mhughes06@outlook.com	07856321516	
8	8	3	Brian	Brennan	12/06/1972	Newry	BT60 2HG	bbrennan72@gmail.com	07586332576	
9	9	3	Kathleen	Kiley	08/02/1997	Armagh	BT34 5UH	kathleenk02@hotmail.com	07688465891	
10	10	3	Mary	McLennan	10/09/1999	Camlough	BT60 3BV	marymcl@hotmail.com	07985463279	Diabetes
11	11	3	Maura	Curren	21/07/2001	Camlough	BT35 9XN	mauracurr01@gmail.com	07890833245	Nut allergy
12	12	3	Lorcan	Maguire	02/03/2002	Newry	BT34 5NH	lorcanm02@hotmail.com	07985344754	
13	13	3	Aidan	McNamara	27/04/1968	Armagh	BT60 0EF	seanlynch06@gmail.com	07985999324	
14	14	3	Sean	Lynch	06/10/2000	Bessbrook	BT35 8UJ	seanlynch06@gmail.com	07957333757	
15	15	3	Carmel	Murphy	18/11/1987	Bessbrook	BT60 8TY	carmelmurph1@hotmail.com	07958555723	

1 - 14 of 15

Go to: 1

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Branches

BranchID	Town	Postcode	Email	Telephone
1	Warrenpoint	BT34 3UF	info@leeopticians.com	028 4175 3030
2	Crossmaglen	BT35 9HG	crossmaglen@leeopticians.com	028 3086 8866
3	Camlough	BT35 7YG	camlough@leeopticians.com	028 3044 2612

1 - 9 of 9

Go to: 1

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Appointments

AppointmentID	CustomerID	AppointmentDate	AppointmentTime	StaffID
---------------	------------	-----------------	-----------------	---------

0 - 0 of 0

Go to: 1

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Orders

Filter in any column

OrderID	OrderDate	BranchID	SupplierID	ProductID	Quantity	OrderTotal
Filter	Filter	Filter	Filter	Filter	Filter	Filter

1 - 1 of 1

Go to: 1

Edit Database Cell

Mode: Text

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Prescriptions

Filter in any column

PrescriptionID	PrescriptionDate	CustomerID	PrescriptionDetails	StaffID
Filter	Filter	Filter	Filter	Filter

1 - 2 of 2

Go to: 1

Edit Database Cell

Mode: Text

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Products

Filter in any column

ProductID	SupplierID	ProductName	ProductDescription	Price
Filter	Filter	Filter	Filter	Filter

0 of 0

Go to: 1

Edit Database Cell

Mode: Text

NAL

Type of data currently in cell: NULL  
0 bytes

Remote

Identity: Public

Name	Commit	Last modified
------	--------	---------------

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Staff

Filter in any column

StaffID	BranchID	Name	Surname	Position	Email	Telephone
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	2	Michael	Gilsenan	Director	mgilsenan@leeopticians.co.uk 07948557697
2	2	3	Carol	Ward	Manager	cward@leeopticians.co.uk 07938576697
3	3	3	Catherine	O'Connor	Head Receptionist	coconnor@leeopticians.co.uk 07948573246
4	4	3	Jennifer	Meehan	Optometrist	jmeehan@leeopticians.co.uk 07928375644
5	5	3	Brian	McKibbin	Optometrist	bmckibbin@leeopticians.co.uk 07983644758

1 of 5

Go to: 1

Edit Database Cell

Mode: Text

1

Type of data currently in cell: Text / Numeric  
1 character

Remote

Identity: Public

Name	Commit	Last modified
------	--------	---------------

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Suppliers Filter in any column

SupplierID	SupplierName	Postcode	Email	Telephone
Filter	Filter	Filter	Filter	Filter

0 of 0

Go to: 1

Edit Database Cell

Mode: Text

Type of data currently in cell: Text / Numeric

1 character

Remote

Identity: Public

Name	Commit	Last modified
------	--------	---------------

SQL Log Plot DB Schema Remote UTF-8

DB Browser for SQLite - C:\Users\Ryan\OneDrive - CNA\A2 2020-2021\Computer Science\A2 Unit 9\Code\LeeOpt.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Login Filter in any column

