



VISUAL PROGRAMMING 2 – INF2611 ASSIGNMENT 2

ASSIGNMENT UNIQUE NUMBER:

1. Introduction

The name of my application is sicario. Sicario is a Latin word that means a hired assassin. The concept came up from my experience as a teacher at a school in Mpumalanga. As a class teacher some of the duties one does is manage learner's textbooks and information about a learner.

The application has got the following features:

- keeps track of textbooks issued to learners
- stores a learner's biographical information which is easily retrieved in case of emergency. When a learner is sick or injured the parents need to be notified as early as possible.
- There are two types of views for viewing information, it can be viewed as a table showing the whole class or information can be viewed for an individual learner.
- There is a search facility for searching for a particular learner

2. Database design specifications

2.1.

Code for creating database

```
CREATE DATABASE IF NOT EXISTS grade8D;
```

Code for creating first table

```
CREATE TABLE IF NOT EXISTS learners
(
learner_name VARCHAR(30) NOT NULL,
address VARCHAR(100),
gender VARCHAR(10),
date_of_birth VARCHAR(10),
PRIMARY KEY (learner_name)
);
```

2.2.

```
show tables;
```

```
MySQL 5.7 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.15-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use grade8D;
Database changed
mysql> show tables;
+-----+
| Tables_in_grade8d |
+-----+
| guardians         |
| learners          |
| textbooks         |
+-----+
3 rows in set (0.05 sec)

mysql>
```

show columns from learners;

```
MySQL 5.7 Command Line Client
owners.

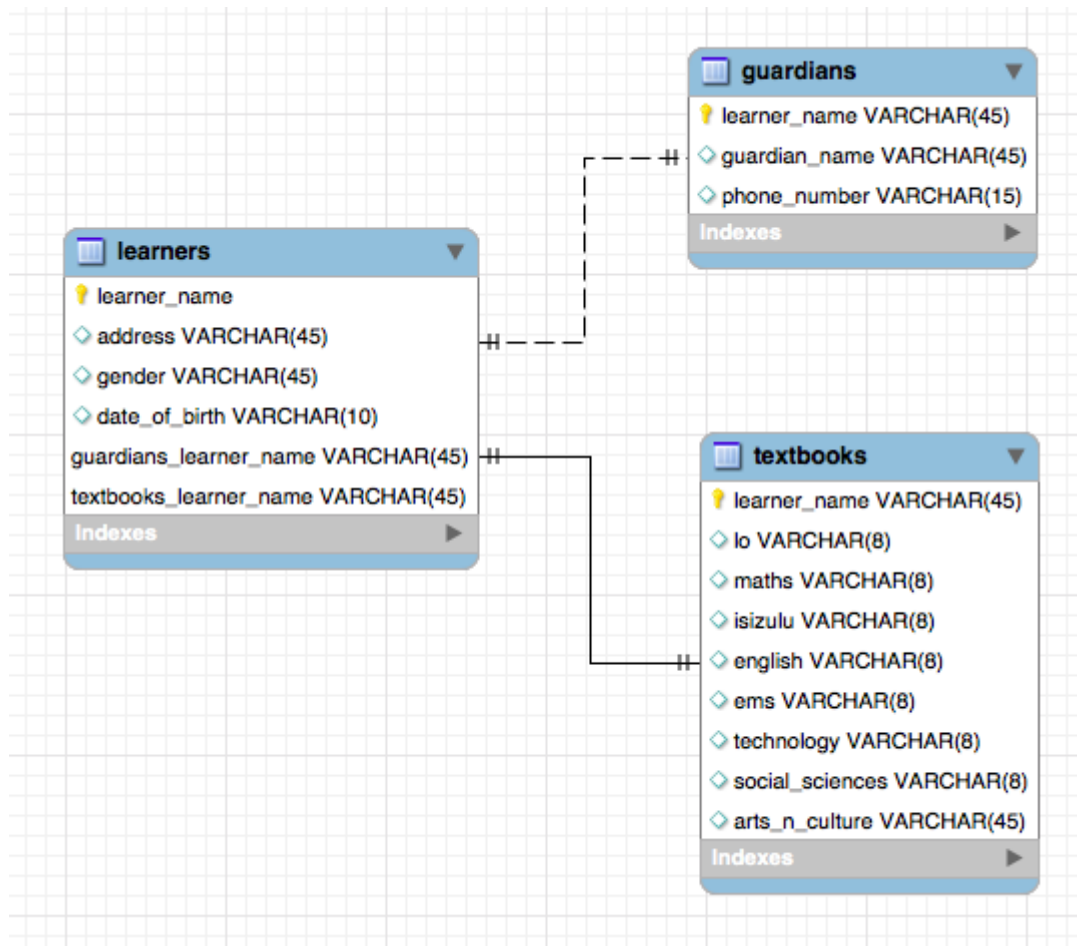
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use grade8D;
Database changed
mysql> show tables;
+-----+
| Tables_in_grade8d |
+-----+
| guardians         |
| learners          |
| textbooks         |
+-----+
3 rows in set (0.05 sec)

mysql> show columns from guardians;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| learner_name | varchar(30) | NO   | PRI | NULL    |       |
| guardian_name | varchar(30) | YES  |     | NULL    |       |
| phone_number  | varchar(9)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.12 sec)

mysql>
```