StaffID	Name	Password	AccessLevel
Filter	Filter	Filter	Filter
1	0 Ryan	admin123	1
2	1 Michael	leeoptscienc2021	1
3	2 Carol	Ward123	1
4	3 Catherine	camlou921	2
5	4 Jennifer	Warrenpost456	3
6	5 Brian	mck899	3

1 of 6

Go to: 1

Edit Database Cell

Mode: Text

Type of data currently in cell: Text / Numeric

1 character

Remote

Identity: Public

Name	Commit	Last modified
------	--------	---------------

SQL Log Plot DB Schema Remote UTF-8


## Login Screen

### Code

```
File Edit Selection View Go Run Terminal Help
LoginScreen.py - Visual Studio Code
C:\Users\Ryan> OneDrive - C&A 2020-2021 > Computer Science > A2 Unit 5 > Code > LoginScreen.py > Toplevel > _init_

70 elif results == 2:
71     accesslevel = ("2")
72     tkinter.messagebox.showinfo("Notification","User Access Level 2")
73 else:
74     accesslevel = ("3")
75     tkinter.messagebox.showinfo("Notification","User Access Level 1")
76
77 def UserLogin(self): #Login function for staff members
78     while True:
79         try:
80             staffid = int(self.Entry1.get()) #StaffID entry
81             password = self.Entry2.get() #password entry
82             with sqlite3.connect("LeeOpt.db") as db: #Connects to the database file
83                 cursor = db.cursor()
84                 find_user = ('SELECT * FROM Login WHERE StaffID = ? AND Password = ?') #Uses a SQL SELECT statement to find the record with the same details input by the user.
85                 cursor.execute(find_user, [(staffid),(password)]) #Executes the SQL statement
86                 results = cursor.fetchall()
87                 if results: #If results are found
88                     for i in results:
89                         self.Access() #Runs the function to display the appropriate access level
90                         MainMenu2.create_Toplevel1(root) #Opens the main menu screen
91                         self.clearentries() #clears entries to provide additional security
92                         self.quit() #closes the login screen
93                     else:
94                         tkinter.messagebox.showerror("Error", "The staffID or password is incorrect, please try again.") #An error message is displayed
95                         time.sleep(1)
96                         return('exit')
97         except ValueError: #If an invalid data type is entered for StaffID
98             tkinter.messagebox.showerror("Validation Error","Please enter the correct data type for StaffID. The field cannot be null and must be an integer.") #An error message is displayed
99             self.clearentries() #entries are cleared to allow the user to try again.
100             break
101
102 def __init__(self, top=None):
103     """This class configures and populates the toplevel window.
104
105 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
106 Windows PowerShell
107 Copyright (c) Microsoft Corporation. All rights reserved.
108
109 Try the new cross-platform PowerShell https://aka.ms/powershell
110
111 PS C:\Users\Ryan\OneDrive - C&A 2020-2021\Computer Science\A2 Unit 5\Code & "C:\Users\Ryan\AppData\Local\Programs\Python\Python38-32\python.exe" "c:\Users\Ryan\.vscode\extensions\ms-python.python-2021.3.68073044\python\files\lib\python\debugpy\launcher" "5
112 3802" -- "c:\Users\Ryan\OneDrive - C&A 2020-2021\Computer Science\A2 Unit 5\Code\CustomForm.py"
113
114 Python 3.8.3 32-bit @ 0.0.7
```

## Result

 Login

Leeopticians

StaffID

Password

Login

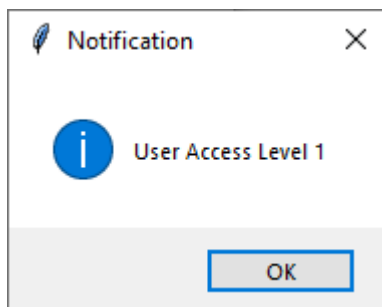


# Login Screen Access Levels

## Code

```
File Edit Selection View Go Run Terminal Help
LoginScreen.py - Visual Studio Code
C:\Users\Ryan> OneDrive - C&A2 2020-2021 > Computer Science > A2 Unit 5 > Code > LoginScreen.py > TopLevel > _init_...
49     return (w, top)
50
51 class TopLevel:
52     def quit(self):
53         w.withdraw() #Closes the window
54
55     def ClearEntries(self):
56         self.Entry1.delete(0,'end') #Deletes all data inside the entry box.
57         self.Entry2.delete(0,'end')
58
59     def Access(self):
60         staffid = self.Entry1.get() #StaffID entry
61         with sqlite3.connect("LeeOpt.db") as db: #Connects to the database file
62             cursor = db.cursor()
63             find_user = ('SELECT accesslevel FROM Login WHERE StaffID = ?') #Searches for the record with the corresponding StaffID and returns the access level.
64             cursor.execute(find_user, [(staffid)]) #Executes the SELECT statement using the input from the GUI.
65             results = cursor.fetchone()
66
67             if results == 1:
68                 accesslevel = ("1")
69                 tkinter.messagebox.showinfo("Notification","User Access Level 1") #The user will be prompted with a message displaying their access level.
70             elif results == 2:
71                 accesslevel = ("2")
72                 tkinter.messagebox.showinfo("Notification","User Access Level 2")
73             else:
74                 accesslevel = ("3")
75                 tkinter.messagebox.showinfo("Notification","User Access Level 1")
76
77     def UserLogin(self): #Login function for staff members
78         while True:
79             try:
80                 staffid = int(self.Entry1.get()) #StaffID entry
81                 password = self.Entry2.get() #password entry
82                 with sqlite3.connect("LeeOpt.db") as db: #Connects to the database file
83                     cursor = db.cursor()
```

## Result

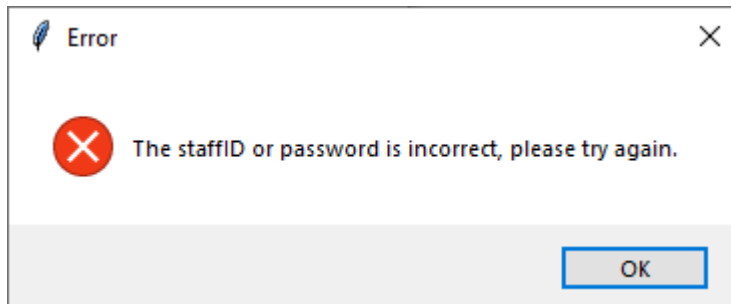


## Incorrect StaffID/Password Combination

### Code

```
93         else:
94             tkinter.messagebox.showerror("Error", "The staffID or password is incorrect, please try again.") #An error message is displayed
95             time.sleep(1)
96             return("exit")
```

### Result

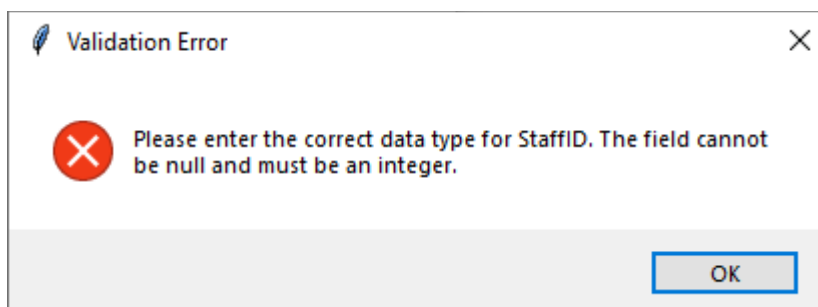


## Invalid Data (Type Check)

### Code

```
97     except ValueError: #If an invalid data type is entered for StaffID
98         tkinter.messagebox.showerror("Validation Error", "Please enter the correct data type for StaffID. The field cannot be null and must be an integer.") #An error message is displayed
99         self.clearEntries() #Entries are cleared to allow the user to try again.
100         break
101
```