2.3



The database is made up of 3 tables, namely learners, guardians and textbook.

Learners table stores the general information of a learner. It has four fields with learner name field being the primary key

```

MySQL 5.7 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.15-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

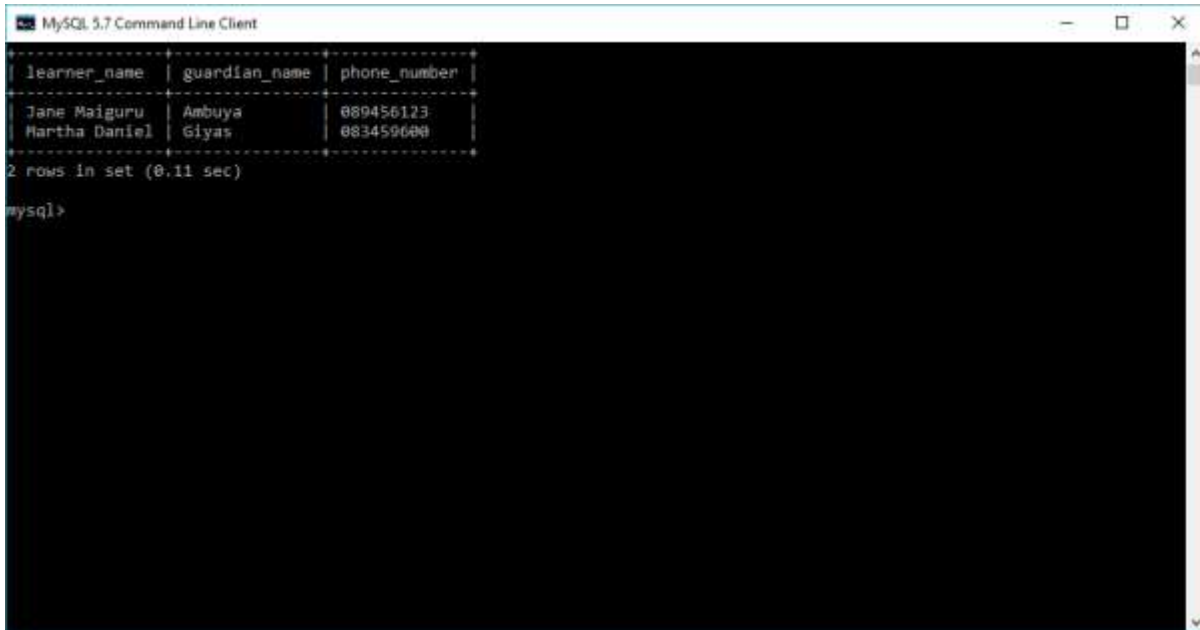
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

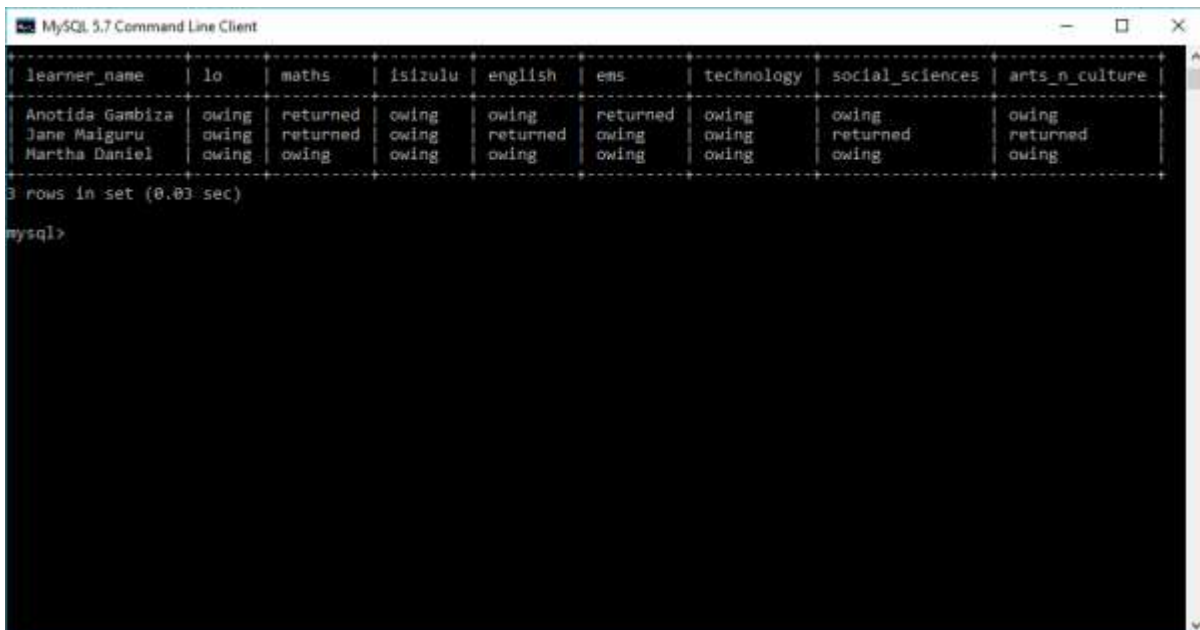
mysql> use grade8d;
Database changed
mysql> select*from learners;
+-----+-----+-----+-----+
| learner_name | address | gender | date_of_birth |
+-----+-----+-----+-----+
| Anotida Gambiza | 1073/24 Stanwest, Standerton | male | 07/06/2013 |
| Jane Maiguru | 10 Mandela Stree Meyerview | female | 01/01/2000 |
| Martha Daniel | 44 Caledon Street Cosmos | female | 30/01/1993 |
+-----+-----+-----+-----+
3 rows in set (0.05 sec)

mysql>
    
```

Guardians table stores information about a learner's guardian. It has 3 fields with the learner name field being the primary key. The guardians table is linked to the learners table by a many to one relationship. A learner can have only one guardian but a guardian can have many learners. The primary key is referenced in the learners table to provide referential integrity.



Textbooks table stores information of the textbooks issued to a learner. The table has nine fields and with learner name being the primary key. This primary key is referenced in the learners table. There is a one to one relationship between the guardians table and the learners table.