### Result



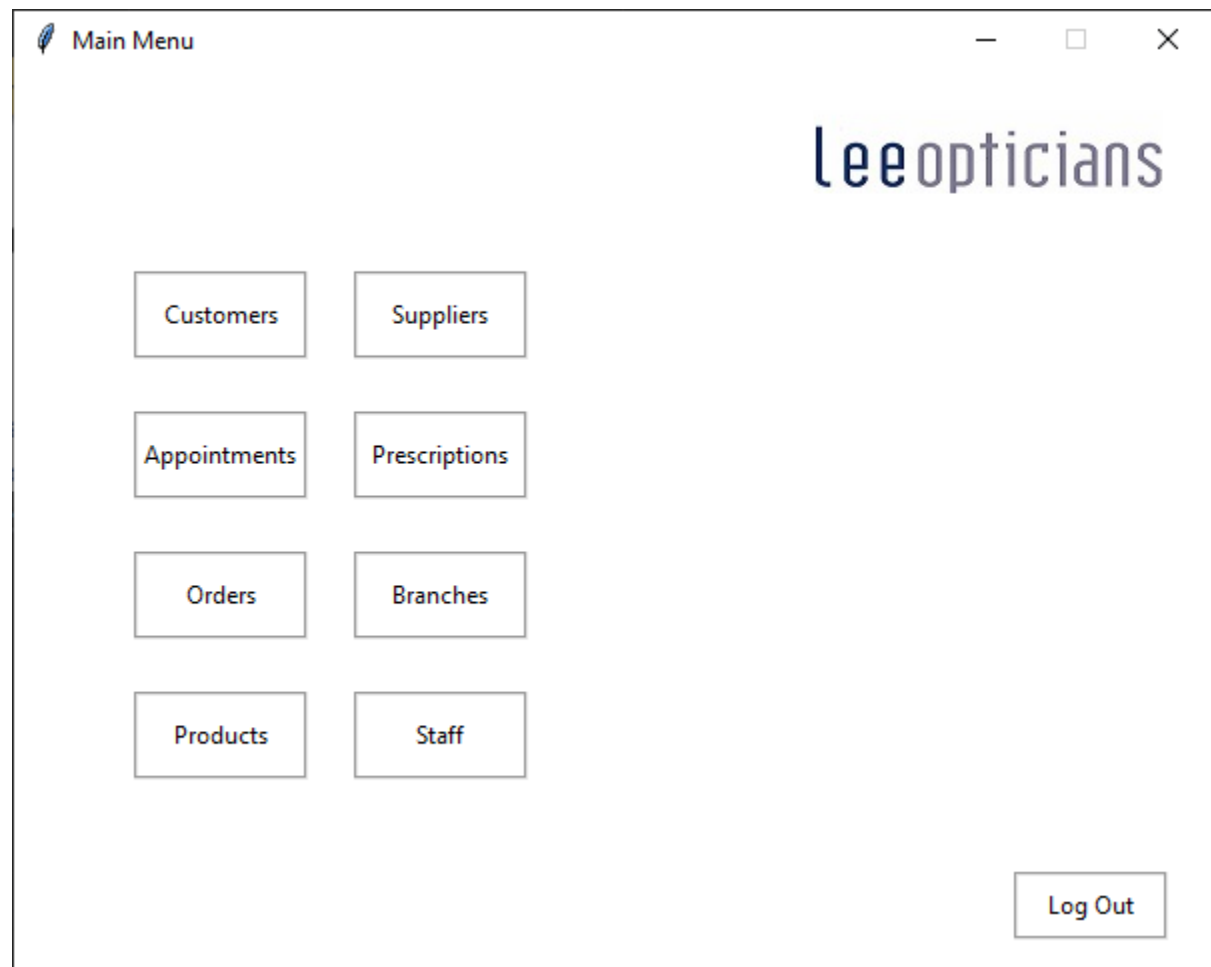
# Main Menu

## Code

```
File Edit Selection View Go Run Terminal Help
MainMenu.py - Visual Studio Code
C:\Users\Ryan> OneDrive - C&A2 2020-2021 > Computer Science > A2 Unit 5 > Code > MainMenu.py > ...
46 def create_toplevel(rt, *args, **kwargs):
47     """Starting point when module is imported by another module.
48     Correct form of call: 'create_toplevel(root, *args, **kwargs)' ."""
49     global w, w_win, root
50     global prog_location
51     prog_call = sys.argv[0]
52     prog_location = os.path.split(prog_call)[0]
53     rt = root
54     root = rt
55     w = tk.Toplevel(root)
56     top = Toplevel(w)
57     MainMenuSupport.init(w, top, *args, **kwargs)
58     return (w, top)
59
60 def destroy_toplevel():
61     global w
62     w.destroy()
63     w = None
64
65 class Toplevel:
66     #Function to close the main menu when the user is finished using the system.
67     def quit(self):
68         w.destroy()
69     #Function to open the customer form.
70     def CustomerTab(self):
71         CustomerForm.create_customerForm(root)
72     #Function to open the appointment form.
73     def AppointmentTab(self):
74         AppointmentForm.create_AppointmentForm(root)
75     #Function to open the branch search form.
76     def BranchesTab(self):
77         BranchSearch.create_BranchSearch(root)
78     #Function to open the order form.
79     def OrderTab(self):
80         OrderForm.create_OrderForm(root)
81
82 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
File "C:\Users\Ryan\OneDrive - C&A2 2020-2021\Computer Science\A2 Unit 5\Code\LoginScreen.py", line 53, in quit
w.withdraw() #Closes the window
AttributeError: 'NoneType' object has no attribute 'withdraw'
Exception in Tkinter callback
Traceback (most recent call last):
File "C:\Users\Ryan\AppData\Local\Programs\Python\Python38-32\lib\tkinter\_init_.py", line 1883, in __call__
return self.func(*args)
File "C:\Users\Ryan\OneDrive - C&A2 2020-2021\Computer Science\A2 Unit 5\Code\LoginScreen.py", line 92, in UserLogin
self.quit() #Closes the login screen
File "C:\Users\Ryan\OneDrive - C&A2 2020-2021\Computer Science\A2 Unit 5\Code\LoginScreen.py", line 53, in quit
w.withdraw() #Closes the window
AttributeError: 'NoneType' object has no attribute 'withdraw'
Python 3.8.3 32-bit @ 0 A 9
Ln 1, Col 1 Spaces 4 UTF-8 CRLF Python
```

```
File Edit Selection View Go Run Terminal Help
MainMenu.py - Visual Studio Code
C:\Users\Ryan> OneDrive - C&A2 2020-2021 > Computer Science > A2 Unit 5 > Code > MainMenu.py > ...
75 #Function to open the branch search form.
76 def BranchesTab(self):
77     BranchSearch.create_BranchSearch(root)
78 #Function to open the order form.
79 def OrderTab(self):
80     OrderForm.create_OrderForm(root)
81 #Function to open the prescription form.
82 def PrescriptionTab(self):
83     PrescriptionForm.create_PrescriptionForm(root)
84 #Function to open the staff form
85 def StaffTab(self):
86     StaffForm.create_toplevel(root)
87
88 def __init__(self, top=None):
89     """This class configures and populates the toplevel window.
90     top is the toplevel containing window."""
91     _bgcolor = "#d9d9d9" # X11 color: 'gray95'
92     _fgcolor = "#000000" # X11 color: 'black'
93     _compcolor = "#d9d9d9" # X11 color: 'gray95'
94     _anadcolor = "#d9d9d9" # X11 color: 'gray95'
95     _anadcolor = "#e0e0e0" # closest X11 color: 'gray92'
96
97     top.geometry("600x400+295+137")
98     top.minsize(120, 1)
99     top.maxsize(1370, 740)
100     top.resizable(0, 0)
101     top.title("Main Menu")
102     top.configure(background="#ffffff")
103
104     self.ButtonCustomer = tk.Button(top)
105     self.ButtonCustomer.place(relx=0.1, rely=0.222, height=44, width=87)
106     self.ButtonCustomer.configure(activebackground="#e0e0e0")
107     self.ButtonCustomer.configure(activeforeground="#000000")
108     self.ButtonCustomer.configure(background="#ffffff")
109     self.ButtonCustomer.configure(borderwidth=1, relief="flat", text="Customer")
110
111 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
File "C:\Users\Ryan\OneDrive - C&A2 2020-2021\Computer Science\A2 Unit 5\Code\LoginScreen.py", line 53, in quit
w.withdraw() #Closes the window
AttributeError: 'NoneType' object has no attribute 'withdraw'
Exception in Tkinter callback
Traceback (most recent call last):
File "C:\Users\Ryan\AppData\Local\Programs\Python\Python38-32\lib\tkinter\_init_.py", line 1883, in __call__
return self.func(*args)
File "C:\Users\Ryan\OneDrive - C&A2 2020-2021\Computer Science\A2 Unit 5\Code\LoginScreen.py", line 92, in UserLogin
self.quit() #Closes the login screen
File "C:\Users\Ryan\OneDrive - C&A2 2020-2021\Computer Science\A2 Unit 5\Code\LoginScreen.py", line 53, in quit
w.withdraw() #Closes the window
AttributeError: 'NoneType' object has no attribute 'withdraw'
Python 3.8.3 32-bit @ 0 A 9
Ln 1, Col 1 Spaces 4 UTF-8 CRLF Python
```