3. User interface design

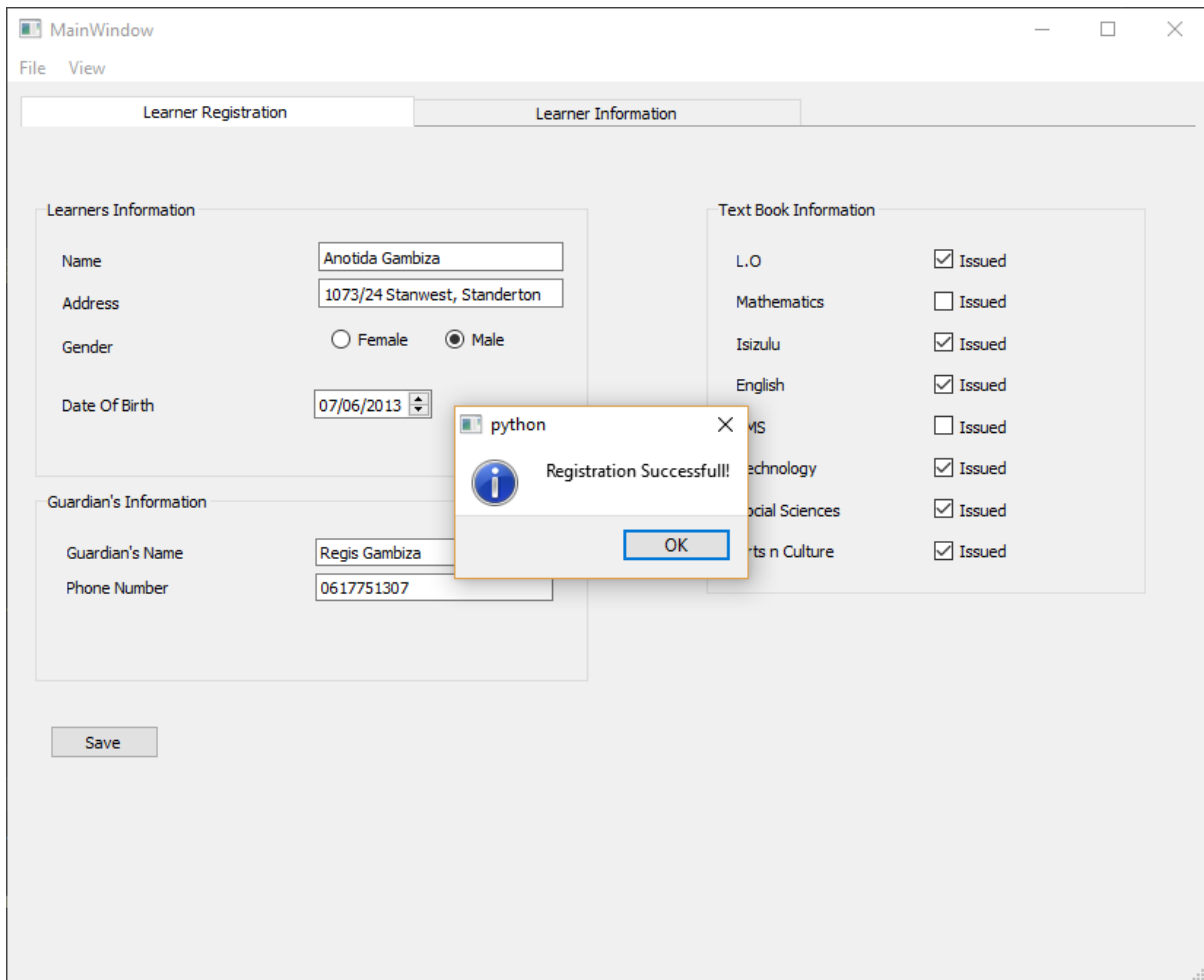
On opening the application there are two tabs, learner registration and learner information. Clicking the registration tab displays the following:

The registration page is where a learner's information is initially saved into the database.