## Result



## Variable Tables

### Database Tables

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
cursor	The cursor for SQLite3 to make changes to the database file.	String	None

## Login

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
staffid	Stores the value input to the staffID entry box by the user.	Integer	Presence check, type check.
cursor	The cursor for SQLite3 to make changes to the database file.	String	None
find_user	Stores the SELECT statement to retrieve the staffID and password from the login table.	String	
results	Stores the results from the SELECT statement included in find_user.	String	None
accesslevel	Stores the access level of the record retrieved by find_user.	Integer	Presence check, type check.
password	Stores the data input to the password entry box by the user.	String	Presence check.
self.Entry1	The entry box for the staffID.	String	None
self.Entry2	The entry box for the password.	String	None
self.Label1	The label for the staffID entry box.	String	None
self.Label2	The label for the password entry box.	String	None
self.Button1	The button used to run the login function with the data input by the user.	String	None
self.Label3	The label for the Lee Opticians logo at the top of the form.	String	None

## Main Menu

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
self.ButtonCustomer	The button to open the customer form.	String	None
self.ButtonAppointments	The button to open the appointments form.	String	None
self.ButtonOrders	The button to open the order form.	String	None
self.ButtonProducts	The button to open the product form.	String	None
self.ButtonSuppliers	The button to open the supplier form.	String	None
self.ButtonPrescriptions	The button to open the prescriptions form.	String	None
self.ButtonBranches	The button to open the branch search form.	String	None
self.ButtonStaff	The button to open the staff form.	String	None
self.LogOutButton	The button to close the main menu.	String	None

## Customer Form

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
self.CustomerEntry	The entry box for the CustomerID.	String	None
self.DOBEntry	The entry box for the date of birth.	String	None
self.TownEntry	Th entry box for the town.	String	None
self.PostcodeEntry	The entry box for the postcode.	String	None
self.EmailEntry	The entry box for the email address.	String	None
self.TelephoneEntry	The entry box for the telephone number.	String	None
self.MedicalEntry	The entry box for medical conditions.	String	None
customerid	Stores the value input to the CustomerID entry box by the user.	Integer	Presence check, type check.
branchid	Stores the value input to the BranchID entry box by the user.	Integer	Presence check, type check.
name	Stores the data input to the name entry box by the user.	String	None
surname	Stores the data input to the surname entry box by the user.	String	None
dateofbirth	Stores the data input to the date of birth entry box by the user.	Date	Presence check, format check.
town	Stores the data input to the town entry box by the user.	String	None
postcode	Stores the data input to the postcode entry box by the user.	String	None
email	Stores the data input to the email entry box by the user.	String	Presence check, format check.
telephone	Stores the data input to the telephone entry box by the user.	String	None
medical	Stores the data input to the medical conditions entry box by the user.	String	None



cursor	The cursor for SQLite3 to make changes to the database file.	String	None
add_customer	Stores the INSERT statement to add customer records to the database file.	String	None
delete_customer	Stores the DELETE statement to remove customer records from the database file.	String	None
update_customer	Stores the UPDATE statement to update customer records in the database file.	String	None
search_customer	Stores the SELECT statement to retrieve the customer record with the CustomerID input by the user.	String	None
results	Stores the results from the SELECT statement included in search_customer.	String	None
self.Frame1	The frame where the buttons are located on the form.	String	None
self.AddCustomerBtn	The button used to run the function to add new records to the database file.	String	None
self.SearchCustomerBtn	The button used to run the function to search for existing records in the database file.	String	None
self.DeleteCustomerBtn	The button used to run the function to delete records from the database file.	String	None
self.UpdateCustomerBtn	The button used to run the function to update existing records in the database file.	String	None
self.CustomerCloseBtn	The button used to run the function to close the window.	String	None

## Appointments

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
self.AppIDEntry	The entry box for the AppointmentID.	String	None
self.CustomerIDEntry	The entry box for the CustomerID.	String	None
self.AppDateEntry	The entry box for the appointment date.	String	None
self.AppTimeEntry	The entry box for the appointment time.	String	None
self.StaffIDEntry	The entry box for the StaffID.	String	None
appointmentid	Stores the value input to the AppointmentID entry box by the user.	Integer	Presence check, type check.
customerid	Stores the value input to the CustomerID entry box by the user.	Integer	Presence check, type check.
date	Stores the data input to the appointment date entry box by the user.	Date	Presence check, format check.
apptime	Stores the data input to the appointment time entry box by the user.	Date/Time	Presence check, format check.
staffid	Stores the value input to the StaffID entry box by the user.	Integer	Presence check, type check.
cursor	The cursor for SQLite3 to make changes to the database file.	String	None
add_appointment	Stores the INSERT statement to add appointment records to the database file.	String	None
delete_appointment	Stores the DELETE statement to remove appointment records from the database file.	String	None
update_appointment	Stores the UPDATE statement to update appointment records in the database file.	String	None
search_appointments	Stores the SELECT statement to retrieve the appointment record with the	String	None

	AppointmentID input by the user.		
results	Stores the results from the SELECT statement included in search_appointments.	String	None
self.Frame1	The frame where the buttons are located on the form.	String	None
self.AddBtn	The button used to run the function to add new records to the database file.	String	None
self.SearchBtn	The button used to run the function to search for existing records in the database file.	String	None
self.DeleteBtn	The button used to run the function to delete records from the database file.	String	None
self.UpdateBtn	The button used to run the function to update existing records in the database file.	String	None
self.CloseBtn	The button used to run the function to close the window.	String	None

## Order Form

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
self.OrderIDEntry	The entry box for the OrderID.	String	None
self.OrderDateEntry	The entry box for the order date	String	None
self.BranchIDEntry	The entry box for the BranchID	String	None
self.SupplierIDEntry	The entry box for the SupplierID	String	None
self.ProductIDEntry	The entry box for the ProductID.	String	None
self.QuantityEntry	The entry box for the quantity.	String	None
orderid	Stores the value input to the OrderID entry box by the user.	Integer	Presence check, type check.
orderdate	Stores the data input to the order date entry box by the user.	Date	Presence check, format check.
branchid	Stores the value input to the BranchID entry box by the user.	Integer	Presence check, type check.
supplierid	Stores the value input to the SupplierID entry box by the user.	Integer	Presence check, type check.
productid	Stores the value input to the ProductID entry box by the user.	Integer	Presence check, type check.
quantity	Stores the value input to the quantity entry box by the user.	Integer	Presence check, type check.
cursor	The cursor for SQLite3 to make changes to the database file.	String	None
result	Stores the results from the SELECT statement included in add_order.	Real	None
price	Stores the calculated value to be inserted into the order total field in the database.	Real	Presence check, type check.
add_order	Stores the INSERT statement to add order records to the database file.	String	None

delete_order	Stores the DELETE statement to remove order records from the database file.	String	None
update_order	Stores the UPDATE statement to update order records in the database file.	String	None
search_order	Stores the SELECT statement to retrieve the order record with the OrderID input by the user.	String	None
results	Stores the results from the SELECT statement included in search_order.	String	None
write	Opens orderinvoice.txt.	String	None
self.Frame1	The frame where the buttons are located on the form.	String	None
self.AddBtn	The button used to run the function to add new records to the database file.	String	None
self.SearchBtn	The button used to run the function to search for existing records in the database file.	String	None
self.UpdateBtn	The button used to run the function to update existing records in the database file.	String	None
self.DeleteBtn	The button used to run the function to delete records from the database file.	String	None
self.PrintBtn	The button used to run the function to write a record to a text file.	String	None
self.CloseBtn	The button used to run the function to close the window.	String	None

## Prescriptions Form

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
self.PresIDEntry	The entry box for the PrescriptionID.	String	None
self.PresDateEntry	The entry box for the prescription date.	String	None
self.CustomerIDEntry	The entry box for the CustomerID.	String	None
self.PresDetailsEntry	The entry box for the prescription details.	String	None
self.StaffIDEntry	The entry box for the StaffID.	String	None
presid	Stores the value input to the PrescriptionID entry box by the user.	Integer	Presence check, type check.
presdate	Stores the data input to the prescription date entry box by the user.	Date	Presence check, format check.
customerid	Stores the value input to the CustomerID entry box by the user.	Integer	Presence check, type check.
presdetails	Stores the data input to the prescription details entry box by the user.	String	Presence check.
staffid	Stores the value input to the StaffID entry box by the user.	Integer	Presence check, type check.
add_prescriptions	Stores the INSERT statement to add prescription records to the database file.	String	None
delete_prescription	Stores the DELETE statement to remove prescription records from the database file.	String	None
update_prescription	Stores the UPDATE statement to update prescription records in the database file.	String	None
search_prescription	Stores the SELECT statement to retrieve the prescription record with the PrescriptionID input by the user.	String	None