Table 1: Components used in registration page

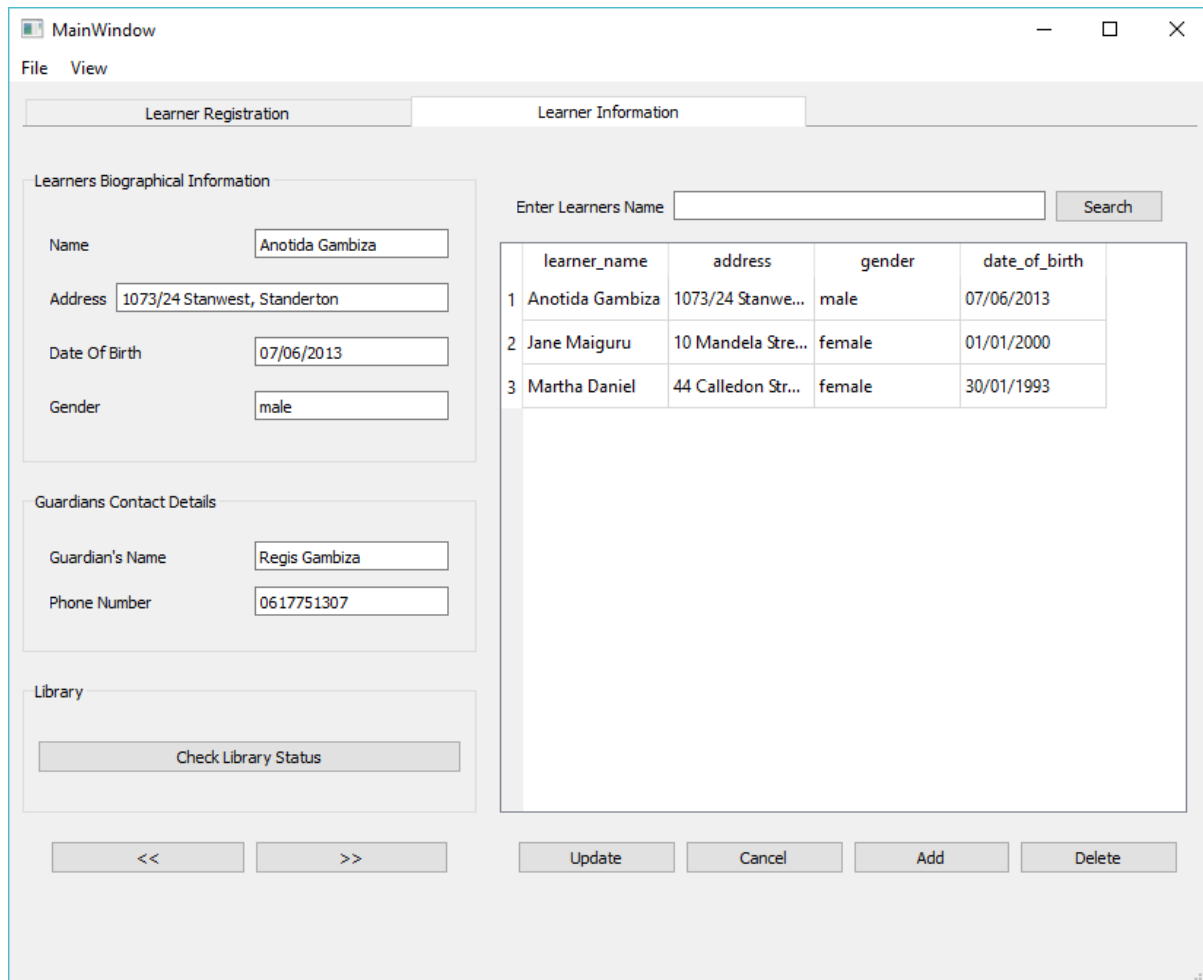
Component	Use
Group box	Group related components
Line edits	Capture information from user
Labels	Display information/identify components
Check box	Selecting certain components
Push button	Save information to the database

Clicking the save push button displays a message box informing the user that was successfully saved.



The component used is a QMessageBox box. Clicking ok closed the message box and more learners can be registered.

Clicking the Learner information tab displays the following.



This page is for viewing information in the database. It is divided into two. On the left you can view information for an individual learner and the right you can view information of all the learners.

Table 1: Components used in Learner information page

Component	Use
Table view	Display information in table format
Line edits	Display information
Labels	Display information/identify components
Push buttons	To start certain actions

Clicking the **check library status** push displays the following returned books page.

	learner_name	lo	maths	isizulu	english	ems	technology	social_sciences	arts
1	Anotida Gambiza	owing	returned	owing	owing	returned	owing	owing	owing
2	Jane Maiguru	owing	returned	owing	returned	owing	owing	returned	return
3	Martha Daniel	owing	owing	owing	owing	owing	owing	owing	owing

The returned books page shows all the textbooks issued to a learner and those that were returned. The tableview widget is used to display the information in table format. This page displays information from the guardians table. The pushbutton is used to close the page.

4. Database manipulation

4.1. Add

In the learner information tab, clicking add creates a new row and a new learner can then be added to the database.

Code snapshot

```
#Function to insert a new row
def InsertRecords(self):
    self.model.insertRow(self.ui.tableView_2.currentIndex().row())

#Connects insert button to the function insert records
QtCore.QObject.connect(self.ui.pushButton_11, QtCore.SIGNAL('clicked()'),
self.InsertRecords)
```

4.2. Edit

Editing in tableview has been enabled, to edit any information the cell to be edited is double clicked and new information is entered the update is clicked to save.

4.3. Delete

To delete a row, the row to be deleted is selected and the delete button is clicked.

Code snapshot**#Function to delete a selected row**

```
def DeleteRecords(self):
    self.model.removeRow(self.ui.tableView_2.currentIndex().row())
    self.model.submitAll()
#Connects delete button to the function delete records
QtCore.QObject.connect(self.ui.pushButton_12, QtCore.SIGNAL('clicked()'),
self.DeleteRecords)
```

4.4. Search

To search for a specific learner, the learners name is entered in the search line edit and the search pushbutton clicked.

Code snapshot**#Filters table view to display matching names only**

```
def FilterRecords(self):
    print("Filter clicked")
    self.model.setFilter("name like'" + self.ui.lineEdit_13.text()+"%")
#Connect the search button to the FilterRecords function
QtCore.QObject.connect(self.ui.pushButton_8, QtCore.SIGNAL('clicked()'),
self.FilterRecords)
```

4.5. Next and previous

When viewing information for an individual learner, next and previous are used to move forward or backward in the database.