result	Stores the results from the SELECT statement included in search_prescription.	String	None
cursor	The cursor for SQLite3 to make changes to the database file.	String	None
write	Opens prescription.txt.	String	None
self.Frame1	The frame where the buttons are located on the form.	String	None
self.AddButton	The button used to run the function to add new records to the database file.	String	None
self.SearchButton	The button used to run the function to search for existing records in the database file.	String	None
self.UpdateButton	The button used to run the function to update existing records in the database file.	String	None
self.PrintButton	The button used to run the function to write a record to a text file.	String	None
self.DeleteButton	The button used to run the function to delete records from the database file.	String	None
self.CloseButton	The button used to run the function to close the window.	String	None

## Staff Form

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
self.StaffIDEntry	The entry box for the StaffID.	String	None
self.BranchIDEntry	The entry box for the BranchID.	String	None
self.NameEntry	The entry box for the staff name.	String	None
self.SurnameEntry	The entry box for the staff surname.	String	None
self.PositionEntry	The entry box for the staff position.	String	None
self.EmailEntry	The entry box for the staff email address.	String	None
self.TelephoneEntry	The entry box for the staff telephone number.	String	None
self.PasswordEntry	The entry box for the staff password to be used to login to the system.	String	None
self.AccessLevelEntry	The entry box for the access level.	String	None
staffid	Stores the value input to the StaffID entry box by the user.	Integer	Presence check, type check.
branchid	Stores the value input to the BranchID entry box by the user.	Integer	Presence check, type check.
name	Stores the data input to the name entry box by the user.	String	None
surname	Stores the data input to the surname entry box by the user.	String	None
position	Stores the data input to the position entry box by the user.	String	None
email	Stores the data input to the email entry box by the user.	String	Presence check, type check.
telephone	Stores the data input to the telephone entry box by the user.	String	None
password	Stores the data input to the password entry box by the user.	String	Presence check.



accesslevel	Stores the value input to the access level entry box by the user.	Integer	Presence check, type check.
cursor	The cursor for SQLite3 to make changes to the database file.	String	None
add_staff	Stores the INSERT statement to add staff records to the database file.	String	None
logindetails	Stores the INSERT statement to add staff records to the database file.	String	None
search_staff	Stores the SELECT statement to retrieve the staff record with the StaffID input by the user.	String	None
results	Stores the results from the SELECT statement included in search_staff.	String	None
delete_staff	Stores the DELETE statement to remove staff records from the database file.	String	None
delete_login	Stores the DELETE statement to remove login records from the database file.	String	None
update_staff	Stores the UPDATE statement to update staff records in the database file.	String	None
update_login	Stores the UPDATE statement to update login records in the database file.	String	None
self.Frame1	The frame where the buttons are located on the form.	String	None
self.AddButton	The button used to run the function to add new records to the database file.	String	None
self.SearchButton	The button used to run the function to search for existing records in the database file.	String	None

self.UpdateButton	The button used to run the function to update existing records in the database file.	String	None
self.DeleteButton	The button used to run the function to delete records from the database file.	String	None
self.CloseButton	The button used to run the function to close the window.	String	None

## Branch Search Form

<i>Variable Name</i>	<i>Description</i>	<i>Data Type</i>	<i>Validation</i>
w	The TopLevel widget.	String	None
self.Entry1	The entry box for the BranchID.	String	None
self.Entry2	The entry box for the town.	String	None
branchid	Stores the value input to the BranchID entry box by the user.	Integer	Presence check, type check.
town	Stores the data input to the town entry box by the user.	String	Presence check.
cursor	The cursor for SQLite3 to make changes to the database file.	String	None
search_branch	Stores the SELECT statement to retrieve the branch record with the BranchID or town input by the user.	String	None
results	Stores the results from the SELECT statement included in search_branch.	String	None
self.Button1	The button used to run the function to search for existing records in the database file.	String	None
self.Button2	The button used to run the function to close the window.	String	None