Code snapshot

```
def dis_next(self):
    """Display next learner information"""
    print("dis_next called successfully")
    MyForm.recno +=1
    if MyForm.recno > self.model.rowCount()-1:
        MyForm.recno = 0
    self.record = self.model.record(MyForm.recno)
    #Displaying learner info

    self.ui.lineEdit_3.setText(self.record.value("learner_name"))
    self.ui.lineEdit_4.setText(self.record.value("address"))
    self.ui.lineEdit_5.setText(self.record.value("date_of_birth"))
    self.ui.lineEdit_6.setText(self.record.value("gender"))
    #Display guardian info
    self.ui.lineEdit_7.setText(self.record.value("guardian_name"))
    self.ui.lineEdit_9.setText(self.record.value("phone_number"))

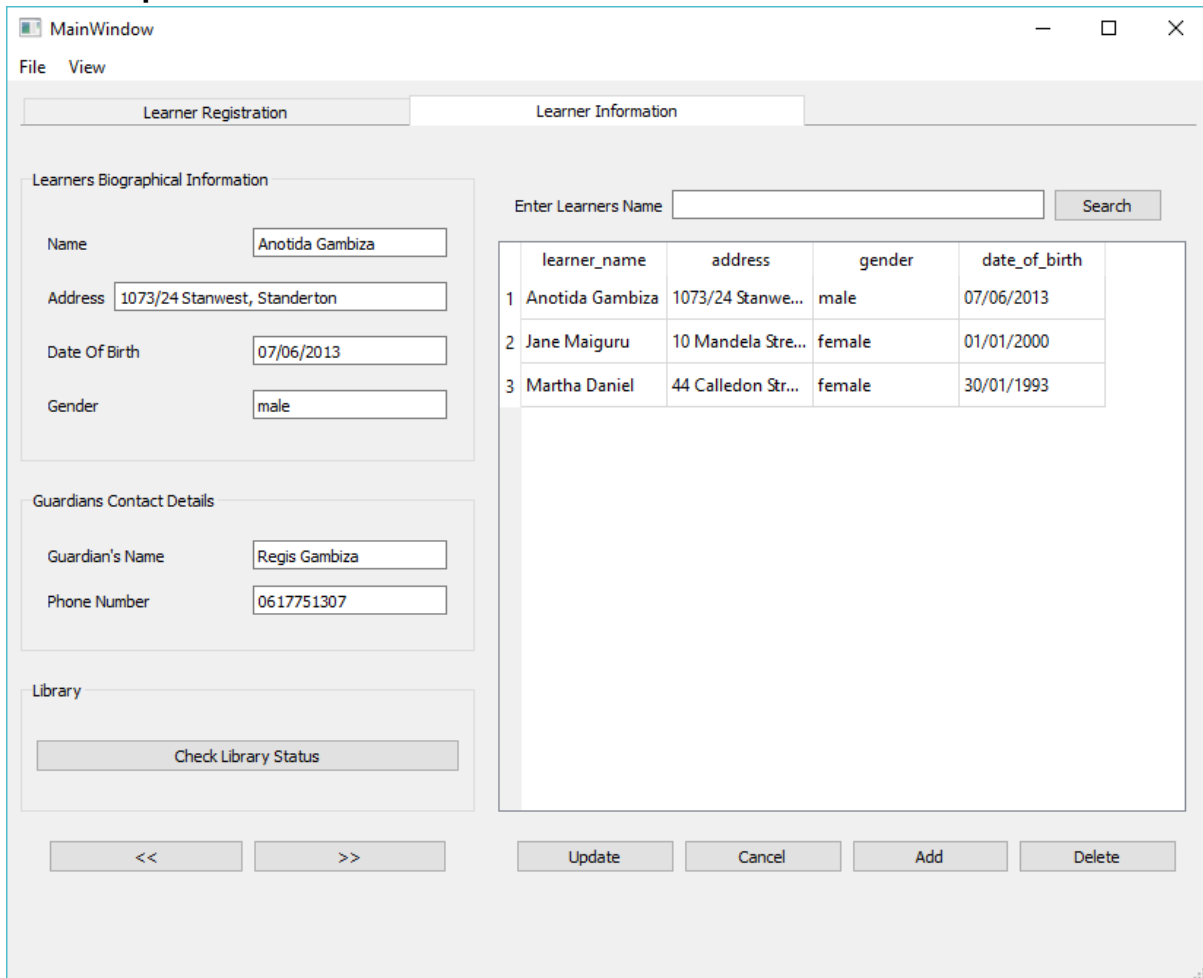
def dis_previous(self):
    """Display previous learner information"""
    print("dis_prev called successfully")
    MyForm.recno -= 1
    if MyForm.recno < 0:
        MyForm.recno=self.model.rowCount() - 1
    self.record = self.model.record(MyForm.recno)
    #Displaying learner info
    self.ui.lineEdit_3.setText(self.record.value("learner_name"))
    self.ui.lineEdit_4.setText(self.record.value("address"))
```

```

self.ui.lineEdit_5.setText(self.record.value("date_of_birth"))
self.ui.lineEdit_6.setText(self.record.value("gender"))
#Display guardian info
self.ui.lineEdit_7.setText(self.record.value("guardian_name"))
self.ui.lineEdit_9.setText(self.record.value("phone_number"))

```

4.6. Manipulation of more than one table



In this window two tables are being displayed simultaneously

Code to display guardians table

```

#Sending data to table view
self.model = QSql.QSqlTableModel(self)
self.model.setTable("learners")
self.model.setEditStrategy(QSql.QSqlTableModel.OnManualSubmit)
self.model.select()
self.ui.tableView_2.setModel(self.model)
self.record = self.model.record(MyForm.recno)
#Displaying learner info
self.ui.lineEdit_3.setText(self.record.value("learner_name"))
self.ui.lineEdit_4.setText(self.record.value("address"))
self.ui.lineEdit_5.setText(self.record.value("date_of_birth"))
self.ui.lineEdit_6.setText(self.record.value("gender"))
#Display guardian info
self.ui.lineEdit_7.setText(self.record.value("guardian_name"))

```

```
self.ui.lineEdit_9.setText(self.record.value("phone_number"))
```